Decrypting the Security+ Beta Exam Objectives

Tcat Houser & Helen O'Boyle
With
Ian Kayne & Angella Hebert

Decrypting the Security+ Beta Exam Objectives

Copyright 2002 AlphaGeekProductions.

Created in United States of America

UPC: 6-43977-21101-8

The sponsoring editor for this book was Bruce Moran and the production supervisor was Chad M. Bayer.

Formatted by www.totalrecallpress.com

Authors: Tcat Houser, Helen O'Boyle, Ian Kayne, and Angella Hebert
Design Concepts: Bruce Moran

LICENSE

This work was “Frozen” 08/23/02_ 18:35

Michael Toennessen

Instructional Technology Specialist

Meeker Junior High School

Kent School District #415

NO WARRANTY

BECAUSE THIS DOCUMENT IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THIS CONTENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK OF USE OF THIS CONTENT IS WITH YOU. SHOULD THIS CONTENT PROVE FAULTY, INACCURATE, OR OTHERWISE UNACCEPTABLE YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MIRROR AND/OR REDISTRIBUTE THIS CONTENT AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE OC, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THIS CONTENT WAS CREATED ACCORDING TO A COMMON BODY OF KNOWLEDGE RELATING TO SECURITY ISSUES. AT THE TIME OF CREATION, SECURITY+ CERTIFCATION HAS NOT BEEN RELEASED. THEREFORE IT IS IMPOSSIBLE TO OFFER ANY WARRANTIES AS TO ACCURACY OR REVELANCE TO SECURITY+.

This book is dedicated to the all the folks willing to ‘take the whooping’ that the Security+ IT beta test delivers to the mind and body.

Tcat Houser

My work on this study guide is dedicated to the denizens of Virginia Commonwealth University computer labs, without whose presence in my life as friends, mentors and occasional adversaries of a white hat who just wanted to learn, I might never have discovered how enjoyable the field of computer security could be, or learned as much about it as I did there.

Helen O'Boyle

This is for Gillian and Eddie.

Jan Kayne

I dedicate the success of this effort to all of the loved ones who gave us moral support through our 7 days of hell, the team members of this success for all roles played, and to the author I owe my sanity to Tcat. You are a strong leader with your expertise and make the desires in all of us run rampid when we think that all of the fires within us have been distinguished. You push us to the challenge. I thank you. Without that little extra you always seem to give, I could not have been such an active participant in this effort.

Angella Hebert

Decrypting the Security+ Beta Exam Objectives

CompTIA Security+ For SYN-101 Examination

About the Authors

Tcat Houser

Tcat Houser (Network+, MCSE, Server+, CTT+) has garnered other milestones such as IBM PSE and Microsoft MCSE/MCT. This is the result of almost 40 years of "fussing with electronics". When not writing or teaching, he is doing research. Tcat accomplishes so many tasks because to him, it isn't work, its fun, and he has fun 18 hours a day, 7 days a week. You can reach him by sending mail to Tcat@Tcat.net or Author@totalrecallpress.com.

Looking back at the last book where I (Tcat Houser) was the lead author (i-Net+ Exam Prep 1576105989) the independent reviews were very good, and most readers understood that I had two intentions. One, ace the test. Two, supply the information to be a reference manual and/or alert the reader to upcoming technologies so as today's hero, the reader was not tomorrow's zero.

In the two years since that release, I led a number of career changers through CompTIA and wrote my own courseware. Building on that learning, you have this release.

Helen O'Boyle

Helen O'Boyle (Network+, MCSE, MCSD, CTT+) has been working with computers for 20 years. Her background is quite varied in regards to both platforms and a complete interest in both the engineering of software and hardware. You may email her at Author@totalrecallpress.com or Hoboyle@mindspring.com.

Ian Kayne

By day I'm a technical specialist with a focus on Internet & Security technologies. By night I'm a scene coder, laying down C++ with Win32, MFC and DirectX code purely for fun. I enjoy a challenge, whether it's building pen-resistant systems or getting this mesh and that pixel shader to render faster. Working with everyone in different time zones and across the Atlantic has been fun. Certain people will understand the reference to Spirit ;).

Angella Hebert

A mom whose passions make her eager to learn while keeping her feet grounded. I contribute my proofreading and English skills. I enjoy the knowledge it gives and of course the friendships I have built along the way (Mr. Tcat himself).

Chad Rees

Cover Design and WebMaster

About the EBook

This manual is designed to provide information to help readers study for and pass CompTIA's Security+ Beta certification exam. Every effort has been made to make this manual as complete and accurate as possible.

Special Offer:

We will accept PayPal Donations because while this work is free, food, etc. is not.

If you do choose to send a $20 USD Donation or more, we will provide you a PDF file which will contain all of our Security reference work.

This Security PDF file, will be released on or before November 01-2002. We estimate this will be about the time when Security+ goes live.

Http://www.alphageekproductions.com

A quick look at the Chapters included in this book:

Read.Me	**1**
General Security Concepts (30%)	**13**
Communication Security (20%)	**41**
Infrastructure Security (20%)	**87**
Basics of Cryptography (15%)	**159**
Operational/Organizational Security (15%)	**171**

Table of Contents

About the Authors VIII
About the EBook X
Preface and Acknowledgments VIII
Frequently Asked Questions x
Post Beta Comments xiii

Read.Me **1**

You're Saved….If 5
Hacker Vs. Cracker 5
Security Tao 6
Security Checklist 8
Security Through Obscurity 8
Technical Oxymoron 9
Resources 9
Summary 10

General Security Concepts (30%) **13**

1.0 General Security Concepts 13
1.1 Access Control 13
1.1.1 MAC/DAC/RBAC 14
1.2 Authentication 15
1.2.1 Kerberos 15
1.2.2 CHAP 15
1.2.3 Certificates 16
1.2.4 Username/Password 16
1.2.5 Tokens 16
1.2.6 Multi-Factor 16
1.2.7 Mutual Authentication 17
1.2.8 Biometrics 17
1.3 Non-essential Services and Protocols 18
1.4 Attacks 20
1.4.1 DOS/DDOS 20
1.4.2 Backdoors 24
1.4.3 Spoofing 26
1.4.4 Man in the Middle 29
1.4.5 Replay 32
1.4.6 TCP/IP Hijacking 33
1.4.7 Weak Keys 34
1.4.8 Mathematical 34
1.4.9 Social Engineering 35
1.4.10 Birthday 35
1.4.11 Password Guessing 35
1.4.11.1 Brute Force 35
1.4.11.2 Dictionary 36
1.4.12 Software Exploitation 36
1.5 Malicious Code 37
1.5.1 Viruses 37

1.5.2 Trojan Horses 37
1.5.3 Logic Bombs 37
1.5.3 Worms 37
1.6 Social Engineering 38
1.7 Auditing 39
Communication Security (20%) 41
2.0 Communication Security 41
2.1 Remote Access 41
2.1.1 802.1x 42
2.1.2 VPN 42
2.1.3 RADIUS 43
2.1.4 TACACS/+ 44
2.1.5 L2TP/PPTP 45
2.1.6 SSH 46
2.1.7 IPSEC 47
2.1.8 Vulnerabilities 48
2.2 Email 50
2.2.1 S/MIME 51
2.2.2 PGP 51
2.2.3 Vulnerabilities 52
2.2.3.1 Spam 54
2.2.3.2 Hoaxes 55
2.3 Web 56
2.3.1 SSL/TLS 56
2.3.2 HTTP/S 58
2.3.3 Instant Messaging 59
2.3.3.1 Vulnerabilities 60
2.3.3.2 - 8.3 Naming Conventions 62
2.3.3.3 Packet Sniffing 62
2.3.3.4 Privacy 63
2.3.4 Vulnerabilities 63
2.3.4.1 Java Script 65
2.3.4.2 ActiveX 66
2.3.4.3 Buffer Overflows 67
2.3.4.4 Cookies 69
2.3.4.5 Signed Applets 71
2.3.4.6 CGI 72
2.3.4.7 SMTP Relay 72
2.4 Directory 74
2.4.1 SSL/TLS 74
2.4.2 LDAP 74
2.5 File Transfer 75
2.5.1 S/FTP 75
2.5.2 Blind FTP/Anonymous 75
2.5.3 File sharing 75
2.5.4 Vulnerabilities 76
2.5.4.1 Packet Sniffing 76
2.6 Wireless 77
2.6.1 WTLS 77

2.6.2 802.11x....78
2.6.3 WEP/WAP....78
2.6.4 Vulnerabilities....80
2.6.4.1 Site Surveys....84

Infrastructure Security (20%) 87

3.0 Infrastructure Security....87
3.1 Devices....88
3.1.1 Firewalls....88
3.1.2 Routers....92
3.1.3 Switches....94
3.1.5 Modems....95
3.1.6 RAS....97
3.1.7 Telecom/PBX....98
3.1.8 VPN....100
3.1.9 IDS....103
3.1.10 Network Monitoring/Diagnostic....106
3.1.11 Workstations....108
3.1.12 Servers....111
3.1.13 Mobile Devices....113
3.2 Media....115
3.2.1 Coax....115
3.2.2 UTP/STP....116
3.2.3 Fiber....117
3.2.4 Removable media....118
3.2.4.1 Tape....119
3.2.4.2 CDR....121
3.2.4.3 Hard drives....123
3.2.4.4 Diskettes....125
3.2.4.5 Flashcards....125
3.2.4.6 Smartcards....128
3.3 Security Topologies....131
3.3.1 Security Zones....131
3.3.1.1 DMZ....131
3.3.1.2 Intranet....133
3.3.1.3 Extranet....133
3.3.2 VLANs....134
3.3.3 NAT....135
3.3.4 Tunneling....137
3.4 Intrusion Detection....139
3.4.1 Network Based....139
3.4.1.1 Active Detection....142
3.4.1.2 Passive Detection....142
3.4.2 Host Based....142
3.4.2.1 Active Detection....143
3.4.2.2 Passive Detection....144
3.4.3 Honey pots....144
3.4.4 Incident Response....145
3.5 Security Baselines....148
3.5.1 OS/NOS Hardening....149

3.5.1.1 File System....150
3.5.1.2 Updates....151
3.5.2.1 Updates (Firmware)....151
3.5.2.2 Configuration....151
3.5.2.2.1 Enabling and Disabling Services and Protocols....151
3.5.2.2.2 Access control lists....152
3.5.3 Application Hardening....153
3.5.3.1 Updates....153
3.5.3.2 Web Servers....153
3.5.3.3 Email Servers....153
3.5.3.4 FTP Servers....153
3.5.3.5 DNS Servers....154
3.5.3.6 NNTP Servers....154
3.5.3.7 File/Print Servers....154
3.5.3.8 DHCP Servers....155
3.5.3.9 Data Repositories....155
3.5.3.9.1 Directory Services....155
3.5.3.9.2 Databases....156

Basics of Cryptography (15%) 159

4.0 Basics of Cryptography....159
4.1 Algorithms....159
4.1.1 Hashing....159
4.1.2 Symmetric....159
4.1.3 Asymmetric....159
4.2 Concepts of using cryptography....160
4.2.1 Confidentiality....160
4.2.2 Integrity....160
4.2.2.1 Digital Signatures....160
4.2.3 Authentication....160
4.2.4 Non-Repudiation....161
4.2.4.1 Digital Signatures....161
4.2.5 Access Control....161
4.3 PKI....161
4.3.1.1 Certificates....162
4.3.1.2 Certificate Policies....162
4.3.1.3 Certificate Practice Statements....162
4.3.2 Revocation....162
4.3.3 Trust Models....162
4.4 Standards and Protocols....164
4.5 Key Management/Certificate Lifecycle....165
4.5.1 Centralized vs. Decentralized....165
4.5.2 Storage....165
4.5.2.1 Hardware vs. Software....165
4.5.2.2 Private Key Protection....165
4.5.3 Escrow....165
4.5.4 Expiration....166
4.5.5 Revocation....166
4.5.5.1 Status Checking....166
4.5.6 Suspension....167

4.5.6.1 Status Checking 167
4.5.7 Recovery 167
4.5.7.1 M of N Control 167
4.5.8 Renewal 167
4.5.9 Destruction 167
4.5.10 Key Usage 168
4.5.10.1 Multiple Key Pairs (Single, Dual) 168
Operational/Organizational Security (15%) 171
5.0 Operational/Organizational Security 171
5.1 Physical Security 171
5.1.1 Access Control 171
5.1.1.1 Physical Barriers 172
5.1.1.2 Biometrics 172
5.1.2 Social Engineering 173
5.1.3 Environment 173
5.1.3.1 Wireless Cells 173
5.1.3.2 Location 174
5.1.3.3 Shielding 174
5.1.3.4 Fire Suppression 174
5.2 Disaster Recovery 176
5.2.1 Backups 176
5.2.1.1 Off Site Storage 176
5.2.2 Secure Recovery 177
5.2.2.1 Alternate Sites 177
5.2.3 Disaster Recovery Plan 178
5.3 Business Continuity 178
5.3.1 Utilities 179
5.3.2 High Availability / Fault Tolerance 179
5.3.3 Backups 179
5.4 Policy and Procedures 181
5.4.1 Security Policy 181
5.4.1.1 Acceptable Use 181
5.4.1.2 Due Care 182
5.4.1.3 Privacy 183
5.4.1.4 Separation of duties 183
5.4.1.5 Need to Know 183
5.4.1.6 Password Management 183
5.4.1.7 SLA 184
5.4.1.8 Disposal / Destruction 184
5.4.1.9 HR Policy 185
5.4.1.9.1 Termination 185
5.4.1.9.2 Hiring 185
5.4.1.9.3 Code of Ethics 186
5.4.2 Incident Response Policy 186
5.5 Privilege Management 187
5.5.1 User/Group/Role Management 187
5.5.2 Single Sign-on 187
5.5.3 Centralized vs. Decentralized 188
5.5.4 Auditing 188

5.5.5 MAC/DAC/RBAC....188
5.6 Computer forensics....190
5.6.1 Chain of Custody....191
5.6.2 Preservation of evidence....192
5.6.3 Collection of evidence....193
5.7 Risk Identification....194
5.7.1 Asset Identification....194
5.7.2 Risk Assessment....194
5.7.3 Threat Identification....195
5.7.4 Vulnerabilities....195
5.8 Education....196
5.8.1 Communication....196
5.8.2 User Awareness....197
5.8.3 Education....197
5.8.4 Online Resources....197
5.9 Documentation....198
5.9.1 Standards and Guidelines....198
5.9.2 Systems Architecture....198
5.9.3 Change Documentation....198
5.9.4 Logs and Inventories....198
5.9.5 Classification....199
5.9.5.1Notification....199
5.9.6 Retention/Storage....199
5.9.7 Destruction....199

Preface and Acknowledgments

Helen and Tcat were working on what we thought Security+ should look like when CompTIA dropped a bomb on August 15th, 2002. They announced very cryptic beta test objectives (a pun on Security+), and further announced the test would go live on the 26th of August.

Typically, we would have cheered for being given a clear direction on our research. However given the bowl of alphabet soup and (only) 10 days from the announcement to the availability of the beta, we were not celebrating. Instead words were exchanged best not put in print. After a few hours of soul searching, we decided to take some sections that we had completed and create a verbose outline against the rest of the objectives to be released **before** the beta exam.

Given that scenario, this book should be sub-titled 7 days in hell with Security+. We are the first to admit, it isn't complete. And we came to the conclusion that something to study from beats nothing to study from. If you are not delighted with this group effort, your refund check is in the mail. OOPS! You didn't pay anything for this, so that is exactly your refund amount.

While we will have a complete reference work done later, this is what we had time for.

As this preface & acknowledgments are being put to screen, we have about 48 hours before this work is frozen. Honestly, I don't know what will be completed (or not) before it becomes a PDF with a MD5 hash.

What will/did happen is we froze it before the test goes to beta. As the term hacker means to some people, someone bad, brain dump is sometimes used to indicate a person who relates actual test questions.

The original definition of a hacker was someone who studied, and did NO damage. A person/group that did damage of any sort was/is called a cracker.

A lifetime ago, I (Tcat) was in the US Army. We called a brain dump what to study, in other words, a guide. Today, some call a brain dump the revealing of actual test questions and suspected answers. Well these fingers call the latter behavior illegal in the civilized word. This body of work is a brain dump in the classic sense. To make the point, we have frozen this work before the test goes to beta.

Not one of your authors or contributors was involved in or with CompTIA and the Security+ test in any way. Helen, Tcat and our contributors have done our very best to give you a roadmap while keeping in mind a concept we call ethics. We would honestly appreciate your feedback, and please no actual test questions. If you think we were weak or flat out incorrect on our training material, we want to hear about it. These fingers are pretty sure that given the limited time, this work isn't perfect. And our little team thinks we gave you the data you need (OK, Text in Objective 4 is a bit light on material. I flat ran out of time. Follow the footnotes to more detail.)

Team Effort

As always, nothing happens without a team effort. In no particular order you get a decent shot at the Security+ beta thanks to:

Ian Kayne who contributed some fine technical language to this work and tech editing.

Angella Hebert for playing traffic cop and language editor.

Larry Maxwell for research and shoring up logistics.

Harry Brelsford for the insights on releasing this work to the world without demanding payment.

Bruce Moran for support on the test simulation that is separate from this PDF (also free). And the formatting assistance with this revised document.

Chad Rees for web site everything and graphics.

Pamela Fanstill for moral support and editing.

Http://www.netlinknorthwest.com for the practical support with 802.11

www.TotalRecallPress.com for hosting the downloads

These folks have contributed to your success without expectation of return.

Finally, best wishes from all of us to all of you.

The Decrypting the CompTIA Security+ Objectives Team

Frequently Asked Questions

Q: I'm interested in Security+. What details do you have?

A: The beta test is available from August 26, 2002 to September 30, 2002. 120 Minutes to complete 125 questions, in English. Body of knowledge is 2 years practical experience. Beta fees are $90. USD. Beta exam # is SY1-101[1]

B. This work is based on the beta objectives as provided by CompTIA[2].

Q: Why are you giving away your work for free?

A: For a number of reasons, which you may or may not think, are good.

1. Security is a part of the price we all have to pay for freedom. Even if you have no plans to take the Security+ test, you can protect yourself better if you understand how security works.
2. Shameless self-promotion. Helen and I are looking for paying work right now and we wanted to keep our skills sharpened, as well as, tell you we have other work out there that is pretty good. Check out ISBN 159095202-2, 159095263-4, they are Network+ titles that get good reviews, alas no marketing. Also we have some A+ stuff such as ISBN 159095200-6. In the arena of Test Simulation, Tcat is pretty proud of the i-Net+ (IKO-002), Network+ N10-002, and A+ O/S (220-202) available from Http://www.BFQ.com This text and the test simulation are also available at:

   a. http://www.totalrecallpress.com/downloads.php
   (French and Spanish versions will be available soon.)

   b. We believe in the work of CompTIA. This non-profit is not perfect (being made up of humans), yet, we don't know of a better group in existence.

3. **We will accept PayPal donations because while this work is free, food, etc. is not. If you do choose to send us $20 USD or more, we will give you a PDF of our reference work, which will be done when Security+ goes live.**

   Http://www.alphageekproductions.com

---

[1] http://www.comptia.org/certification/securityplus/beta_objectives.htm
[2] http://www.comptia.org/certification/securityplus/beta_objectives.pdf

Q: By putting out a study guide for a beta test won't the passing score for the final test be really high?

A. We hope so. We want Security+ certification to have meaning.

Q: How did you know what to write about?

A: We are not mind readers, but security has basic premises. We started writing about these premises quite a while ago, and just started mashing them into the objectives that were released by CompTIA. We didn't write or see any CompTIA test questions, since the test has not been released prior to this work, we haven't been able to take the test. This is not the cheat sheet guide to Security+. It is a quick overview with footnotes that link to data on the Internet for more information.

Q: Aren't you telling the bad guys how to do things?

A: In a word, no. There are a number of sites on the Internet that tell a person step by step how to cause havoc. We have just put together a guidebook that gives you the basics and where to find more information for defense.

Q: There are comments back and forth between the authors. Did you forget to take that out?

A: Humm, no. Here's the deal. Unless you have law enforcement background, saying "Chain of Evidence" or "Chain of Custody" probably sounds like a new heavy metal rock band.

This work was "frozen" by Mike just over a week ago into a PDF and emailed to over 100 people, including managers at CompTIA. It was not well formatted, and we wanted to demonstrate the work was done before the beta was available the following week. With that accomplished, we have cleaned up formatting. So, the body of work was modified with track changes ON. This makes an evidence chain of what was altered.

Chain of Evidence and Chain of Custody are two objectives in Security+. While our final reference work has some nice sample forms, we didn't have time to accomplish that in this work. So, you have an analogy of the concepts.

This is the same work we released previously, just cleaned up. If you really want the version with less than excellent formatting, write to Tcat@tcat.net . I'll fix ya up.

Q: Besides the two of you, who else helped with this work?

A: A small army. The bulk of the word beyond Helen and Tcat fell on the following people. There are others listed in Preface and Acknowledgments

Ian Kayne
By day I'm a technical specialist with a focus on Internet & Security technologies. By night I'm a scene coder, laying down C++ with Win32, MFC and DirectX code purely for fun. I enjoy a challenge, whether it's building pen-resistant systems or getting this mesh and that pixel shader to render faster. Working with everyone in different time zones and across the Atlantic has been fun. Certain people will understand the reference to Spirit ;).

Bruce Moran of BFQ
Contributed raw data, test administration software and engine to create a test simulation for Security+ (SYN-101). This test simulation and the required engine are available at http://www.bfq.com

Angella Hebert
A mom whose passions make her eager to learn while keeping her feet grounded. I contribute my proofreading and English skills. I enjoy the knowledge it gives and of course the friendships I have built along the way (Mr. Tcat himself).

Chad Rees
Cover Design and WebMaster

Additions

Angella did some clean up with Track Changes on.

Bruce Moran re-formatted and re-created the PDF

At: (September 01, 12:00)

French and Spanish versions will be released shortly.

A MD 5 hash was created to confirm integrity.

Post Beta Comments

I (Tcat Houser) took the beta test on 29 August 2002. The following is what I posted on the Internet to discussion groups on Security+

Tcat's Post Beta Security+ report.

Difficulity (1-10) 4 on par with Network+

Fairness (1-10) 9 one of the more fair tests from CompTIA.

Decrypting the Security+ Beta Objectives 'hit rate' 93%

Comments:

DAMN glad we put the study guide out BEFORE the test.

Proves either we're oracle's or everyone wrote to the objectives,

including the SME's.

If the material makes sense, the beta is yours.

Wish I had more time to have worked on PKI.

Had to guess a two or three questions that would

have been obvious if I was an admin in a large firm.

We will have a cleaned up version of the PDF out

monday. Again, if the material makes sense, go

save $100 and if your feeling kind, send a few

bucks back via http://www.alphageekproductions.com

Tcat

"Security isn't technology. Security is a process, though it is enabled by technology."

James Christiansen
CSO, General Motors

Chapter 0000

Read.Me

This brief chapter is for *everyone*. It doesn't matter if you are a manager, a technical guru or in marketing. While the body of this work examines security issues in depth, here you will discover the overview of security. In July 2002, Bill Gates gave an update on Microsoft's Security Initiative. In short, it took the company two months, not one to begin to clean up the software holes, at a cost of 100 million bucks. More importantly, his memo called upon users doing their part.

Giving this work a quick glance could leave the reader with a thought similar to, "we're doomed." This certainly does not have to be the case.

In this title you will discover that:

A. Security is an ongoing challenge, not a one-time fix. Consider it job security.
B. Security costs can be quantified and given a positive Return On Investment (ROI).
C. Security approaches must be balanced between physical security, technical security and people.
D. The biggest challenge in security is people.
E. While there are no absolutes, many attacks are easily stopped.

We begin your overview by examining these points.

As the chief security officer of GM stated, security is a process. Forget for the moment that new holes are found in systems that have existed for years (such as the veritable Apache web server), and, remember that new technologies give rise to new issues. Because security is not a pleasant thought for most of us, there is a human tendency to just pretend it doesn't exist or at best, demand a "fixit!" so it doesn't have to be thought about again. Sorry! That isn't how the world works. Any psychology student can tell you that fear of the unknown is a very powerful emotion. This in fact is what drives the irrational need to either ignore security issues or demand a one-time permanent solution.

There is an old saying “knowledge is power”. That truism builds on the behavior of human fears of the unknown. This leads to the next bullet point.

Security costs can be quantified and can be given a positive ROI. While more on this matter is discussed in the legal chapter, the Secure Business Quarterly[3] reports, “Results demonstrate that efficient gains greater than 3 percent are realized when introducing appropriate security. The quantifiable benefits are decreased maintenance costs and obvious savings due to the reduction of successful attacks.

If the functional ROI is not a motivator for you, consider the legal ramifications. Not taking prudent security measures will vary in outcome, depending on your industry. If you are lucky, you will be explaining this as a lesson learned while interviewing for your next job. If you happen to work in an industry[4] that requires not only security, but, filing a Suspicious Activity Report (SAR) and you fail to do so, you may not have to worry about finding a new job. The Federal government could be providing you a job, along with food, shelter and clothing for a long time in the Iron Bar hotel.

Continue our high-level view by considering the next bullet point.

Security is a matter of balancing technical, physical and people issues. The first point in this bullet is balance. Balance is a reference to both the degree to which you invoke a solution on one of the three points, and balancing the points of physical, technical and personal against themselves.

A strong caution is offered here and now. This book was written to provide you the data to pass the CompTIA Security+ as well as give you data to be successful in the real world. CompTIA is short for Computer Trade Industry Association. This makes a reasonable assumption that Security+ as a test focuses on the technical portion in what we refer to as the Triangle of Security. This means we have gone beyond the Security+ test and given you a map to succeed and excel in the real world. When you pass Security+, you have our pat on the back, but tat does not make you a security guru. Another group has put a fair amount of effort into security, which does not just focus on the technical side.

---

[3] Volume one: Issue Two http://www.sbq.com/sbq/rosi/sbq_rosi_efficiencies.pdf

[4] 12 CFR Part 21 Minimum Security Devices and Procedures, Reports of Suspicious Activities, and Bank Secrecy Act Compliance Program http://www.occ.treas.gov/fr/cfrparts/12CFR21.htm

This group is called the CISSP or Certification for Information System Security Professional. This is a higher-level certification that requires years of documented field experience as part of the certification process. The core of the CISSP philosophy is Planning Policies, Procedures and People. As a Security+ professional you may be reporting to a CISSP. Keep in mind they have more experience and training on the actual implementation for what your authors have called the Triangle of Security. You may be a specialist in technicality, and that is wonderful, but a specialist in the U.S. Army is called a specialist too, which does not make his title a commanding officer. Remember your role on the team, and don't be too smug with your new certification. This human thought brings us to the next bullet point.

The biggest challenge in security is people.

There is a term for this challenge. It is called, social engineering. There is one school of thought that suggests social engineering is not part of the security issue. The argument attempts to split the technology from people. While your authors understand the difference, this is a work concerning security and we have kept in mind that people are part of the security issue.

The first part of your effort here is 'selling' the idea that security is in the other person's best interest. It was Samuel Clemens (also known as Mark Twain) who, in his essay "What Is Man?" stated that, "every one of us acts at all times in his own interest." The trick is to supply the data to the users in a form through which security can be properly applied to suit their own best interests.

Your authors do not mean to imply that most workers don't care about their jobs. Social engineering is the most successful form of attack used by the outside intruder. For example, the first phone call reveals the name of the person in charge of the phone system. The second phone call wants to know the make and model of the phone system. This gives the intruders the data to make the third phone call to get a person do perform some simple step, as requested by (insert obtained name from the first call). With this, the intruders just got an employee, who thought they were doing their job, to prop open a door on the phone system for themselves. A perfect example for this attack is the Kevin Mitnik scandal.

This is just one example that was carefully chosen to highlight a typical method into your network that is not thought of. Modems tend to be forgotten because they are not as new and super fast as a network broadband connection.

Even at 28Kbps, a modem that is not monitored can over a few days be the loophole in a fair amount of data slipping out undetected.

This brings us to the last bullet point, which is Risk Assessment, and then we get to end with some good news.

The sad reality is the fact that most computer networks have little to no protection. Our guess is this is the "I don't want to think about anything bad" approach. In our research we found two disturbing issues. The first is that an unprotected computer will be hacked within 2 to 3 days of being plugged into the Internet. This is a statistical average[5]. There are reports at honeynet.org of being invaded within 15 minutes of going on-line. Brian & Tom's Linux book goes so far as to suggest that if a computer has a broadband connection to select "no Ethernet" as the selection when first installing a Linux configuration[6] due to multiple reports of being hacked within 10 minutes.

The good news is overall while there are many numerous 'bad guys' out there, most of them are what are known as 'script kiddies'. This means they grab some software designed to probe for gross lapses in security. Once one of the many targets has been identified, they run additional software that will cause havoc. The script kiddie has virtually no technical knowledge. The typically self-taught script kiddie has little more understanding than a monkey that has learned to "push button – get bananas". It is the sheer number of script kiddies existing that make reports of being hacked within minutes a reality. This doesn't sound like good news, does it? Aah, but there is good news. Because the script kiddie is pretty clueless about the underlying technology, combined with so many completely naked targets, a digital equivalent of checking to see if the car door is locked makes the typical site uninviting unless they can walk right in. Security specialists call this Low Hanging Fruit (LHF). The last chapter in this work "Thinking Outside the Box" reveals the depth of this statement with an email (used with permission) between one of your authors and Fred Cohen. Mr. Cohen is one of the pioneers of computer security with hundreds of publications on the topic over a 20-year period and popularized the term 'computer virus'. Further, he has one of the most popular "honey pot" programs ever.

[5] http://www.honeynet.org/papers/stats/
[6] http://www.orbdesigns.com/pages/btlb/ch03/btlb/btlb_c03.html

We are sorry to report there are elements that are possibly more damaging then the script kiddie, and the good news is it hardly takes any time or money to spot this group. And, that is our next topic,

You're Saved….If

If you have scanned the first couple pages of this book you understand there are tradeoffs for every choice. Your authors wish to caution you to *any* "solution" that appears to be a cure-all. Over 100 years ago the western United States had marketing types who rode in wagons selling magic in a bottle to the citizens in less than pleasant conditions. They were known as 'snake oil' salesmen. Any marketing guru that promises a sure-fire cure to end all should be considered a 21st century version of snake oil. There are numerous legitimate solutions that drastically aid in a challenge.

Again, there are no absolutes in this game. The worm Code Red ripped through the Internet because of holes in Microsoft's Web Server, IIS in early 2002. By the time the year was half over; successful attacks on Linux systems ran 2 to 1 over Microsoft systems.

The best solution today is probably tomorrow's less than best, and may become 'fish wrap'. As this introduction has pointed out several times, security is a process, not something you throw some attention to and forget. Any zealot telling you something to the effect that Linux is secure and Windows isn't has not done their homework. And if Redmond suddenly starts proclaiming that Windows is more secure than Linux, well that's marketing. The reality is everybody, including Apple, is a target, simply because it is there.

In this introduction, we have thrown a number of terms at you. In the interest of clear communications, we want to introduce a couple more.

Hacker Vs. Cracker

Language is a fluid thing. In days of old (in Internet Time) a hacker was a person who studied for possible vulnerabilities in a system and reported the results. A cracker was a term for someone who did bad things to a system once they discovered some vulnerability or came upon the work of a hacker's results. Http://www.hackers.com is on a mission to be a seeker of knowledge without the destructive use of this information. As people who need data to successfully perform in a Security+ role, the goal of hackers.com is in alignment with our needs. Given that, we understand the need to help language differentiation and support their goals. Hacker is a knowledge seeker. A cracker is the bad guy seeking to hurt you.

A recent study was released on the personality of a hacker.

Quoting by InfoSecNews[7].

"In a study consisting of a questionnaire and longer-form answer section started at the hacker convention H2K and Def Con 8 in 2000, Bernadette Schell, dean of Business Information Technology, University of Ontario Institute of Technology and John Dodge, professor at the School of Commerce and The Department of Math and Computer Science at Laurentian University, profiled 216 hackers and their style of thinking, coping with life, and problem-solving."

The researchers found that the respondents, whose median age was 25, have

"extremely low" tendencies towards terrorist and obsessive traits and possess

"relatively balanced temperaments," according to Schell.

"Respondents also tested as particularly creative," she said, noting that the top score for creativity was 20 and that 62 percent of those polled scored 15 or higher on the test.

The combination of creativity and problem-solving styles revealed a commonality between hackers and a group that might not expect they have much in common with hackers, corporate presidents and chief executive officers. The combination of analytical and directive problem-solving styles is shared by both hackers and corporate executives, Dodge said."

Security Tao

Tao is a term that generally means the subtle reality of the universe cannot be described. While choosing the word Tao would seem to suggest it is not possible to describe security, our goal is to give you an outline on the general 'how' of security, without pretending to say, 'step by step, here are all of your answers'.

[7] http://www.c4i.org/isn.html

In the simpler times of living in a domain, the number one challenge was physical security. There was no computer called the domain controller, or network cable to defeat. A person defeating domain security would be destroying his own family and possessions, effectively reducing the threat. This allowed physical security to be evaluated to a high science. Without boring you about details, such as the openings in the walls were slanted to deflect incoming arrows, let's look at the logistics of protecting a village. The basic plans are the same today.

Villagers would work outside the castle walls in fields. The castle had watchtowers placed so a complete view of any possible threat could be observed, while allowing for preparation time. When a watchtower sounded an alarm, the villagers would retreat behind the castle walls, the drawbridge was raised and defensive counter measures were lit up (often literally).

Defenders with long bows had the advantages of height and protection from the castle walls. They would begin by sending arrows out in an attempt to break the solid front of the attackers. If there were dry fields between the opposing factions, the arrows would be on fire, to create a 'wall of fire' (firewall) causing further delay and damage to the attacker.

Assuming this was a serious attack, the defenders would then pour hot oil on the advancing attackers from atop the castle walls. Catapults would be used to throw rocks and/or fireballs at the rear of the attacking force destroying supplies.

With proper planning, a castle would have stores of supplies to outlast a siege (blockade) while the opposing force had no practical method to penetrate the castle walls.

The elements briefly described here demonstrate a defensive system that while old is still effective. It is known as a zone of security. Zones of security offer a defensive point while buying time to either start heating up oil to pour on the attackers or to page an administrator to begin counter measures. Another common element is that zones of security start out as simple physical/psychological barriers, yielding to more hardened elements.

Today, you may find concrete barriers preventing parking within a given distance of an airport functioning as the outside perimeter. The closer in you come to the airport, you find closed circuit TV and police.

Move to the inside of a terminal and you find in addition to the closed circuit TV and police, plain clothes (undercover) security forces, along with inspections. This is a modern day example of zones of security.

Zones of security are not limited to physical attributes. This concept can and should be applied to all forms of security.

Security Checklist

Regardless of physical or network considerations the following points need to be addressed. They are:

A. Proof of identity (authentication)
B. Access permissions (authorization)
C. Eavesdropping prevention/Encryption (confidentiality)
D. Protection of information from modification (integrity)
E. Proof of involvement (non-repudiation)
F. Connectivity between parties (availability)

An effective overall security plan addresses these six issues. Typically this requires a collection of resources. A single element may offer more than one attribute. For example, strong encryption of data addresses both confidentiality and integrity. Other means must be used for the other factors.

Physical security should be modeled with the same principles. Consider a system involving access cards without connectivity. This makes checking a centralized database (authorization) or logging access (non-repudiation) difficult at best.

Security Through Obscurity

More than one article has suggested security via obscurity. The theory goes that by selecting a less than popular O/S, such as AIX from IBM or an older Mac selection, as the O/S of choice the back hat would move on due to "it isn't exciting" enough, and finding published flaws would not be as easy, making them move on to other LHF.

The sole devil's advocate we could find to this argument came from Writing Secure Code[8] by Michael Howard and David LeBlanc. On page 34 "…it is trivially easy for an attacker to determine obscured information." Other parts of this book show many examples of how such information can be found.

[8] ISBN 0-7356-1588-8

Appendix B, “The Ten Immutable Laws of Security” and Appendix C “The Ten Immutable Laws of Security Administration”, make this book a must have on the reference shelf of every IT person.

Technical Oxymoron

From a technical viewpoint we face an oxymoron. Under the single umbrella of security we have two opposing solutions. One is to filter out potentially bad stuff, such as closing ports or examining traffic for something bad. The other solution is to encrypt at some level of the OSI model. The challenge lies in the fact that once you encrypt at a given layer of the OSI model, you can no longer filter it because it is encrypted!

Whoever came up with the phrase, “the devil is in the details” may well have been thinking of the challenges in security.

Resources

We have divided this work into three separate thoughts. The primary is to be what you need for the real world. The second is to let you smoke Security+. The third is to be a resource, not only as a stand-alone book but your guide to the best expert authorities of a specific sub-topic.

Since the specifics of security change with the clock, we will point you to the best sources we have found. Since this opening chapter is a generic chapter, we offer you a generic email list of great value. That is the INS list at attrition.org. It was from that list we were alerted of an article at the ComputerWorld site, even before the email alert from the ComptuerWorld web site came to one of your authors. The particular article can be found online[9]. In an effort to both entice you, as well as, a small offering that we are not just making this up, we offer a few lines from Dan Verton, posted July 18, 2002, beginning next.

“A hacker nicknamed RaFa is the ex-leader of the now defunct World of Hell defacement group, which racked up thousands of Web site defacements before disbanding last year. He said that in addition to making simple configuration mistakes, most administrators don't keep up with updates and patches released by their software vendors.”

[9] www.rootkit.com

"They don't update services running on the system, and, they set up permissions and software settings the wrong way on the Web server," said RaFa. "Think about all of the zero-day exploits I've used. The vendors knew about 90% of those."

“However, the real problem is not laziness, it's trust,” said Genocide, the leader of the Genocide2600 hacker group. “Most administrators and corporate managers simply trust that they are secure,” he said.

"That is their first and biggest mistake," said Genocide. "People believe that since their company may not have anything that someone would want they are free from attack. What administrators really need to do is treat every day as if they were at war and as if the enemy were always planning an attack,” he added.

"It's the companies, administrators and CEOs that don't see it that way who become the easy targets," said Genocide. "They are the ones who don't keep their firewalls, intrusion-detection systems and software upgraded."

Summary

In closing, you learned that there are steps within easy reach with a positive ROI if by chance you were either a high profile target or a ‘personal challenge’ to an intruder because of your LHF. These steps allow you to cut them off and sleep at night. By balancing the triangle of security, physical, technical and people side with on-going efforts you will accomplish all that you set before you. You discovered there are many resources available to you, and failure to protect yourself and your company could bring you serious legal grief.

Chapter 0001

General Security Concepts (30%)

1.0 General Security Concepts

Security can be broken down into three thoughts. A password is something you know. A smartcard is something you have. Biometrics is something you are; it may be a fingerprint, an eye scan (iris or retinal) or a voice pattern. Currently the most popular biometric is the unique way a person enters keystrokes on a keyboard. This is popular because it is non-intrusive (as opposed to hold still while the laser scans your eyes), and the least costly of the biometric options. See Also 5.1.1.2

1.1 Access Control

Access Control can mean different things to different Operating Systems. Access Control to an AS/400[10] is different than to a UNIX or NT based machine[11]. Briefly, Access Control is a combination of Authentication (proving who you claim to be) and Authorization (what are you allowed to see, presuming you are whom you claim you are.)

Different Operating Systems have different capabilities when it comes to enforcing access control. The Security+ test examines three different types of Access Control. These three types are covered in the next section. Before moving on to 1.1.1 we would like to point out that all three control types have some degree of being **real**. The point we are attempting to make is any Windows 3.x/9.x/Me Operating System is, despite any claims of the marketing department from the vendor in Redmond, WA, DOS based. We are not knocking DOS. It is a great **stand-alone** Operating System. Hooking a machine up with one of these Operating Systems has NO security at the client machine level when placed on a network. If you don't want to log into the network with one of these O/S offerings, just press the ESCape key and you are in the machine locally.

[10] http://www.inetmi.com/pubs/aaTut.htm
[11] http://directory.google.com/Top/Computers/Security/Authentication/

1.1.1 MAC/DAC/RBAC

(Also see 5.5.5)

It appears that Domain 1 of Security+ is definitional in nature. For more details on these three Access Control types, read up in 5.5.5 later in this work.

Mandatory Access Control is military-strength access control. In absence of permission, you are not getting in. Every thing is an object and every object gets a label. Everything has to match up for access. MAC is hierarchical in nature. For example, someone with a secret clearance can see confidential, but not top-secret. Labels may be used to define projects. This means that while you may have a top-secret clearance, you are not automatically granted access to a secret project if that particular project is not assigned to your area.

Discretionary Access Control is what is more typically found in the PC world. NT/W2K, Linux for the most part use DAC. Limitations found in DAC that do not exist in MAC are little items such as copying. If you have read access in DAC, you can copy the data (via copy/paste). That cannot happen with MAC.

DAC uses an Access Control List ACL.

Role-Based Access Control[12]. One of the most challenging problems in managing large networked systems is the complexity of security administration. Today, security administration is costly and prone to error because administrators usually specify access control lists for each user on the system individually. Role-based access control (RBAC) is a technology that is attracting increasing attention, particularly for commercial applications, because of its potential for reducing the complexity and cost of security administration in large networked applications.

With RBAC, security is managed at a level that corresponds closely to the organization's structure. Each user is assigned one or more roles, and each role is assigned one or more privileges that are permitted to users in that role. Security administration with RBAC consists of determining the operations that must be executed by persons in particular jobs, and assigning employees to the proper roles. Complexities introduced by mutually exclusive roles or by role hierarchies are handled by the RBAC software, making security administration easier.

[12]http://csrc.nist.gov/rbac/

1.2 Authentication

As stated previously, authentication is proving in some way who you say you are. This can be with a password (something you know), a device such as a smart card (something you have) or by biometrics (something you are). See *1.2.8* and *5.1* for more on biometrics. Authentication is not a slam-dunk when considering security[13]

1.2.1 Kerberos[14]

Kerberos[15] is used when a user wishes to avail his self to a network service. The user presents a ticket that is issued by the Authentication Server (AS). This ticket is a demonstration that the user is whom they say they are. For a ticket to have validity, it must be linked to the user with a high degree of confidence, for example, a password. The ticket is encrypted and has an expiration date.

Kerberos assumes the use of a strong password.

When a user makes a request to log into a network service, the AS makes two keys, known as session keys. The nature of Kerberos is to prevent eavesdropping and replay attacks. Kerberos provides for integrity in a data stream and provides for encryption (confidentiality). Kerberos uses mutual authentication.

1.2.2 CHAP

Challenge Handshake Authentication Protocol authenticates a user by sending the client a random value in which the sender and receiver share a predetermined secret. The client works with the random value that is sent, along with the ID and the secret, to create a hash using MD5. The result of the MD5 value is sent back to the server. The server performs the same function. In theory, the MD5 hash values will be equal, which gives authentication. By changing the ID value with each session, a replay attack is not possible. See 4.1.1 for an explanation of hashing.

Note that firms such as Cisco and Microsoft can have variations on the basic CHAP model.

[13] http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci816117,00.html
[14] http://www.ietf.org/rfc/rfc1510.txt
[15] http://www.faqs.org/faqs/kerberos-faq/user/

1.2.3 Certificates

Refer also to 4.3.1 and 5.5.2

1.2.4 Username/Password

The username is a human name which typically, 'under the hood', becomes equated with a number set that was generated at the time the user account was created or a hashed value. The password is supposed to be kept secret and should be sufficiently long enough to prevent a brute force (trying all the permutations) attack. Further, a password should not be a name found in a native language to prevent a dictionary attack.

1.2.5 Tokens

Not sure where the SMEs are going with this. It could be: a hardware device (sometimes known as a smart card), which changes an ID code on a frequent basis. This code is the basis for encrypting data. Since the code is always changing, it makes it difficult to 'crack the code'.

Submitted by Ian Kayne Sunday August 24, 2002 7:52 AM

1.2.5: I'm guessing it's referring to SecureID tokens. They are hardware devices, about the size of a box of TicTacs with an LCD screen on the front. The screen displays a seeded number which changes every 60 seconds. When you connect to the server (normally a VPN tunnel etc), you enter your PIN number followed by the number showing on your token. The server then authenticates you based on that, as the server has the same seed as your token, knows your PIN, and can therefore confirm the 2 match. This complies with the "something you have and something you know" model.

1.2.6 Multi-Factor

In the introduction we said something like a password or pass phrase is something you know. A device such as a smart card is something you have. Biometrics is something you are.

Multi-Factor takes any two of these concepts and uses them together. This greatly increases security. There is a high probability you use Multi-Factor authentication everyday. The password (something you know) is weak; it is a mere 4 digits and numeric only.

Yet most of us find that combining this incredibly weak password with something you have to be sufficient security. If you haven't guessed yet, it is your ATM/Check Card/Debit Card.

1.2.7 Mutual Authentication

Also refer to 4.3

Mutual authentication may be two-way IE. The server is authentic to a client and the client is authentic to a server. Another option is for a client and a server to trust a third party, such as a certificate authority.

1.2.8 Biometrics

While biometrics may be something you are, there is many a something that falls into this category. The least intrusive of all current biometric devices is software that compares how long you hold a key on a keyboard and the speed at which you type. Your typing style is almost as unique as a fingerprint. This is also the least expensive option.

Other options include fingerprint scanning or iris or retinal scanning of the eye.

1.3 Non-essential Services and Protocols

By simple math, the more services and protocols a host has running, the more targets an attacker has to aim at. As an example, if he can't find an exploit for "Sulphur FTP Server v1.0", running on the host, he can move on to attacking "Ravian SNMP Management Tool v2.3". But if the host in question is simply a file and print server, are these services required? They may come installed by default as part of the operating system, but they provide potential routes into an organization by the unwary system administrator. Similarly, unrequired, but open ports on boundary firewalls are inviting targets for attackers to probe.

There are 2 approaches to this problem. First, you can choose the optimistic route. This involves leaving everything exactly as it is, and only removing services and closing access points (such as firewall ports), as they become an issue. An example of this may be the IIS.htr remote overflow exploit16. An optimistic system administrator may put a default installation of IIS onto the corporate network and hope for the best.

Then, when the .htr advisory is released, the system administrator may choose to disable the .htr extension filter only. Unfortunately, because the administrator was on holiday and didn't read the advisory until 3 days later, the corporate web server was already hacked and "trojaned" (see 1.4.2) before the hole was closed. Now, the administrator has a lot more than just an IIS extension filter to worry about, which is why the second approach is recommended.

The second approach is the pessimistic route. You take the view that nothing on your network is required, and close every port, service and share. You then open only the ports that are specifically required and justifiable, while keeping firewall rules extremely tight. For example, a corporate web server that is publicly accessible from the Internet would only require port 80 inbound to be opened on the firewall – a web server should never independently make a connection out, and unless it's running other services it should never be on anything other than port 80. This methodology extends to services on servers themselves. The IIS exploit mentioned above would be ineffective against this corporate web server if the system administrator had disabled unused IIS extensions prior to deploying it, and so, the 3 day holiday didn't result in the compromise of corporate security.

---

[16] http://www.eeye.com/html/Research/Advisories/AD20020612.html

This approach does not stop at boundary firewalls, routers and servers. “Defense in depth” is an important concept. Apply the same level of security to your internal systems as you would to your external systems. Remember that according to various studies at least 50% with numbers ranging up to 80% (average +/- 70%) of compromised data comes from within.

Remember of course that a balance must be struck between functionality and security. Section 5 deals with Operational and Organizational Security.

1.4 Attacks

To make a general statement, you can split malicious attacks into 2 broad categories – Protocol/Service based and Application based. While application based attacks strike at flaws in pieces of software (such as the IIS Unicode attack, which allowed attackers to traverse out of the website directory structure and into the operating system of the machine running it); protocol attacks (such as Teardrop) strike at the mechanisms used to transfer data on a network. In some instances, the protocol or service can be used against itself.

1.4.1 DOS/DDOS

One of the more prolific attacks of recent times is the DOS attack, or "Denial Of Service". It's based on a simple premise – the attacker attempts to flood the target with large amounts of data so that the network it sits on becomes saturated with this data. Eventually a device on the network (such as a firewall or router) or the targeted host itself will succumb to this flood of data, and stop serving legitimate requests. The variant on this attack – DDOS, or "Distributed Denial Of Service" – produces the same result by sending a coordinated flood of data from multiple hosts. These hosts are usually machines that the attacker has previously hacked and "trojaned" with a DDOS client such as Trinoo. At the attackers signal, these hosts spring into life and start sending data to the target as quickly as possible. This type of attack is becoming more of a concern as more home-users gain broadband connections and place systems on public networks without properly securing them first. To help you understand how this happens, we have a brief account of a recent incident.

February 2000 gave rise to the first widespread amplified attack. The nature was serious enough to involve the Federal Bureau of Investigation (FBI) and the President of the United States. Accordingly, each attack is by time of occurrence, along with internal thinking. The first was Yahoo!. At first, the Information Technology group thought it was equipment failure. The flood peaked at one Gigabit per second. The service was down for five hours. Buy.com was the next target, the next day. Coming from different points, the flood ran about 800 Megabits per second, from different sources.

Later that afternoon, it was time to sink eBay. This firm did not release the details of the amount of data. Their only response to questions when pressed for technical details was, "We are taking multiple measures to fight this."

These events proved to be a new form of attack. How this was accomplished is outlined next.

High performance computers with access to large Internet connections were targeted by port scanning to find security holes, allowing access to the root account. As mentioned in chapter two, a daemon is a servant. The attacker installed Unix daemon on the machines. Using strong encryption, the attacker told the daemons what IP addresses to attack. Using a client machine, the attacker launched all the daemon at once.

The missing piece in the equation, in the heat of the battle was, where all this processing power and Internet access with such bandwidth originated? The attacker planted the daemons on some of the most powerful computers with 'fat pipes' at the locations with the least security, large college campuses.

To make these attacks harder to block, attackers normally spoof the source IP address of the data to appear as though it originates from a different network. This makes it harder for the network administrators (whose devices this flood is passing through) to filter and drop the data. Due to their simplicity DDOS attacks are a favorite tool of "script kiddies", a term used to describe amateur hackers with little skills, who just use tools and exploits created by other people without really understanding what they are doing. It takes minimal work to run an automated IIS hack script against an IP subnet, successfully break into a number of unsecured web servers, and then, trojan them for use in a DDOS attack.

Why do these attacks work?

Unfortunately, having sufficient bandwidth to cope with the flood of data is not necessarily enough to protect you. These attacks not only exhaust bandwidth, they can be specifically targeted to take down hosts on the network by resource exhaustion. For example, if XYZ Corp has a single web server sitting on a 10gigabit line to the Internet, it is unlikely that an attacker could summon enough DDOS clients to sufficiently exhaust the bandwidth on XYZ Corp's network segment. However, they may attack the web server directly. Let's briefly examine how the TCP protocol works.

When a client machine somewhere on the Internet wishes to view a web page on XYZ Corp's web server, they type the domain name into their web browser. After name resolution has taken place, and the client machine knows the IP address of the web server, it sends a TCP SYN packet to it. The web server then allocates the client a port to communicate on and replies with a TCP SYNACK packet. Finally, under normal circumstances the client will reply with a TCP ACK packet; the TCP session will be established and the HTTP data will be transferred from web server to client.

The issue here lies with the TCP "conversation". Once the web server receives the initial SYN packet, it has no choice but to allocate a port for communication with the client prior to replying with a TCP SYNACK. If, the client then does not respond with a TCP ACK, the web server cannot complete the conversation. However, a "feature" of the TCP protocol is that it has error-tolerance and will keep the port open for a while in case the client's SYNACK packet has become lost or damaged in transit. This means that the web server has now allocated a resource – one port – to a conversation that will never finish. If an attacker were to make repeated connections to a server without terminating the TCP conversation properly, the server would quickly exhaust all available ports , become unable to accept any new connections, and thus, preventing legitimate users from accessing the server. Generally the time that the TCP stack waits before resetting a port is more than enough for an attacker to send new requests to exhaust more resources. This type of attack is called a SYN Flood, as flooding the target with SYN packets is precisely what it does.

There are many other variants on the DOS attack. ICMP (more commonly known as “ping”, after the command line tool) floods are used to send ICMP_ECHO packets to a single host, which quickly and exponentially exhausts bandwidth on the target host’s network. The reverse ICMP flood, or Smurf attack, sends a ping to the broadcast address of the target network. Because packets sent to the broadcast address are seen by and responded to by all hosts on the network, you potentially have hundreds of machines replying to one single ping packet. As you can imagine, this would also quickly exhaust the available bandwidth.

Other DOS attacks include variants on the ping flood, IGMP fragmentation, out-of-bounds nukes and teardrop attacks.

What can I do?

DDOS attacks are hard to prevent, and are, unfortunately, a fact of life on public networks. However, there are some simple precautions you can take. First of all make sure you have a good relationship with your ISP and that you have an emergency contact number to reach a technical person. Time wasted calling around to find the right person to help means time wasted getting your public services back online. Secondly, many operating systems & TCP/IP stack implementations provide an option for changing the timeout on a TCP conversation.

If you are able to reduce the amount of time before a reset of an unfinished TCP connection occurs, you will make it harder for an attacker to continually keep the server’s resources occupied. Thirdly, consider whether to configure your boundary routers and firewalls to drop ICMP packets. This is both a blessing and a curse. It does reduce the effectiveness of ICMP floods by preventing any responses from inside your network (note that it cannot by definition prevent ICMP packets from actually arriving at your network boundaries), however, troubleshooting network connectivity problems becomes much more troublesome without the help of the good old ping command. Finally, be a good netizen. If client machines on your private networks have access to public systems, implement source filtering on boundary firewalls and routers. Source filtering prevents spoofed IP packets from leaving your network, as any device they pass through will check the source address against the known LAT. If the source IP does not match an internal subnet, the packet is discarded.

1.4.2 Backdoors

Trojans and rootkits are types of “backdoors”. But what exactly is a “backdoor”? The simplest description is a deliberate configuration or specific program designed to allow access to a system without requiring the usual security checks. This doesn’t necessarily mean it’s solely for a hacker’s benefit. A system administrator may indeed leave a backdoor into all clients’ PCs that he installs for users just in case they inadvertently lock themselves out. However, the type of backdoor we are concerned with here is the malicious kind.

Malicious backdoors range from simple to complex and feature-laden. One of the simpler backdoors is NetBus, a trojan that allows complete remote access to any system it’s installed on. It provides a remote-control type desktop, much like PCAnywhere or Microsoft’s RDP, giving an attacker complete control over a machine as if they were physically in front of it. NetBus is relatively unsophisticated though, and easy to detect.

A far more complete package in the trojan arena is BackOrifice. BackOrifice provides a configuration screen where you select the modules you wish to include and the overall configuration of listener ports, behavior, etc. When you’ve set your configuration, BackOrifice builds a custom executable for you to distribute as you wish.

There are a wide range of modules available for BackOrifice, including port mappers, remote control, key capturing and application binding (where your BackOrifice executable attaches itself to system files like a virus).

Rootkits are a different breed of backdoor. Rather than attaching themselves to executable files and running on the host operating system, rootkits attempt to drill down to the kernel level and replace, modify or divert core operating system functions. As a basic example, a rootkit with a filename rootkit.exe may prevent any process from showing this file in its output, for example Windows Task Manager. This makes them very difficult to detect, as they are operating at the kernel level and effectively filtering what applications and users are allowed to see. Normally, rootkits are seen under (and originally come from) UNIX based operating systems, however, versions are available for Windows[17].

17

They take their name from the UNIX system administrator name “root”, which has complete control over the system – they literally are “kits to provide you with root access”.

One of the more insidious facets of Trojans and rootkits is their creation and purpose. To code a Trojan or rootkit requires a detailed understanding of the target system’s architecture and some considerable coding ability. But to use these tools requires only a few point-and-click type operations, which again makes them favorite tools of the script kiddie. Searching Yahoo for “+rootkit +download” returned almost 1,400 matches, which seemed to be split between tools to detect rootkits, and tools to create them.

One often overlooked backdoor can be user-installed applications. A user who installs VNC so he can access his desktop machine from home creates an inviting target for an attacker, especially, if he has not configured a session password. NetCat, a network administration tool, can also become a potential backdoor. With a single command under Windows, it’s possible to get NetCat to bind a command shell to a port so that incoming telnet sessions on the port receive a DOS prompt as if they were sitting at the local machine. Legitimately installed network diagnostic tools can be quickly turned against the infrastructure by an attacker.

Most modern anti-virus software will quickly pick up backdoor software, however, very new trojans or rootkits can be easily missed. In this instance, detection becomes much harder.

However, analysis of network traffic and processor/memory utilization will usually yield some clues. Once a box has had a backdoor installed, it cannot be considered safe to use until it has been wiped and rebuilt from scratch. Take the machine off the network immediately and perform your forensic investigations. Once you have finished, power the machine down, boot from a write-protected “clean” floppy disk and remove all data & partitions from the system. Then reboot using your operating system install floppies or CD, and reinstall the operating system from scratch.

Trojans are also one of the best reasons for ensuring your firewall rules are correctly configured. Stick to the “Deny by default” methodology and ensure that your firewall prevents all but the minimum requirement of traffic to pass through it. If, an attacker configures their trojan to listen for incoming connections on port 31337 and your firewall only allows traffic to pass on port 80, you’re making it much harder for them to gain further access into your network.

1.4.3 Spoofing

The word “spoof” or “spoofing” is defined as “a hoax” or “to deceive”. When applied to networking, spoofing is the practice of making data appear to originate from a source different than its true origin. To understand how this is possible, we need to go back to the origins of TCP/IP itself.

TCP/IP was originally developed by the Advanced Research Projects Agency (ARPA), which is the research arm of the Department of Defense, in the 1960’s and 1970’s. Its goals included flexibility (to handle applications from file transfer to voice over IP), redundancy (allowing for multiple routes between two sites, so that communication didn’t depend on a particular path through the inter-network being available) and decentralization (so that the destruction of one machine couldn’t bring down the network). TCP/IP is very good at adapting to changes in the network structure, since it is packet-switched rather than circuit-switched. For example, if you are sending packets from site A to site D, and in between the packets pass through sites B and C, if site C goes down TCP/IP detects the problem and finds another route to D, and the conversation continues normally. The applications communicating with each other don’t even need to know that there was a problem. This packet-switching approach is part of what makes TCP/IP a viable protocol suite for decentralized networks like the Internet – since administrators of Internet sites control only their own networks, the Internet needs to be smart enough to maintain communications even if some intermediate site disappears in the middle of a conversation.

At the time TCP/IP was designed, ordinary users of machines on TCP/IP networks were expected to not possess elevated user privileges, such as Administrator or root. Most TCP/IP protocol details were implemented inside the operating system and this model meant that unprivileged users would never have the chance to interfere with packet structures or sequencing. The designers appear to have figured, “If you have physical access to a network device, or a privileged account on it, it must be OK for you to do whatever you wish with it.” This led to network nodes trusting the packets they received, despite the lack of any failsafe way to guarantee that packets really did originate from where they claimed to originate from, adhered to the protocol specs, and had not been tampered with in between. This blind trust caused a few problems when TCP/IP began to be used outside of trusted networks.

Because TCP is inherently a 2-way communication protocol (as opposed to UDP, which is a "fire and forget" protocol), there has to be a way to identify where the data is coming from and where replies should be sent. This is called the "source IP" and is held inside every TCP packet's header (a packet header is a section of information at the start of every packet sent across a network). Under normal circumstances, a user would load an application such as a web browser and type in the address of the website they wished to view. The network stack in their machine would create the appropriate packet (in this example, an HTTP request packet), add the packet header including the machine's IP address, and send it off across the network.

However, if the user has sufficient privileges on their machine, it is possible for them to change the packet header before the packet is sent. This allows them to change the source IP address in the packet header, thus "spoofing" their IP by making the packet appear to originate from a different source.

In the example of a user browsing a website, this technique is fairly pointless. However, if this technique is used in conjunction with a Denial Of Service attack (as discussed in 1.4.1), an attacker can use this to their advantage. First, they can hide their true source IP address, which, in today's world of lawsuits against hackers is a fairly strong motivation, making it difficult to trace them. Secondly, they can continually change the source IP of the DOS packets, making it hard for network administrators to drop the packets at firewalls using source IP filters. To take it a stage further, an attacker could spoof the source IP of the DOS packets to make them appear to originate from within the target's network. While this technique may not always work, it can give poorly configured firewalls and routers a lot of difficulties.

A second use of IP spoofing is to circumvent trusted host configurations. Kevin Mitnick publicized this as a technique he used to break into a bank's transaction system. Take the following as an example. A company has 2 systems to control its online automated purchasing service. System 1, let's call it Freedom, controls the stock and picking system. System 2, let's call it Spirit, controls the banking credit and debit system. Whenever an order or goods return request is placed, an application on Freedom reduces or increases the stock count as appropriate, and sends a purchase/refund request to Spirit. Spirit then connects to the bank and completes the transaction.

Because both Freedom and Spirit sit in the company's backend network, the inexperienced system administrator believes that it would be safe to configure them with a trusted host system only. In other words, he configured Spirit so that it would only ever accept connections from Freedom, because that's the only host that should ever talk to it. Both systems are, however, completely secured and up to date with patches etc.

So if the systems are secure, how can an attacker use this configuration to their advantage? The answer lies in IP spoofing. While the attacker can't actually break into either Freedom or Spirit, he can control Spirit's behavior by manually creating packets with Freedom's IP address as the source. If the attacker crafts a packet containing data to make a transaction of £1million into a bank account, then sets the source IP address to Freedom's IP address, when Spirit receives this packet it will check the source IP, see that it matches Freedom's IP, process it as normal and the fake transaction will go through.

Submitted by Ian Kayne same email as 1.2.5 revision

Page 22, Footnote 17 is missing a link.

The rootkit is at www.rootkits.com, or

http://www.megasecurity.org/Tools/Nt_rootkit_all.html

While this is a slightly wild example, the theory is valid. Because the IP stack does not provide any measures for verifying the source IP address, systems that do not employ other measures are vulnerable to this type of spoofing attack.

It should be noted that spoofing is broken down into 2 categories – normal spoofing, and "blind" spoofing, denoting the type of control an attacker has. Normal spoofing is the easier of the 2 to control. This type of attack is a combination of IP spoofing and packet sniffing (see 2.5.4.1). Because the attacker is faking the source IP address of the packets he is sending, the responses from the target machine will obviously be directed to that source IP, and not the attackers "true" IP. This means that for the attacker to see the responses of the machine he is sending these spoofed packets to, he must sniff the network and use a packet capture & decoding tool to read the responses of the machine. Taking the Freedom/Spirit example again, when the attacker sends spoofed packets to Spirit with Freedom's IP address as the source, Spirit will send its replies back to Freedom.

To read these replies, the attacker must sniff the network and decode the packets as they are sent. To do this, the attacker must be able to place a network card on the same network segment as the hosts into promiscuous mode. However, tools such as Antisniff [18] are able to detect this.

Blind spoofing removes the requirement for sniffing the network, and operates on a "best guess" principle. The attacker sends spoofed packets to the target as before, but instead of sniffing the network and reading the replies, he just guesses at what the replies will be in hope that when he has completed his attack, the system will have performed the actions he requested. The advantage here is that packets can be sent from any network that has a route to the target and there is no requirement for sniffing the packets on the target network. It does of course make the attack harder to perform because if it fails the attacker has no way of diagnosing what went wrong.

At this point you should note that there are legitimate uses for changing a packet's source IP address. The most obvious one is NAT, or Network Address Translation, where a device, such as a router, deliberately and legitimately rewrites packet headers. See 3.3.3 for more information.

Unfortunately, the problems spoofing presents do not end here.

1.4.4 Man in the Middle

The Man In The Middle attack, or MITM for short, is another attack made possible by lax security in the IP protocol. As we already know, a normal TCP/IP conversation takes place between 2 hosts, and involves the sending, receiving and acknowledgment of packets. An MITM attack can be compared to inserting a black box in between the 2 hosts participating in the conversation.

If an attacker can place himself in a position where he is on the network between the 2 hosts, it is technically possible for the attacker to control what data is sent between the hosts. Going back to the Freedom and Spirit example above.

Assume now that Freedom and Spirit communicate with each other, but instead of on a trusted backend network, it is across a public network. First of all, an attacker must ensure that packets sent between the 2 hosts all pass through their machine.

[18] http://www.securitysoftware.com/antisniff/download.html

This may mean that the attacker sets up a machine with 2 network cards, then sends out fake ARP requests to force packet routing to go via his machine. He then takes the incoming packets from Freedom, decodes them and checks the contents. If he sees something he wants, for example a bank transaction he would like to divert, he can rewrite the data in the packet, and send it off to Spirit. By the same token, when packets come in from Spirit destined for Freedom, he can change the contents of those. He can also pass packets through unchanged, if he wishes.

The effect is somewhat like the spoofing example presented above, however, this is far harder to detect. While the spoofing attack relies on sending brand new packets to a host, the MITM attack is actually changing data sent between hosts on the fly. Therefore, should you ever audit the data to find out why the bank transactions seem to have gone awry, you'll see that data was sent by Freedom and received by Spirit in correct order, it just "got changed somehow". In addition, because the MITM attack manipulates packets already created and sent, you are removing the "luck" factor from a spoofing attack. For example, if you have correctly set up the MITM attack you can be assured, that provided you stay within the parameters of the data, any information you change will be readily acceptable by the destination host. However with a spoof attack, you are effectively working blind, as there is no guarantee that the target host will accept your data.

At this point, it may seem as though all hope is lost, and that no communication on a public network is safe. After all, wouldn't every so-called hacker be doing this? Well, fortunately not. As in practice, this technique is very difficult to implement. First of all, we have the physical routing issue. As noted above, for this attack to work, an attacker must ensure that every single packet sent between the 2 hosts he is attacking is routed via his machine. Taking the Internet as an example, this is a huge technical issue. First of all you would need to be physically close to the target machines – no matter how hard you try, packets sent between 2 servers in London will not be easily diverted and routed via China.

Secondly, an attacker has TCP Sequence numbers to contend with. In a nutshell, every TCP/IP connection negotiates a TCP Sequence between both hosts. Subsequently, every TCP packet sent between them has a TCP Sequence number included in the packet header. This number is changed for every packet by a prearranged formula, decided on during the TCP handshake stage.

This allows both hosts to ensure they are receiving all the packets in a TCP conversation, and to ensure that the packets are being assembled in the correct order. In other words, the TCP Sequence number is responsible for the quality control of the protocol. If the sequence number of a packet is wildly out of sequence or just plain wrong, the packet is discarded (with a few additional checks). If an attacker is unable to break the TCP Sequence formula, they won't be able to initiate an MITM attack. Tools such as Nmap[19] have options to check the TCP Sequence formula of the IP stack on a machine and inform you how difficult it would be to "break" it. This is called TCP Sequence Prediction.

Thirdly, speed is a major issue. Consider the speed with which packets are sent across a network, even a slow 10megabit one. While applications can quite happily send and receive data at this speed, it poses more of a problem for humans. To perform an MITM attack and change data on the fly would be practically close to impossible. The only practical way to do it would be to write a custom application that searched for specific strings in packets and manipulated them to preset rules. This does of course introduce an element for error – one wrongly interpreted packet could cause a change that triggers IDS alarms, and gets an attacker caught.

One area where MITM attacks do stand out is in hacking standard telnet user-type connections. Because you're dealing with a human user on one side, the speed of data transfer will be easily manageable. With a plain-text type connection, such as telnet, you'll be able to see the user's passwords and private information exactly as they access it. This is an excellent reason not to employ plain-text authentication and communication on your trusted network.

Encryption and secure protocols are one final area to be mentioned here. An MITM attack relies on being able to read the data contained in the IP packets. If an attacker can't read the data, the attack is useless. Using secure protocols on your network will reduce the consequences of an MITM attack to almost zero. Currently there is no available hardware that is able to break 3DES-encrypted data quick enough to make an MITM attack impossible.

Bearing all this in mind, applications do still exist to perform MITM attacks. An example is SW-MITM[20]

[19] www.insecure.com
[20] [0] http://packetstorm.acm.miami.edu/UNIX/misc/sw-mitm.tar.gz[0]

It should be noted that the vast majority of tools to perform low-level IP hacks are only available on Unix and the like. The reason for this is simply control. Under Unix, the IP stack is completely open for manipulation by anyone with sufficient privileges. However, under Windows this is not the case. In every version of Windows, except XP, the network stack has been kept "private" to the operating system and low-level hacks, such as spoofing, have not been possible. To combat this, a few organizations have coded ports of network libraries such as Eeye's LibNetNT, which do allow for low-level manipulation. Windows XP, however, contains "raw sockets" which allow for this manipulation natively. This issue has been massively debated and demands for its removal have been championed by Steve Gibson21, who believes this to be extremely dangerous in functionality, despite it existing well before Windows XP was even dreamt of.

1.4.5 Replay

Replay attacks are exactly as their name suggests. An attacker uses a packet capture tool to grab and save the packets in a conversation, then resends one side of the conversation to a host as he wishes. In the example of Freedom and Spirit, an attacker may capture a conversation between the machines in which a bank transaction is requested. The attacker may then resend all of Freedom's packets to Spirit multiple times, causing Spirit to process the same transaction many times over.

Again, this type of attack is made possible due to lax security in the protocol. As we have already discussed, every TCP/IP conversation is stamped with TCP Sequence numbers. In theory, no two conversations will generate the same set of numbers due to differences in the seed used and client/server TCP/IP stack differences, amongst other factors. In addition, many operating systems implement pseudo-random number generation for the initial sequence number. It has been proved however than in many implementations, the formula for producing these numbers is weak, and TCP sequence prediction is not as difficult as could be. If an attacker is able to guess the sequence (or indeed flood the network with packets containing likely sequence numbers), he may be able to control, disconnect or insert arbitrary data into a user's session.

[21] www.grc.com

There are countermeasures against this type of attack. Secure protocols such as IPSec operate at the network layer and extend the IP protocol while maintaining compatibility with existing TCP/IP stacks by using additional protocol numbers. The ESP (Encapsulating Security Protocol) portion of IPSec handles most of the security of the protocol, such as authentication, data protection and anti-replay measures. IPSec relies heavily on cryptography and IKE (Internet Key Exchange) to create session keys, which is outside the scope of this section. Cryptography is discussed in section 4.

1.4.6 TCP/IP Hijacking

Hijacking is the attacker's action to forcefully take control of a legitimate conversation between a server and a client. It's generally the result of a successful attack using a different technique, such as a replay or MITM attack, allowing the attacker to impersonate a legitimate user. The most common example of this is web session hijacking, where an attacker takes advantage of lax security to take control of a user's browser session. Most web servers use cookies, small data files accessible by server scripting languages such as PHP or ASP, to authenticate and track users. When a user connects to a website and authenticates, an attacker may be able to hijack their session by loading a hacked cookie or by inputting a specific URL on a poorly configured web server into their browser. The legitimate user will most likely be kicked out, or at least be shown an error page indicating their login has failed.

Another opportunity for session hijacking is poorly configured server time-outs. If a web developer makes the session time-out (length of time of no activity before the web server disconnects the user) too long, it provides a larger window of opportunity for an attacker to hijack the session.

Hijacking is not limited to web based sessions only. Like an MITM attack, hijacking is especially suited to telnet type plaintext connections, where an attacker can watch a TCP session being initiated and data being passed between client and server. If the attacker sees something interesting they can break into the conversation and take control of the user's session for their own purposes.

There are several precautions you can take against this type of attack. The most simple is to re-authenticate the user before performing important actions. For web servers, creating unique session cookies also mitigates the risk somewhat.

The more unique the cookie, the harder it is to break and hijack. Finally, if possible, use secure protocols and the employ the same countermeasures used against replay attacks. An excellent resource for further study is available in the footnote[22].

1.4.7 Weak Keys

The term “weak keys” relates directly to secure protocols, such as IPSec or SSL. As mentioned in 1.4.5, these secure protocols use cryptography and IKE to create a secure connection between 2 hosts using session keys. As discussed in section 4, a host must “decide” on a mathematical algorithm to use for encryption of secure sessions, such as DES, 3DES or RC4. 40-bit key lengths, and indeed 56bit DES are not considered secure, and modern hardware is allowing attackers (and researchers) to break these encryption algorithms quicker all the time. Unfortunately due to the legacy from old export laws from the USA restricting the export of strong encryption, a significant proportion of servers (especially web servers) still use these weak keys.

Much of the rest of this domain overlaps with Domain 4 Cryptography. It appears to be definitional in Domain 1, and more How based in Domain 4.

1.4.8 Mathematical

The short story is when using a key set that has few combinations it is more apt being a weak key. See 1.4.11.1 . Popular belief states that 40-bit encryption is weak, and 128-bit is strong. This is only a general statement23, as you will see when you look at WEP in 802.11.

Even 256-bit encryption is pretty useless if the key is has a simple pattern such as 1,2,3,4,5,6 ....

[22] http://blinky-lights.org/script.html
[23] http://www.counterpane.com/pitfalls.html

1.4.9 Social Engineering

(Also Refer to 5.1.2)

In the introduction you leaned that Social Engineering is the biggest hole we face. The best cryptography in the world is useless if someone is fooled into giving away the keys to the city. Lurking on the Internet, we have learned that the best 'black hats' rely on exploiting human nature more than any technical exploit.

Typical exploits (see also 1.4.12) include:

No password for the Administrator (god) account
Easy or predictable passwords (the name “password” for a password)
Shared passwords
Lax or obsolete shares on files and folders

1.4.10 Birthday

People try to simply life by creating passwords that are easy to remember. A popular method is to use a birthday, which may be your own or a famous date in history, such as, 4July1776. This makes it easy to break the password either with a guessing program or brute force.

1.4.11 Password Guessing

A 'black hat' will have in their toolkit several password guessing programs. The most commonly used is the dictionary attack (See 1.4.11.2)

1.4.11.1 Brute Force

Brute Force, even just a couple years ago, was considered difficult due to the lack of lost cost processing capable of the sheer crunching power needed. Today, the AMD 2000XP processor is < $100 USD and the 2400XP is shipping. That puts the brute force method within reach of anyone. Rather than go on with the usual blah-blah about strong passwords, we are encouraging you to follow the footnote to a free Brute-Force Password Cracking Simulator[24].

[24] http://www.alpinesnow.com/bpcs.shtml

The program is less than one Megabyte in size and runs in Windows. Instead of actually attempting to beat up a stored password, you just set the variables (including testing a real password) and it will calculate how long the brute force method takes. In one test Brute Force with a 1.5Ghtz processor would take 170 years, 309 days, 21 hours, 32 minutes, and 22 seconds to crack 4July1776. However, a dictionary password program would rip that same password almost instantly.

1.4.11.2 Dictionary

Currently in the *nix world, dictionary cracking is the most popular exploit. The FAQ of this work states that this is not a treasure map for folks who want to do damage. We have included the footnote to John the Ripper[25] because the download site appears right at the top of Google; so, we are not giving anything away here. John The Ripper works on both DOS and UNIX based systems and a CD can be ordered that contains 20 different languages. It is a whopping $15 USD airmailed to any address in the world.

1.4.12 Software Exploitation

In short, software exploitation is either the use of a buffer overflow or a social engineering issue. Social engineering has been touched several times; so, it is time to look at a simple definition of a buffer overflow.

The most famous buffer overflow occurred on November 2, 1998. The “Morris worm” caused incredible damage by exploiting a buffer overflow in the UNIX program fingered (a daemon AKA server service) for the finger utility.

Generically speaking a buffer overflow occurs when an input string is used to send more data to a buffer than it was intended to handle. Software coders are famous for checking to see if something works and being blind to what happens if. The footnote has a great overview[26]. The most recent PIA virus in the buffer overflow category is Code Red[27].

While many would consider the use of a buffer overflow malicious code, it appears the SME (Subject Matter Experts) for Security+ had a different idea. They specifically break out malicious code in the next sub-domain.

---

[25] http://www.openwall.com/john/
[26] http://www.enseirb.fr/~glaume/bof/report.html
[27] http://www.cert.org/advisories/CA-2001-19.html

1.5 Malicious Code

As pointed out in the preceding section the SMEs have a different breakdown on Malicious Code than what appears to me (Tcat) to be the Common Body of Knowledge. Whatever.

The objectives are specifically calling Malicious Code software written to do damage, as the sub-domains reveal. I guess there is a difference between writing code that replicates itself and Malicious Code that was merely written to do damage.

1.5.1 Viruses

This is a catch-all phrase for any man-made code that is anything from annoying (WAZZU[28]) to destructive, such as boot sector viruses[29]

1.5.2 Trojan Horses

This one should tie tightly (at least in your mind) to social engineering. This pretends to be something you want, yet, is really a virus. Typically the Trojan relies on gaining users' interests with something that sounds fun or curious. Sometimes they masquerade as some sort of data file where a user is asking for help. (Advice, if you don't know the name of the person it came from, delete without opening. It may contain a zero day[30] virus.

1.5.3 Logic Bombs

A logic bomb is a virus of some ilk with a time delay fuse. The most famous virus with a fuse was Michelangelo in 1992. More practically speaking, a really upset employee is more of a danger.

1.5.3 Worms

This was completely missing in the first draft. To correct my mistake (I own it, Tcat) I have put a link to a definition.
http://www.webopedia.com/TERM/w/worm.html

http://www.itworld.com/nl/unix_sec/09132001/

---

[28] http://open.jeffersonhospital.org/tju/dis/virus/desc/wazzu.html
(I wonder if a U of W <Huskie> wrote this?)

[29] http://www.sophos.com/virusinfo/analyses/index_dosexe.html.

[30] Zero day viruses. Not yet known to be in the wild, and you're the lucky person to "discover" it.

1.6 Social Engineering

Again! We are not clear what is going on in the minds of the SMEs. It appears here you need to know the definition of what social engineering is. Well if you have been taking this work in a lineal fashion, we won't bore you here. If you are skipping around, read the introduction section.

1.7 Auditing

There are several sections in the released beta objectives that ask about auditing. If we are correct that this domain is about definitions then auditing is creating logs where you:

A. Establish a baseline of “normal” activity.
B. Then Monitor against the baseline for
C. Abnormal results.

A point should be made here as to what and where to monitor. You cannot monitor everything in a typical environment (such as a completion of a print job) because you become flooded in data. The chairman of IBM has in his office the motto: “Think”. Monitor the obvious. Anything less than obvious to monitor is outside any firewall. Create a baseline of “normal” activity outside and monitor so you have an idea of what an attack looks like *before someone gets in.*

Chapter 0010

Communication Security (20%)

2.0 Communication Security

Communication security involves the security of data and administrative information that are traveling on public and private networks. It is generally concerned with the privacy (can unauthorized users read it?) and integrity (can it be changed undetectably before it reaches its destination) of network traffic.

2.1 Remote Access

Remote access involves giving users outside the bounds of your physical network (i.e. “remote users”), access to network resources, usually, by permitting them to join the network as an actual node.

A special case of remote access is the Remote Access Service (RAS), provided by Windows, which allows a server connected to one or more modems to be used as a network access point by dial-in users.

Communication security concerns are generally addressed by use of standardized authentication and authorization mechanisms (such as RADIUS, TACACS+ or a VPN’s proprietary user authentication/authorization scheme), and by encryption of traffic on the wire as protection against snooping and modification of data.

2.1.1 802.1x

When enabling WEP (Wired Equivalent Protection) the encryption key must be the same among all the devices. As mentioned elsewhere, software such as AirSnort and WEPcrack can compromise WEP.

802.1x[31] uses the Extensible Authentication Protocol (EAP) to dynamically vary the encryption used by WEP. EAP comes in different flavors such as:

Transport Layer Security (EAP-TLS)

EAP Tunneled Transport Layer Security (EAP-TTLS) Built into XP, Win CE 4.0

RADIUS (FRC 2138,2139)

LEAP[32] Cisco

2.1.2 VPN

VPNs, Virtual Private Networks, are secure "virtual" networks built atop physically connected networks. Generally, the physically connected network over which a VPN is implemented is a public network – that is, one which is generally accessible and has less security than the organization implementing the VPN desires. It is also possible for an organization to implement a VPN over a private network to provide an additional level of confidentiality for its most sensitive communications.

Each node participating in a VPN is (or is connected to) an endpoint that knows how to wrap the virtual network's traffic (which can be TCP/IP, Netware's IPX/SPX, AppleTalk, etc.) in packets understood by the public network carrying its traffic (usually TCP/IP), and then, unwrap the packets upon receipt. This wrapping/unwrapping process is known as tunneling, since it takes what is normally a data link layer protocol such as IP, and wraps it within a "tunnel" of an outer protocol instead of placing it directly on the wire. Tunneling may be accomplished by a special hardware box that speaks the VPN's protocol, or, by client software installed on computers that are individual nodes.

---

[31] www.drizzle.com/~aboba/IEEE/ 11-02-TBDr0-I-Pre-Authentication.doc

[32] http://www.cisco.com/global/AT/veranstaltungen_seminare/downloads/files/03_wlan_security.pdf

VPN's usually utilize user authentication (by means of certificates, user/password, etc.), and traffic encryption to create a private network. Common protocols for VPN encryption include PPTP, L2TP, SSH and IPsec. For more information, see the VPN topic later in this work.

2.1.3 RADIUS

Remote Authentication Dial In User Service, or RADIUS, is the de-facto standard client/server user authentication protocol that authenticates and authorizes users connecting to a network, to access the network's resources. You can think of it as protecting the "radius" of a network by not letting in those who are unauthorized to be there. Its client/server architecture allows centralized administration of a user database, even if users' locations may span an entire organization, town, state, country, etc. Being the de-facto standard, as specified in RFC 2865, the RADIUS protocol is supported by just about every device out there, new and legacy.

In general, the way RADIUS based authentication works is:

A. A user dials in (via modem, DSL, etc.) as a client to a remote access server, and provides credentials (user/password) in response to the remote access server's request
B. The remote access server (itself a client to a RADIUS server) communicates the credentials to the RADIUS server, after encrypting it by computing an MD5 hash of it using a "secret" shared between client and server (this is an example of one way in which credentials are communicated)
C. The RADIUS server uses a user/password database or perhaps integration with a network-based authentication system like Windows passwords or LDAP to validate the password, and returns the results to the remote access server
D. The remote access server then accepts or denies the connection

More info on how RADIUS works can be found in the footnote[33].

It is regarded by many as providing more security during remote access user authentication than its main competitors, LDAP and TACACS+.[34]

[33] http://www.cisco.com/warp/public/707/32.html
[34] Hill, Joshua, "An Analysis of the RADIUS Authentication Protocol," http://www.untruth.org/~josh/security/radius/radius-auth.html

Recent advancements have included “Distributed RADIUS” in which multiple tiers of RADIUS servers are connected together; and, forward authentication through which requests go up the RADIUS server tree via a proxy RADIUS protocol.

2.1.4 TACACS/+

TACACS is the Terminal Access Controller Access Control System, another client/server user authentication protocol similar to RADIUS, which works similarly to RADIUS. For authentication, it allows use of user/password information, Kerberos-style authentication that does not require keys being passed over the wire, or even dynamic password systems in which smart cards are used to generate one-time passwords.

Over the years, three generations of TACACS have been developed:

A. TACACS, the original, which performs authentication and authorization.
B. XTACACS, or Extended TACACS, which separates the tasks of authentication, authorization and accounting/logging.
C. TACACS+, which builds on XTACACS by adding a two-factor user authentication (proving that a user is who they say they are through both something they know, like a password, and something they have, like a smart card), system and encrypting all client/server communication.

TACACS+ has some security vulnerabilities that may concern you if end-users have access to the network over which TACACS+ traffic travels:

A. Since accounting information is sent in clear text, and, the only verification performed is that the received accounting record packet length = the length that was sent, someone could intercept the communication and alter it or inject spurious accounting records.
B. Encryption is potentially vulnerable due to the small size of the session id key used for encryption.
C. Lengths of user passwords can be determined by watching traffic, because, the protocol provides for sending a password only as long as there are characters in the password.
D. Theoretical issues with MD5 hashes (see Solar Designer’s paper for more details).

E. A handful of overflow/resource hogging vulnerabilities in some popular implementations of the protocol, which can lead to denial of service[35].

2.1.5 L2TP/PPTP

L2TP (Layer 2 Tunneling Protocol) and PPTP (Point to Point Tunneling Protocol) are both Layer 2 tunneling technologies.

PPTP is probably the most popular tunneling protocol today. It was developed by a consortium inclusive of Microsoft, Ascend Communications, US Robotics and ECI Telematics. Over the years it has gained prominence because of its use for remote access in Microsoft-based network environments. It implements tunneling over a PPP (usually dial-up) connection. Typically users choose the VPN endpoint to which they are connecting after the PPP connection negotiation has completed, a situation that is known as voluntary tunneling.

Microsoft's PPTP implementation uses its RAS "shared-secret" encryption process with an RSA RC4 cipher based on a 40 or 128-bit session key. In the Microsoft implementation, the shared secret is the user password. In other implementations, the shared secret might be a public key (see PKI, later in this work).[36] PPTP typically lets you use any authentication mechanism, including PAP and CHAP, but if you want to use an encrypted tunnel, it requires that you use the more secure MS-CHAP authentication mechanism. PPTP uses TCP port 1723 for communication with the destination host, so if, you are passing PPTP communication through a firewall make sure that that port is open.

L2TP was intended as a replacement for PPTP by Cisco because they didn't care for some (rather lack of) features in PPTP. It combines features from PPTP and Cisco's L2F protocol, which was designed to facilitate tunneling over a variety of media/lower-level protocols such as frame relay and ATM, in addition to the IP-based tunneling supported by PPTP. As opposed to PPTP, whose client access is normally implemented by software running on individual desktops, L2TP clients most commonly connect into their VPN by going through a special hardware device that handles the L2TP tunneling.

---

[35] Solar Designer, "An Analysis of the TACACS+ Protocol and its Implementations," BugTraq mailing list, http://online.securityfocus.com/archive/1/62742

[36] "Understanding Point-to-Point Tunneling Protocol (PPTP)," Microsoft Corporation, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebtool/html/understanding_pptp.asp

Windows 2000 is quite capable of supporting L2TP natively and some firms don't want to consume the recourses of the server, so they handle it outside of W2K. Additionally, with L2TP, the server side generally chooses the endpoint of the communication, a situation that is known as compulsory tunneling. This scenario lends itself to the construction of hierarchically routed networks which gradually concentrate VPN traffic over fewer but higher bandwidth lines for more efficient transmission over a long haul.[37]

L2TP can use PAP, CHAP, MS-CHAP and other authentication mechanisms. IPSec is the preferred encryption mechanism used in conjunction with L2TP, but sometimes 40 or 56-bit DES may be used as well. L2TP uses UDP port 1701 for communication on the source and destination hosts, so if you are passing L2TP through a firewall, make sure that port is open.

2.1.6 SSH

SSH, or Secure Shell, began as a replacement for traditionally insecure methods of accessing a UNIX host.

"In the beginning," there was telnet. And the geeks connected to other geeks' systems over an academic Internet via telnet, providing user name and password to authenticate themselves; and, it was good! Then more geeks out on the West Coast at Berkeley said, "If we connect to a certain system all the time, and its administrators trust us, why should we always have to keep typing our passwords in?" By the way, it's unsecure to transmit passwords in clear-text across the Internet the way telnet does, because anyone who can 'sniff' the bits on the network will be able to discover your passwords. "So let's create another remote terminal access program which optionally uses our user and source host name to automatically authenticate us instead of requiring us to type in our user name and password?" and thus came up with another remote host access method, known as rlogin (or the infamous "Berkeley r- commands"). Alas, both of these mechanisms have flaws – both transmit the password, if used, in clear-text, so that it's vulnerable to being sniffed. The r-commands have the added flaw of relying on DNS information, which can be spoofed by attackers, for authentication.

---

[37] "Layer 2 Tunnel Protocol", Cisco Systems, http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t1/l2tpt.htm

SSH was developed as an answer to these issues largely replacing telnet, rlogin (and other r- commands such as rexec and rcp, a remote file copy utility), and adding the capability of forwarding secure X Window System connections (X is the underlying GUI used on most UNIX systems). It is implemented on servers through the use of the ssh daemon which listens for incoming client connections on TCP port 22.

Then someone looked at the technology in SSH, and decided it would make a good low-cost, general-purpose VPN protocol. Today SSH can be used over PPP, to create a VPN at a higher OSI layer, redirecting TCP/IP ports to allow encrypted services, and proxy X Window System traffic. The SSH 1.x protocol permits secure authentication by way of RSA key exchange between client host and server or individual user to server. The client host and user keys are normally 1K in length, and the SSH server key 768 bits. The SSH 2.x protocol, used freely by available clients and servers avoids the use of the then-patented RSA algorithms, opting instead for the DH and DSA algorithms. SSH supports a wide variety of encryption options, including RC4, 3DES, Blowfish and AES-256, to ensure data integrity and privacy. (Note that all SSH implementations don't support all possible encryption options. For example, OpenSSH[38] steers clear of patented algorithms.)

2.1.7 IPSEC

IPsec, or IP Security, is a set of standard protocols developed by the IETF (Internet Engineering Task Force) that supports the secure exchange of packets at the IP (network) layer. It is the most popular layer 3 tunneling approach for VPN's. Unlike PPP, it supports only IP – which today is not the drawback it might have been several years ago when fewer organizations' backbones were IP-based.

IPsec packets include the authentication header (AH) for data integrity and the encapsulating security payload (EPS) for data confidentiality and integrity. IPsec supports Transport and Tunnel modes for encryption. The "Transport" mode encrypts only the data portion of the encapsulated packet, while the "Tunnel" mode encrypts both the data and the header portions of the encapsulated packet hiding more information about the underlying communications.

[38] Open SSH, http://www.openssh.org

IPsec uses public key encryption technology. That is, the sending and receiving devices share a public key with the server who has a secret "private" key. Key management and security associations are handled by the IKE (Internet Key Exchange, previously known as ISAKMP/Oakely) protocol, which provides for key exchange and authentication, and uses digital certificates to allow its authentication to scale to the Internet.[39]

Specific encryption technologies used by IPsec include:

Diffie-Hellman key exchange between peers on a public network;

Public key cryptography for signing the Diffie-Hellman exchanges to guard against identity-spoofing and man-in-the-middle attacks;

Standard algorithms such as DES for data encryption;

Keyed (HMAC) and non-keyed (MD5, SHA) hashing for packet authentication;

Signed digital certificates used to provide proof of identity[40].

Unlike PPTP, IPsec's TCP control packets are authenticated, so DOS attacks that depend on the use of TCP control messages sent by attackers to which PPTP-based networks are vulnerable do not affect IPsec based networks.

One thing to be aware of when using IPsec is that interoperability among different vendors' implementations of the protocols is an on-going effort. As more and more standards are finalized, successful interoperability is becoming more the rule than the exception. But this is not always the case, particularly if you are integrating new hardware or software into an existing "IPsec" based network, test the new equipment on your network to make sure that it works well with your existing configuration.

2.1.8 Vulnerabilities

Vulnerabilities are mostly related to the ability to "sniff" passwords and data off the wire, and to spoof user identities. These are addressed by increasingly sophisticated authentication mechanisms, which use certificate-based or Challenge-Response technology, rather than, requiring plain-text or encrypted transmission of authentication data through the use of increasingly complex ciphers.

---

[39] "White Paper – IPsec Executive Summary", Cisco Systems, http://www.cisco.com/warp/public/cc/so/neso/sqso/eqso/ipsec_wp.htm

[40] "White Paper – IPsec Executive Summary", Cisco Systems, http://www.cisco.com/warp/public/cc/so/neso/sqso/eqso/ipsec_wp.htm

There are also occasional implementation vulnerabilities in the code for devices and software that implement remote access, due to programmer error. These vulnerabilities are often exploited to cause a Denial of Service, by crashing the client or server. As with all other functionality implemented on computer systems, new bugs are being discovered in communication software all the time. Somewhat complicating things is that some communication code is produced as open source and often finds its way into multiple vendors' implementations of a particular standard protocol. This means that an implementation flaw in a protocol that is used by numerous manufacturers can affect numerous devices sold by numerous vendors.

2.2 Email

Email, or electronic mail, involves the transmission of messages over a network. An email system consists of back-end storage for email files, programs that allows users to access stored email files ("read" email), send email, and programs that relay email back and forth among email servers (called "relaying") and clients.

Some email systems use proprietary communication protocols specific to a certain email system, and some use standardized protocols. In the Internet world, the two most common email-related protocols are SMTP (Simple Mail Transport Protocol, via TCP port 25) for sending mail, and POP3 (Post Office Protocol v3, via TCP port 110) used by clients to retrieve incoming email from its storage location on a server. A newer protocol providing a superset of the POP3 functionality allowing access to a hierarchy of mail storage folders, is IMAP (Internet Mail Application Protocol, via TCP port 143).

Internet email consists of an envelope of headers (of the form Header name: value), each on a separate line, which are interpreted by email servers and email clients, followed by a content area containing the actual message sent. Typical headers include "To:", "From:", "Message-ID:" (the serial number for the message), "Reply-To:" (if the reply address is different from the sender's address), "Subject:", and "Received:" (which is very handy when attempting to trace the source of Spam email, discussed in a later section).

A standard used for email content encoding on the Internet, is MIME (or Multipurpose Internet Mail Extensions). Email was originally used solely to transmit text; and, the SMTP protocol used for email transmission was text-based as well. But then a funny thing happened… people realized email would be useful for sending things other than text, such as pictures or even programs. Enterprising techies came up with the idea of encoding binary objects as a series of alphanumeric characters so that they could be sent through email and decoded and viewed/used by the receiver. A number of standards for encoding email content were used over the years, but MIME was the one which took hold because of its flexible approach of using additional headers to specify the type of content included in each binary data segment. These headers are, in turn, interpreted by email client software, which determines which program to use to display or save the object, based on the object's type.

It's those MIME headers, in conjunction with the settings in your email program, which let your email program know to open faxes with fax-viewer software, Excel documents with Excel, .doc files with Word, .mp3 files with your favorite player, etc.

2.2.1 S/MIME

S/MIME, or Secure/MIME, provides sender authentication and message privacy for email. It was developed by RSA Security[41]), and uses standardized formats for message data and digital certifications -- the PKCS #7 data format for messages, and X.509v3 format for digital certificates used for authentication. S/MIME is a scalable secure email solution in which the standard hierarchies used in managing X.509v3 digital certificates are used to handle the chores of public key exchange and certificate authentication. For symmetric encryption, S/MIME can use 3DES, DES and RC2 algorithms.[42]

2.2.2 PGP

PGP, or Pretty Good Privacy, provides much the same functionality as S/MIME, but with message data digital certificate formats designed from the ground up, rather than being based on existing standards. By implementing PGP without relying on controlled/patented algorithms, so that it could be distributed anywhere without license fees or patent issues, the developers hoped that the idea of using privacy-enhanced email would really catch on.

As in S/MIME, 3DES is used for symmetric encryption of message data, and SHA-1 for hashing. Unlike with S/MIME, individual users are responsible for exchanging their public keys with each other and deciding that they trust the public key they received as being proof of the other party's identity before messages can be sent.

Back in the days when encryption code was considered munitions and eligible for export only under certain circumstances (read: only when the US government possessed the means to easily defeat it), the primary developer of PGP, Phil Zimmerman, got into a bit of legal trouble for exporting PGP code.

---

[41] http://www.rsasecurity.com
[42] "S/Mime Frequently Asked Questions", RSA Security, Inc., http://www.rsasecurity.com/standards/smime/faq.html

To show you how obscure some laws related to computer security can be, the same code in printed book form, courtesy of MIT, instead of on floppy or CD was ruled eligible for export (at least for a short time).

2.2.3 Vulnerabilities

Data privacy is a major issue in email, as with many other types of network communication. It isn't enough to ensure that your data is encrypted during transport and storage – rather, you should also ensure it is encrypted with a strong algorithm[43]. For I kid you not, a Windows screen saver which attempts to brute-force with S/MIME 40-bit RC2 keys, which are considered a weak encryption mechanism, if you intend more than casual protection from prying eyes. **[Where are we going here?]** Sometimes casual protection is enough, but often business requirements demand more security. Your organization would be advised to develop a policy for email security, specifying to users which levels of encryption, if any, are to be used for which types of communication, how long email is retained, who is permitted access to it, etc.

As Microsoft found out during the anti-trust trial, old email can come back to haunt you in a variety of ways. In Microsoft's case, internal memos mined from corporate email archives were presented as evidence by both sides. Did it help them or hurt them? It was probably a bit of both. Think about the potential for this when developing your retention policy.

There's no getting around it -- email has a disturbing tendency to hang around for longer than you expect, and to be seen by more people than you expect. Whether it's an email by a curious user to "BEDLAM DL3"[44], asking what the list (distributed to all of Microsoft.com) was for, personal discussions on a company server that ended up in a temp file seen by a system administrator, an email from one administrator to another about an intrusion that was just detected (which was read by a cracker who then knew to cover his tracks), or messages that should have been sent encrypted that were deleted the next morning by the receiver (but which still sit on backup tapes, protected in the corporate data vault, a year later), the contents just tend to be much less private than you think they are. When in doubt about whether you really "should" email something or not, just pick up the phone.

---

[43] http://www.counterpane.com/smime.html

[44] Winser, Michael, "Bedlam", UserLand discussion group archives, http://static.userland.com/userLandDiscussArchive/msg000484.html

Another email issue is forgery due to the lack of sender authentication in vanilla (as opposed to PGP or S/MIME) email. If you've gotten spam, you've probably seen forged email. By "forged", we mean email whose "From:" information along with other possible identifiable information, is deliberately incorrect. Email can be forged for a variety of reasons, such as not wanting replies (senders of "make money fast" pleas generally don't want tens of thousands of replies telling them to bug off, in their personal mailboxes), not wanting their identity to be known (in the case of someone who wants to communicate anonymously, just to protect their privacy), or wanting to pretend to be someone else (like the virus email which masquerades as an email from Microsoft Security). How do they do it? Sometimes, it's as simple as changing the "Name" field in their mail program, but leaving the email address as is. (Not very effective, but it is occasionally done, particularly when someone is using a "throwaway" Hotmail or ISP account to send a large volume of junk mail, and doesn't care how many replies the sending account receives.) At other times, users take advantage of a feature of the SMTP protocol – you can claim to be anyone, without having to prove to the mail server that you are that person, by manually issuing commands to the SMTP server (or using a program designed to issue these commands for you, supplying whatever identifying information you provide it).

Another issue with email-related communication is password security. Many implementations of the POP3 mail-retrieval protocol still require users to send their user name and password to the mail server in clear text (that is, unencoded, in a form that can be easily observed by someone monitoring network traffic). Clear text passwords are a bad thing, particularly because they often allow access to resources beyond a POP3 server – like an ISP's network, a UNIX host, a Windows user account, etc. Microsoft's answer to this was to create a "Secure Password Authentication" mechanism for POP3 connections to Exchange, which is best described as a version of the NT authentication process with a few enhancements. While it avoids clear text passwords, this "solution" seems to cause other security issues, at least when the Outlook Express client is used for an SPA connection.[45]

Other email issues are more on the client side than the server side, taking advantage of a bit of social engineering combined with shortcomings in the client program.

---

[45] 3APA3A@security.nnov.ru, "Outlook Express and SPA," http://www.security.nnov.ru/advisories/oespa.asp

For example, be warned that some email client programs when encountering an emailed file called something.doc.exe, will ask you if you'd like to open "something.doc" (ie, only listing the characters after the first period, not the second), and then proceed to gleefully launch a virus contained in the actual "something.doc.exe," file when you say OK.

And then, as always, there are implementation flaws. Both Sendmail and Microsoft Exchange are infamous for bugs affecting mail server security. Rather than go through a laundry list of bugs here, we'll just say that there have been problems in both which could result in an attacker gaining system administrator privileges. See your vendor for details – they'll have plenty of them. There is no excuse for not watching for security bulletins and keeping key software packages, particularly ones as widely used as those email servers, updated with the vendor's latest security fixes. If you don't, your mail server risks "death by script kiddie".

Still other vulnerabilities involve purely content issues -- undesirable messages sent via email, including Spam and hoaxes – which are discussed in the next section.

2.2.3.1 Spam

Spam is a registered trademark of Hormel, a US food company that has been gracious enough to not make a legal issue of its trademark being used for what is officially known as UCE or Unsolicited Commercial Email. (Thank you, Hormel.). UCE begins by 'harvesting' email addresses from web sites and selling millions of them for a fee to offer products or services. Based on my (Tcat) inbox, it appears that the number one purchaser of email addresses is the adult entertainment industry or phony products that offer to either improve my sex life or make me more attractive. I guess with an unusual first name it is difficult to determine if I am male or female so I get offers to make something bigger for both genders. Home mortgage offers seem to be leading over credit card offers, followed by some phone services. UCE comes in two types of email. They are plain text or HTML based.

Plain text is the easiest to dump. I use key words to filter them out to a folder to delete en mass. HTML is quite a bit sneakier. Because HTML email can tell a site I opened an email at a certain date and time, the sponsor knows they have a valid email address that they resell at a higher price. It doesn't matter that I deleted it, since the mere opening of the email even in preview mode triggers a validation that the email has been received and read, validating the address.

(Tip: Turn off that Outlook "preview pane" to keep Outlook from automatically opening these HTML-based Spam messages. Yes, it's an inconvenience when viewing desired mail. And more Spam, caused by Outlook reporting a valid address back to a Spammer's web server is quite possibly more of an inconvenience.)

There are several ways to approach UCE.

1.0 Filter and delete

1.1 Create a set of rules in an email server or client.

1.2 Purchase a commercial program to filter and delete.

1.3 Use a service such as Spam Cop (Http://www.spamcop.net)

2.0 Generally speaking DO NOT respond with a message telling them to remove you. This only confirms an email address.

If you are not going to follow the advice of #2, be prepared for the time and expense of using whatever state/providence law available to take the fight to the ISP level of the firm that bought in to UCE to have your domain blocked. This is the most costly route in terms of time/money/energy, and (for me) has sometimes been worth it. For starters, DO NOT believe the "From: " address, as it can be spoofed. The best hints as to the email message's origin are found in the "Received" header which almost always indicates the IP address or domain name of the mail server which was used to send the mail.

2.2.3.2 Hoaxes

Web hoaxes are similar to email hoaxes except they are posted at a web site. Some sites are just for fun and actually are real[46], while others are fiction. Remember anyone can host a web site.

Quoting from the web site "The legend grows as it goes". Hoaxes are "information" that sound credible and are passed on. The sheer volume of hoaxes is too numerous to even begin to discuss here. A popular example includes some government entity wanting to tax email[47].

[46] http://www.cheeseracing.org/
[47] http://www.tafkac.org/ulz/emailtax.html

2.3 Web

The Web, by virtue of being a widely used Internet service, generates a lot of interesting network traffic. Traffic to and from the web on a daily basis ranges from the boringly mundane (that day's headlines from a local newspaper's web site), to the acutely personal (someone's brokerage account number and password and the list of holdings in that account). As use of the web has grown, so has the list of technologies related to it. We look at a variety of these technologies and their security implications below.

As far as the web server itself goes, the web server is responsible for receiving requests from clients and sending back to the client the data that satisfies the request. In many (perhaps most) cases, the client requests a file whose name ends in ".html" or ".htm", which indicates a static text file stored on the server, then the server reads the file into memory and sends it back to the client. In other cases, the client requests a file ending in ".jsp" or ".asp", which the server recognizes as being a program it should load and run, and then send the results of running that program back to the client. Standard operating system security techniques such as file access protections, as well as optional web-server-specific add on security techniques like configuration files specifying which directories on the web server are accessible to users, are used to control the data which can be retrieved by the web server and sent down to the client.

FYI reminder, just as there are email hoaxes, there are also Web hoaxes -- similar to email hoaxes except they are posted at a web site. Some sites are just for fun and actually are real[48], while others are fictions. **[This footnote and footnote 46 say the same thing!!!!]**

2.3.1 SSL/TLS

SSL, or Secure Sockets Layer, is a protocol developed by Netscape for securely transmitting confidential information like credit card numbers across the Internet, by means of public key encryption technology. It provides assurance that transmitted data remains private and unmodified, as well as providing a way for the sender to verify that the server to which it will be sent really is someone authorized to have the data.

[48] http://www.cheeseracing.org/

This assurance is given by viewing the certificate information for the server. In practice, most users never do this, but theoretically, it could be done.

SSL implementations can (but are not required to) support a huge variety of encryption ciphers, from 3DES to RSA, RC2 to DSA, MD5 hashing for message integrity verification, etc.[49]

Typically, you will know that a site is using an SSL connection when you see a URL beginning "https:" rather than "http:". If you want SSL communication to be passed through your firewall, traffic to the destination TCP port 443 should be permitted.

TLS, or Transport Level Security, is a transport layer protocol based on SSL and is considered to be a more flexible successor to it. Although TLS isn't compatible with SSL v3.0, the TSL protocol does contain provisions for a TSL connection to back down to SSL v3.0 functionality if required. Unlike SSL, it is application-independent. When a connection is made, the TLS Record Protocol first calls the TLS Handshake Protocol, which enables both sides of a communication to authenticate themselves to each other (if desired – this step is currently optional) vi X.509 public-key certificates, negotiate an (optional) encryption algorithm supported by both sides, and exchange key information. After that, the TLS Record Protocol uses the agreed upon encryption algorithm for data exchange, and the agreed upon hashing algorithm to ensure that the message was not altered during transport. The OpenSSL Project includes an implementation of TLS in addition to SSL.

Implementations of SMTP, IMAP and POP3 have all been layered over TLS. Each of these, because of the encryption and additional authentication by TLS, has been assigned a new port number for incoming communication, so that clients contact one server for unencrypted communication, and another for encrypted communication. If you're planning to allow connections to any of these through your firewall, be sure that the appropriate destination port number is open. A NetworkWorldFusion article has the scoop (as of 1999) as to what port numbers are used for which services.[50]

---

[49] "Introduction to SSL", Netscape, http://developer.netscape.com/docs/manuals/security/sslin/contents.htm

[50] Snyder, Joel, "How can TLS increase email security?", http://www.nwfusion.com/newsletters/gwm/0329gw1.html

In terms of OSI layers, SSL and TLS sit above TCP/IP, but below application protocols such as HTTP or IMAP.

Note that SSL is not the same as S-HTTP (Secure HyperText Transfer Protocol), which is designed to send individual messages securely, rather than set up and maintain a secure connection between two computers, as with SSL.

2.3.2 HTTP/S

This objective MAY be referring to the common URL prefixes “http:” and “https:”, or to the S-HTTP protocol. Each page (or URL) you visit on the web is accessed using a particular protocol which is specified prior to the “:” in the URL.

HTTP, used for URL’s beginning with “http:”, is the HyperText Transport Protocol used for unencrypted general communications between web browsers and web servers. It takes care of packaging up page requests, page contents, variables, cookies and the like, and transmitting them between browser and server, or server and browser. HTTP communication occurs by default over TCP port 80, so, you would need to have that port open on your firewall in the direction of the web server (outbound if you just want to let your users surf; inbound if you want Internet users to surf your server).

HTTPS, used for URL’s beginning with “https:”, is HTTP with SSL encryption and authentication extensions. It performs the same function as HTTP, but does so in a more secure manner and is, thus, better suited for transmission of data requiring confidentiality. As mentioned earlier, it uses port 443 instead of 80.

As noted above, S-HTTP is an alternative to SSL for secure communications between a web browser and web server. It provides similar functionality, but uses different techniques to do so. Because of Netscape’s dominance on the Web, SSL took off as the primary secure HTTP protocol, and URL’s referencing S-HTTP, which start “shttp:” are rarely seen today.

HTTP and HTTPS used to be purely web browsing protocols, but a funny thing happened on the way to the future, network administrators, the world over, started blocking firewall ports used for any services they did not feel were absolutely essential in the name of security (a good practice).

Then users were sad, because peer-to-peer chat services, instant messaging, CD database lookup programs, and other fun but non-essential utilities that used ports blocked by firewalls stopped working. Then developers deployed fancy applications using technology like Microsoft's DCOM, and found out that because of the way most network administrators had configured their firewalls, DCOM traffic didn't get through. But developers everywhere that were boxed in by network administrators' security efforts, eventually realized that almost every site allows port 80 and port 443 traffic through – and that hiding (or tunneling) their application-specific protocols inside HTTP, was a way to get them through the firewall's port-level blocking. Much like using a VPN tunneled inside a normal TCP/IP connection hides what's really going on in the virtual network from the tools that manage the physical network, tunneling an application inside another application protocol like HTTP hides the workings of that inner application protocol from utilities and devices which seek to observe or filter it. And thus, began the next chapter in the saga of hackers finding a creative way to accomplish something and security folks scrambling to prevent them from doing it. Today, many firewalls feature content filtering of HTTP traffic so that certain URL's or URL patterns can be blocked, to prevent these other applications from piggybacking into (or out of) the site via HTTP or HTTPS.

2.3.3 Instant Messaging

And here, the CompTIA objectives take an interesting turn, as we find Instant Messaging, which seems, to us, to be a separate category on the level of Email, Web or Directory Services, within the Web category. Ours is not to reason why, just to provide you with information about each of the objectives, so we're going with the flow here.

Instant messaging, or IM, is a "communications service which allows you to create a private chat room with another individual" and communicate back and forth with them, a sentence or phrase at a time. Typical features provided by an instant messaging service in addition to this include: informing a user when someone in their "favorite users" or "buddy list" logs in or out, the ability to set a descriptive indicator to tell your "buddies" whether you are not to be disturbed, if you are available to take calls, etc., and transferring files in addition to text messages.

There are several instant messaging standards out there with limited interoperability between them. Prominent instant messaging protocols include those used by ICQ (AOL's entry in the Instant Messaging sweepstakes) and MSN Messenger. For a time, MSN Messenger played a game of catch-up with AOL/ICQ trying to preserve compatibility between its clients and the protocol used by AOL's clients. It seemed that new versions of each would be released every few hours, as Microsoft caught up yet again, and AOL tweaked something to thwart Microsoft's efforts. Things between them have cooled down since, but AOL is apparently under a government mandate to communicate with other vendors' IM applications. We'll see. In the meantime, many IM vendors are backing the SIMPLE (Session Initiation protocol for Instant Messaging and Presence Leveraging Extensions) protocol, an open alternative to the proprietary protocols in primary use today. Among other things, it specifies standards for voice transmission and conferences over the net.

IM was originally a personal application, popularized by individual users around the Internet using IM to keep in close contact with friends around the world. Then corporate America realized that this technology had a use within the organization as well, increasing productivity by allowing staff to send short notifications to each other without creating yet another email message or dealing with the all-too-likely voicemail message they'd receive via phone.

As of late 2001, it is estimated that 180 million business users currently use IM of some sort, and that by late 2003, 70% of businesses will be using IM.[51]

2.3.3.1 Vulnerabilities

Issues with IM in the enterprise today are many and varied. Since IM is often peer-to-peer as far as individual users go, there is less control over the activities of IM users than many technology managers would like. Who uses it, when and for what purposes are decisions that are left up to users.

Most IM systems do not natively support encryption, possibly leading to confidential information being transmitted via IM and then sniffed by an attacker.

[51] "Instant Messaging Overview", Serverwatch

(*Editor Note: Trillian and ICQ at least support SIM (I think it's called), SecureIM. You might want to check that out.) <Tcat Note: Give Ian a virtual cigar! Yes, it is called Secure IM, supports a bunch of IM's. I have it and will be playing in September to report details in the big reference book.>*.

The entire world can see the underbelly of how documentation comes together. This is a fast and furious freebie. I don't want to take the time to put a 'Hollywood face' on a freebie, and the readers get the added bonus of a peek under the hood of what it takes to bring even a simple book together. Maybe it will knock some of the fantasy of glamour of writing a book off the map.

Additionally, there's little in the way of authentication. In most cases, the IM program starts up when the user logs in. If the user walks away, they are still logged into the IM program and anyone can potentially walk up to their PC, click on a buddy within the organization's accounting department, send a message asking for customer Jane Doe's credit card number, and order themselves a nice new stereo from their favorite net electronics provider without the accounting clerk's knowledge that the person who submitted the credit card number request was unauthorized to do it. (After all, in IM today, messages tend to be sent in text rather than by voice, so they lack even the primitive authentication mechanism we have with voice conversations, that of, "Does that voice sound like the person I expect to be talking with?")

There are also issues with record keeping. For one, users can keep logs of IM conversations, and store them on their personal computers. Depending on the organization's policies on keeping logs of written communications in a central place (or not at all, or only for 30 days, or whatever), this could be an issue. Another potential issue is simply the equivalent of "call detail recording" for IM conversations. Some organizations would like a centralized audit trail that lets them know which users within the organization are communicating via IM, and when they're doing it.

And of course, need I even mention this? There are implementation flaws in just about everything. If a way hasn't been found yet in your preferred IM client, to allow someone to send you a file to be stored under a pathname of their choosing, to run a command of the attacker's choosing on your system, or to accomplish some other impolite activity that would generally be regarded as a security threat, just give it time. Monitor your IM vendor's site for updates, and install them.

2.3.3.2 - 8.3 Naming Conventions

If you know what the 8.3 naming convention has to do with IM, please tell us! Do we take this out now that we know what it is????? No. We document *everything* so we cannot be accused of re-writing history.

Briefly, the term "8.3 naming convention" harkens back to the days of old, when MS-DOS only permitted filenames of the form AAAAAAAA.BBB (that is, up to 8 characters, followed by a period, followed by up to 3 more). Later, Microsoft introduced long file names, but kept in Windows the ability to refer to each file by a short, or 8.3, name. For example, you might see a folder called "Program Files" on a Windows system. Its 8.3 name is typically "PROGRA~1". Windows knows how to translate from one name to the other, and will accept either name for that folder, when accessing it. Why? This is done to maintain compatibility with (the now nearly 10 years old) application programs which were written in the old 8.3 days, and only understand file names if they're in the 8.3 format. The fact that some programs still truncate file names to fit this convention (particularly the suffix, the 3 characters), has occasionally been used by crackers to sneak by an access control rule, getting it to allow access to a file which should not be accessible by having the rule check for access to a file of one name (either the long or 8.3 name), and the actual access occur to the OTHER file, due to a bug in the program's code. But upon a quick look out on the web, we haven't seen mention of an exploit of this in conjunction with IM.

Mikop has a suggestion which reads "in the context I would say the distribution of trojan and other malicious codes by not adhearing to the conventional naming. watchthisporn.jpg.exe etc.[52]

2.3.3.3 Packet Sniffing

As noted above, packet sniffing enables an attacker to view any unencrypted traffic on the network, including IM traffic. If a user within your organization is using IM to communicate with, say, a supplier or support person across the Internet, it's possible that customer numbers, system or network passwords, and all manner of other information you'd rather see kept private, is traveling in clear text across the Internet, from your network to the other party's.

---

[52] http://www.examnotes.net/forums/showthread.php?s=&threadid=56952&perpage=10&pagenumber=1

2.3.3.4 Privacy

There are privacy issues with the use of this technology, because most IM systems do not support encryption or authentication. See the comments above on packet sniffing.

2.3.4 Vulnerabilities

And now, following the numbering of the objectives as furnished by CompTIA, it appears we're back to discussing vulnerabilities related to the web.

In addition to the vulnerabilities related to the categories below, which are described within the individual categories, there are a number of various types of other security issues with the web.

The logging that occurs on a web server can compromise user privacy by providing a history of the user's visits to a site. In addition to recording which IP addresses (which may uniquely identify a particular user) visited, which web page, and the time and date, some browsers provide more information such as the name and version of the browser program used for the access, the user's email address, and other details. Perhaps more damaging to individual privacy are browser history, "favorites" or "bookmark" lists, and the cache of visited pages maintained on client machines, detailing the most recent pages accessed by that user, and the ones they're most interested in (favorites). It is, thus, no surprise that utilities are available to clear these sources of information about a user's browsing habits.

Sometimes webmasters opt to protect information on a web server not through actual security measures, but merely by "security through obscurity". That is, they restrict access to a document by not listing links to it on their web site, and by only providing the exact URL of the document to those they feel are authorized to have access to it. The problem with this is that once someone has that little-known URL, they're free to pass it on to others, who can then access the document without authorization.

*(Editors Note: Might want to mention that more than once Google has picked up a page on one of my websites that's not linked to by my or *any* site on the web (I can prove it via Hitbox tracker amongst other things). Google is a powerful search tool for people looking for stuff they shouldn't have access to!* ***[I am assuming we take this out. Sure, in the production (reference work due in November) this will be graciously woven in. The free PDF is 'warts and all'.]***

Yet another web-related security issue is similar to the issue of email retention. Once a piece of information is out on the web, it really DOES seem to consciously WANT to be free. Documents get picked up by random surfers and reposted to other sites (with or without the original webmaster's permission), or a search service like Google snares a copy of it for indexing purposes, kindly, squirreling it away in its cache for searchers' convenience later. After all, it's quite an inconvenience to an attacker when he sees the memo about the default password policy for new system accounts, which he found through Google, is no longer on-line at its original site. Google empathizes with the attacker, and provides him a cached copy of the document, as originally indexed, from its own terabytes (petabytes?) **[which is it]** of disk upon request. And finally, to ensure that no bad site design ever goes unremembered by history, there's the Internet Archive Wayback Machine[53]. This is a time-based web archiving service that takes snapshots of web pages periodically allowing it to serve as a history of the evolution of web sites throughout the months and years. As with the Google cache, in addition to providing a look at what the site was like at a certain time, it could also provide access to data that the site has since thought to remove from public view.

And all together now: Web servers and browsers, just like other software programs, have implementation flaws that have either been discovered, or will almost certainly eventually be discovered. Keep the server, including the web software server itself AND ALL OTHER HELPER PROGRAMS ON THE WEB SERVER, like perl, php, database interfaces, etc., up-to-date. The same goes for browser software, and its helper programs (Flash, anyone?) **[are we not sure here??]** up-to-date. (Helen is sure. This is geek humor.)

[53] http://www.archive.org

There are many other aspects of web vulnerability – too many to list here. Check the WWW Security FAQ [54] for additional information, and further details on many of the web security topics listed below.

2.3.4.1 Java Script

Java Script refers to program code which is transmitted to your PC via a web browser, and which runs as part of the page. It got its name from its resemblance to Java program code, but the two are no more closely related than that. It is frequently used for displaying an animation when the page loads, highlighting buttons as you move the mouse over them, causing menus to expand or contract when you click on various menu items. In addition to being seen on the web, Java Script is also found in many HTML emails, particularly ads and, most particularly, Spam. As noted earlier in our discussion of Spam, the ability of HTML emails to touch web links and execute script code merely by being displayed in Outlook's preview pane is a security issue.

This is particularly true because, whether you're talking about Java Script in an HTML email message, or on a web page, it can sometimes be used to do some more "interesting" things. For example, in certain versions of Netscape's browser, it can be used to retrieve the contents of users' "bookmark" files (listing web pages they've marked as particularly interesting) and send the info to a server. In versions of another browser, it allows harvesting cached cookies from previously visited sites during that surfing session, and sending them (and any information in them, such as full name, address, SSN, financial data, passwords, etc.) to the nefarious server. Although some security sites recommend turning off Java Script support entirely (if your browser allows you to do that) in order to avoid future data privacy compromises, this often impairs the functionality of web pages and is not generally a realistic option.

Java Script code is written in plain text, and can be viewed by anyone upon viewing a web page. This has some security implications from the point of view of intellectual property. Any Java Script code appearing in a web page is effectively open source, available for the taking by anyone who thinks a particular effect is "neat". I've seen the same code in one place on the net, with copyright notice in comments, and in a completely different place, with the same script variable names and everything, sans copyright notice.

[54] http://www.w3.org/Security/Faq/www-security-faq.html

(Java Script can also run on the server side, with many of the same vulnerability concerns as CGI … but we're assuming that this objective refers particularly to client-side Java Script and the data-privacy and data-integrity issues that result on desktops from its use).

2.3.4.2 ActiveX

ActiveX is a Microsoft technology for downloading miniature executable programs (generally called ActiveX controls) to client machines and then executing them to enhance the user experience of the web site. Like Java Script and Java applets, ActiveX controls are most often used to provide fancier graphics or enhanced user interface functionality, beyond what is supported by HTML.

Unlike Java Script and Java applets, ActiveX controls must be specifically compiled for the processor and operating system on which they will run. As far as I know, this means that at this point, and probably the foreseeable future, they're limited to running on Windows/Intel machines (since other processor architectures like Alpha have fallen out of favor with Microsoft).

(Editors Note: I'm not sure but PocketPC (ARM processors) might be able to run them too.) ***[Do we know about this yet?????]*** Not settled yet.

ActiveX technology is somewhat similar to Java technology in that both provide a way to automatically download and run programs within the web browser window, but the two differ considerably in terms of their security model. In the Java applet model, any action that might be used for suspicious purposes (such as writing a file to disk or doing other things to interact with the electronic world outside the java applet's safe run-time "sandbox") is considered suspect (and often denied). In the ActiveX model, the programmer can write code to do whatever they please. The emphasis is not so much on prevention (since a control downloaded from any arbitrary web site is free to do whatever it wishes on the user's computer), as it is on using digital code signing (discussed in 2.3.4.5) to enable a victim of an ActiveX-based attack to determine who was responsible for it and to go after them. This is not necessarily the best model for secure client/server communications in situations where the client has any reason to distrust the server (read: most Internet web browsing).

One reason is because there are plenty of things an ActiveX control can do to compromise the security of a client machine (like read data off the user's system and send it up to a web server), that users typically cannot even detect – what good is accountability, if users never suspect there's a problem? The other reason is that while digital signing guarantees that someone proved their identity to the some certifying organization, there's no guaranteeing that the certifying organization was someone other than a random geek in a basement with a signature-generating program on his PC, or that the user will even bother to check to see that the source of the digital signature was a respected site.

Because ActiveX controls are distributed to client machines as compiled code which is effectively unreadable by curious users (as opposed to Java Script's text or Java's somewhat-reversible "byte code"), they provide an additional level of intellectual property security – it's more difficult for someone to steal your fancy new button lighting effect and adapt it for their own purposes, if they can't see the code.

2.3.4.3 Buffer Overflows

A buffer overflow is a condition which occurs when a software program tries to copy too much data into too small an area in the computer's memory (called a buffer), causing the data to fill up that area and proceed to overwrite other areas of memory that follow the original area. Generally it results from programmers making an incorrect (too small) assumption about how much data is being moved.

Computers move data back and forth in memory by means of variables, which are blocks of memory, each of which has a location in memory and a size. If you think of each variable as being a pitcher of a certain size and in a certain location on a table full of pitchers, a buffer overflow might look like a huge pitcher of orange juice being poured into a small one, and the overflow juice, which needs somewhere to go, accidentally seeping over the edge of the small pitcher into the pitcher next to it (the next variable in memory). If the pitcher next door originally held fine Australian red wine, and the overflow replaces, or "overwrites", any or all of the wine with orange juice, the next person who tries to use that pitcher of wine is going to find an unpleasant surprise. Much the same thing happens with buffer overflows on a computer. The excess data overflows into adjacent areas of memory, overwriting what was there.

Since the previous contents of that memory were likely in use before it was overwritten, when the software program next accesses that variable, they're going to find unexpected contents, which may cause the program to misbehave or even crash.

A more severe case of buffer overflow occurs when the excess data overflows not just the part of memory holding the program's data, but also the part holding the program's executable code. Program code works like a list of instructions. The computer goes through the list of instructions in order, performing each one on the list, just like you would when assembling a piece of IKEA furniture. If part of that list is overwritten by excess data from a buffer overflow, well, the computer doesn't know any better, and it will continue to treat what's there as a list of instructions, and try to follow them. If what's there is random gobbledygook, the program will probably crash. If it's carefully-crafted gobbledygook which just happens to look exactly like the program code which would, for example, delete an important system file, it'll do that too. The art of exploiting a buffer overflow, thus, requires knowledge of how to construct such sequences of instructions in computer machine language, as well as, knowledge of how to get those sequences to be executed.

A buffer overflow can be created directly by a malicious user in a variety of ways. For example, providing a bogus URL that is thousands of characters long to a web server has been known to crash a web server. The URL doesn't have to be valid. It doesn't have to point to a real web page location … the trick is just in jamming thousands of unexpected characters of any sort down the throat of the web server. At the end of the thousands of unexpected characters might be a bunch of binary data, which is designed to end up in just the right place in memory, that after the buffer overflow takes place, the computer will see that binary data as instructions, and run them.

It seems like new buffer overflows are being discovered multiple times a week these days – in web servers, database servers, compressor libraries, scripting languages. No code seems really immune to the problem, for multiple reasons:

Programmers don't always write perfect code (why else do you think we keep urging you to stay up-to-date with patches and updates?)

Today's systems are made up of so many layers of program code, often supplied by half a dozen or more different sources, that it's as difficult to know what's going on under the hood of a program as it is to know exactly what's going on under the hood of your car, with the hood down.

A programmer might know the code they wrote, but three layers down is another piece of code written by a different company, that is called by another part of the program again not written by our hapless programmer, that passes data in a careless way and causes the overflow.

What to do? If you deploy internally written web applications, do make sure that your programmers are educated as to the pitfalls of buffer overflows in code and how to avoid them, and, of course, stay up-to-date on patches and updates for your web-related software.

2.3.4.4 Cookies

Cookies are a technique developed to make up for a limitation of the web. As originally designed, a web server sat around and handled requests for web pages, without trying to remember who asked for what page last, what data they submitted in that web form asking for their address, etc. Each page request was treated as a separate task, completely unrelated to any tasks (other page requests or web page form data submissions) that came before it. This type of behavior is referred to as "stateless" because the server does not keep track of the "state" of any of the clients who might have requested web pages from it.

Because many web applications need to keep track of where a user has been (and who the user is), the idea of a "cookie" was born. The way it works is that the web server passes the browser a "cookie" which may contain data gathered by the web server (such as the user's email address) or maybe simply a "magic number" understood internally by the web server that identifies which connection (series of transaction) this page request is associated with, and, whenever the browser contacts the web server, it sends back the cookie, thus, identifying itself to the web server. The cookie is often (but not always) stored as a text file on disk.

Cookies themselves contain a variety of information, such as the host to which the cookie should be sent, as well as, whatever unique identifying information (or other data) that the web server wants you to furnish whenever you try to contact it. Those who worry about such things might be concerned that a cookie can contain anything on your hard disk should be consoled by the fact that cookies can only contain information that you once provided to the web server who created the cookie –

while sites could store user ID, password, credit card numbers, etc. that you provide them in a cookie, they generally don't (knowing that it's not a good practice), and instead opt to not store that information at all, or store it on their own secure servers, using a "magic number" cookie to look it up later, as described above.

A generic problem with cookies is that every time your browser sends a cookie back to a server, it's giving someone information about you or your browsing habits. In the case of an e-commerce site using the cookie to look up the contents of your shopping cart from the last time you visited the site, this might not be something you mind. In the case of advertisers' sites using cookies to inform a service like Doubleclick which sites you visit, so that they can construct a more complete profile of your browsing habits, this might be something you mind. To look at it from the other perspective, if you are a webmaster and you are using that sort of tactic to track users' activities, you're taking a risk as well – that your user base won't someday deluge your organization with bad publicity for its privacy infringing techniques.

Although most cookies are of the harmless shopping-cart-pointer variety, there are a number of services out there like Doubleclick that use them to learn more about you. To combat this, browsers offer a number of cookie management security features. For example, you can:

A. Delete some or all of your cookies;
B. Automatically accept or reject cookies from certain sites of your own choosing;
C. Disable all use of cookies by your browser (which leads to frustration, since many sites don't work right without them);
D. Have your browser warn you when it is about to send a cookie to a server, and give you the option of not sending it (which may, as in the above case, lead to the site not working properly because it insists on receiving a cookie from your browser);
E. Choose to save cookies only for the duration of this web browsing session (as opposed to permanently on disk);
F. Disallow cookies that are to be sent to sites other than the main one you're browsing (which protects against the kind of cross-site tracking done by Doubleclick and its ilk).

Another issue with cookies is that they are sometimes used to cache authentication information. For example, being able to provide a cookie to a site when requested, might be interpreted as proof that you're allowed to access the site. If someone copies the cookie from your disk, or captures it by sniffing the network, and then installs it in their Cookie directory on another machine, they've managed to gain access to the site without authenticating themselves. Not a problem? What if that cookie contains a session ID that is used by a bank's web banking application to identify which checking account you're viewing the balance of right now? (Aahh, thought you might start to care right about now…. ;-)

Practically speaking, these issues are becoming less of a concern, as sites use more sophisticated methods to maintain session state, and developers are taking more steps to help minimize the damage done by a compromised cookie, such as tying the cookie to the IP address of the machine for which it was created or including time limits in the cookie which are checked by the server before every use. If too much time has elapsed since the cookie was first created (when the user first logged in, for example), your session is considered expired and you are asked to authenticate yourself to the web server again. Nevertheless, there are still creative ways found occasionally to obtain and then use unauthorized cookies, such as those detailed at http://www.sidesport.com/hijack/ just last year, at prominent sites like Hotmail and Yahoo. Periodically an attacker will brag about finding yet another way to read someone else's web-based email, the site will be taken down, the bug fixed, and life continues.

2.3.4.5 Signed Applets

Signed applets are those applets whose authenticity and data integrity is guaranteed by their author, whose authenticity is in turn guaranteed by a trusted certifying agency such as Verisign.

While the term "applet" is used mostly in the Java world, then, "signed applet" therefore refers to a signed piece of downloadable Java code. Other types of code, such as ActiveX controls can be signed as well.

2.3.4.6 CGI

CGI stands for “Common Gateway Interface”, and it amounts to a way of executing an external program or “script” by sending to the web server a URL request containing the name of the program to execute. The server then runs the program or script, and sends the output (if any) back to the client, providing dynamic content, in contrast to the “static”, fixed page images displayed when “.html” pages are retrieved. For example, there are CGI programs to display web counters, maintain web site guest books, add entries to web “blogs”, display the time using pictures of the relevant numbers, etc.

The problem with CGI scripts is that it’s very difficult to get them right. Most system administrators of large UNIX sites can tell at least one horror story about some way a user found to cause a UNIX shell script that ran with root permissions, to do something other than what it was originally intended, like perhaps copy any file on the system, regardless of file access permissions set on it, to a location readable by all system users. And the same is true of CGI scripts, which are vulnerable to the same sorts of misdirection that historically has enabled exploitation of UNIX shell scripts in non-web server environments.

For further information on how to write more secure CGI scripts, see the WWW Security FAQ[55].

2.3.4.7 SMTP Relay

Once again, your hapless authors are met with puzzlement. SMTP Relay has only vague relevance to the topic of the web, and we’re not sure why it was put here rather than under the Email category earlier in the list of objectives.

An SMTP relay is an SMTP email server that accepts connections from users wishing to send email. It receives the message they wish to send, and then sends (or “relays”, hence the name) that message on to the SMTP server which delivers mail for the recipient’s domain.

SMTP relays are a fine thing. Without them, Internet users around the world who depend on clients such as Eudora and Outlook express to send mail would never be able to get a message out to the Internet.

[55] http://www.w3.org/Security/Faq/www-security-faq.html

However, there's a down side to them as well, which we hinted at when discussing Email and spam earlier in this section. Users connect to SMTP servers, for the purposes of sending email, and then simply start dumping message data into them, without authenticating themselves to the SMTP server. Consider that connecting to one ISP's SMTP server is the same as connecting to another – after all, everyone uses the same standard protocol to send mail. What, then is to stop a spammer from connecting to ANY ISP's SMTP server to send mail, as a way of helping obscure their identity? The answer is, very little. Although SMTP servers didn't start out this way, most now provide the administrator with the capability to block connections from anyone except users who are connecting from addresses in the SMTP server's Internet domain, as a way of prohibiting anyone and everyone from using that SMTP server as a way to dump zillions of spam messages into the Internet.

SMTP relays that do not perform this connection domain check are referred to as "open relays", and numerous administrators regard them as evil. Some administrators, on a perennial quest to rid their corner of the Internet of junk mail, maintain "black hole lists" of sites whose SMTP servers are open relays, and refuse to accept any email from those domains. This can be a minor nightmare for an administrator of one of the blocked domains who has a user who needs to send email to the other domain, and who has fixed the original open relay issue that landed them on the "black hole list" to begin with – but who finds themselves still on the "black hole list", possibly due to inattention by the maintainers of that list.

2.4 Directory

Directory Services are akin to a phone book. Several types of Directory Services exist such as NDS and Microsoft Active Directory. Much has been said about security flaws in Active Directory (replication latency). Robert Williams claims this is not a technical issue but an administration issue. His answer is to be aware of the design of Active Directory and only make changes to a single Domain Controller at one time[56].

Other issues such as the "Mixed Object Access" were discovered[57]. A patch was released less than a week after discovery. Once again, this only reinforces the point of keeping on top of updates and applying patches.

2.4.1 SSL/TLS

The Secure Sockets Layer has proven to be a popular method of encryption that operates just above TCP/IP[58]. SSL has been regrouped to the Transport Layer Security Protocol[59] providing symmetric encryption as provided for by RFC 2246[60].

2.4.2 LDAP

NDS and AD are based on LDAP, which is a subset of X.509. Nobody follows X.509 completely because it is too much of a monster. The subset that is followed is Lightweight Directory Access Protocol.

LDAP deals with authentication and authorization. To make sure authentications are legit, UNIX password hashes access by NIS can be used as well as RADIUS, TACACS+ or Kerberos. Kerberos has an advantage of providing single sign-on, in addition to secure authentication

[56] http://www.windowsadvantage.com/tech_edge/04-16-01_alleged_flaw.asp
[57] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/Security/Bulletin/ms00-026.asp
[58] http://developer.netscape.com/docs/manuals/security/sslin/contents.htm
[59] http://www.kegel.com/ssl
[60] ftp://ftp.isi.edu/in-notes/rfc2246.txt

2.5 File Transfer

Transferring of files generally speaking should almost always be secured. Plain vanilla FTP (File Transfer Protocol) sends data in clear text. That means account information such as passwords is subject to being read. Two solutions exist for this challenge and seen in the next paragraph.

2.5.1 S/FTP

S/FTP is Secure FTP. Two versions exist. The now outdated (and unsupported) SSH1[61] and SSH2. A variety of programs exist for SSH2, from commercial offerings to Open SSH[62]. Note that a trojan was inserted in Open SSH and distributed between July 30th and August 1. If you downloaded Open SSH in that time, you are strongly advised to check the web site. Linux uses Open SSH and has found issues in the source code[63].

2.5.2 Blind FTP/Anonymous

Blind logins can be done only as ANONYMOUS and anonymous users *cannot* see the files and directories within the FTP site.

Anonymous FTP allows you to gain access to a remote machine. There you have limited privileges sufficient to allow for transfer of files from and sometimes to designated areas.

2.5.3 File sharing

Not clear what the SMEs want here. **Of course** File Transfer (Protocol) is about sharing files. Do they want to be sure you know to block Port 21 to prevent? Something about logging (monitoring?) **[What do we want to do here…..Leave as is or do we find the info for it I suspect it could be about files that are shared could have viruses, or lack of accountability (changes, leaked data, etc.)]**

[61] http://www.ciac.org/ciac/bulletins/l-047.shtml
[62] http://www.openssh.com/
[63] http://www.linuxsecurity.com/advisories/other_advisory-2218.html

2.5.4 Vulnerabilities

Beyond the obvious, not sure what the SMEs want. Possibly the hole from June 2000 on Cisco PIX firewall[64] Another possible target is the FTP daemon from multiple sources[65]. A slam against everyone from Caldera to Sun[66]? **[Another hole, do we fill in with facts?? We'll just wait this one out. Consensus will come via the discussion boards.]**

2.5.4.1 Packet Sniffing

Since vanilla FTP is clear text, using a packet sniffer to see the data is not a great feat. In 2.5.1 we mentioned SSH2. Encrypted FTP used the freely available BlowFish encryption algorithm (448-bit Encryption and can be found for free in the footnote[67].

[64] http://www.cisco.com/warp/public/707/pixftp-pub.shtml
[65] http://www.wwdsi.com/demo/saint_tutorials/FTP_vulnerabilities.html
[66] http://www.cert.org/advisories/CA-2002-20.html#vendors
[67] http://www.eftp.org/

2.6 Wireless

For the most part, humans have a tendency to not believe something unless they can directly observe it with one of the five senses. Frequently, even when presented with evidence, humans have a way of denying the data. It took the medical society a long time to believe Anton Van Leeuwenhoek when he discovered life that could not be seen without his new invention, the microscope.

Similarly radio waves cannot be directly observed without equipment. This fact alone makes data sent by this means a difficult risk to manage. Compound the issue with the realization that radio waves can glide through walls and glass like it wasn't there and you have the recipe for a security risk that at first seems impossible. Fortunately, radio is not a new technology and its behaviors are well known. Continue on and discover how you can use wireless technology to your benefit and minimize the security risks.

Think of throwing a stone into a still pond. At the point of impact, the waves travel outward in a circle decreasing in strength, as it gets larger. If the wave of water does not hit an object, it eventually disappears. If it does hit an object, such as a large stone in the water, the wave action bounces back. Radio waves behave in a quite similar fashion. The difference is since a radio wave is electromagnetic; metal is used.

2.6.1 WTLS

The Wireless Application Protocol is another wireless technology that was created for wireless resources with limited capability such as a cellular phone. Among the protocols within WAP is the security layer Wireless Transport Layer Security (WTLS). As pointed out by Markku-Juhani Saarinen[68] at the University of Jyvaskyla, Finland the WTLS protocol design limits the effectiveness a Certificate Authority such as Verisign can have while supporting WTLS. It appears that this research has been taken to heart as seen in the papers available from the wapforum.org[69]. If you visit the site you may notice the group is in the process of quietly renaming itself to the Open Mobile Alliance.

[68] http://www.cc.jyu.fi/~mjos/wtls.pdf
[69] http://www.wapforum.org/what/technical.htm

2.6.2 802.11x

802.11 is really a family of wireless speeds. The first was plain 802.11 with no letter extension. Released in 1997 with a data rate of 1 to 2Mbit. Two years later both 802.11a and 802.11b specifications were unveiled. The b version quickly caught on because of reasonable speed (11Mbit) at a price point far lower than 802.11a with a maximum throughput speed of 54Mbit.

While 802.11 are Ethernet, not all Ethernet is the same. Wired Ethernet uses collision detection, while 802.11 (wireless Ethernet) uses collision avoidance. This behavior difference is subtle but critical. In wired Ethernet (802.2, 802.3) if two data packets attempt to share the wire at the same time, the irresistible force meets the immovable object and they both die. That forces a resend of each packet (held back by a random time-out on each transmitting unit).

With 802.11, each wireless node broadcasts its intention to transmit telling the other nodes to wait their turn. This is great until one node cannot see the wireless signal due to interference. This is called the hidden node problem. The access point solves this by using the old serial protocol Request To Send/Clear To Send (RTS/CTS) and does not grant one of the nodes a CTS after the RTS

2.6.3 WEP/WAP

WEP

The Wired Equivalency Protection model sounds really good, and, the reality is it really isn't. That is not to say don't use WEP; it's better than nothing. A peek at WEP will reveal how it works and what you can to do better the odds.

WEP works by using a RC4 encryption scheme, (Refer to encryption for details on RC4) with a key that can be 40, 64 or 128 bits in length. (New models released in the 2nd half of 2002 now offer 256-bit encryption.) The design in 802.11 for RC4 uses a shared key. The access point sends a random number at the registration request. The receiving node assigns the key with a secret key that was preshared. The access point checks the results and allows the node to sign on. Data between the devices is encrypted by one of the values listed.

The method described is known as one-way authentication. Stated another way, the access point knows it is from some group of computers that has the preshared key and cannot identify a specific computer.

Given this, it is possible for a rogue computer to pretend it is an access point.

Another issue with WEP is RC4 being used in wireless. RC4 was designed for a synchronous stream. The nature of wireless communications is such that the signal can be dropped very easily. The designers address this challenge by changing the key for every packet. This uses up unique keys very rapidly, which forces key reuse. Key reuse breaks a cardinal rule in RC4 design. This is the good part of WEP.

The less than stellar news in the design of 802.11b and WEP is the use of RC4 has as part of the logic a number known as an initialization vector or IV that is not encrypted. Too many product offerings start the IV at the number we call 1 then use 2 for the next IV, followed by 3, etc. So, scoop up about 5 million packets of data and you can figure out the WEP pattern. In a large wireless network with heavy usage the combination of keys is used within hours, as proven by research at the University of Maryland[70] and the Berkley campus of the University of California[71]. A single intruder sending an email to a valid email address on the wireless network further reduces security since the intruder knows what the unencrypted message contained, narrowing the search pattern. If an intruder doesn't want to work hard they simply use the lazy approach and use a program such as Airsnort [72] or airtraf[73].

The moral to the story is change keys often.

WAP

WAP has two meanings. The older is Wireless Application Protocol. WAP is a newer protocol in the TCP/IP suite. At first look, it appears that existing protocols such as HTTP and TCP fill the need. These protocols were designed with the idea, that a device has continuous connection and can send multiple requests for data. Wireless devices have considerable constrains regarding power, processing and display parameters.

---

[70] http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm William A. Arbaugh University of Maryland

[71] http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html Nikita Borisov, Ian Goldberg, and David Wagner

[72] http://sourceforge.net/projects/airsnort

[73] http://www.elixar.net/wireless/download/download.html

Other considerations such as signal fade, long latently times, security and low bandwidth all contributed to the need to release WAP.

WAP addresses these issues in creative ways. Instead of uncompressed text strings, data is sent in compressed binary packets. *Hyper Text Markup Language* (HTML) is replaced with the *Wireless Markup Language* (WML), based on the *Extensible Markup Language* (XML) format. This allows data to be optimized for smaller displays found in hand-held devices such as Web enabled *Personal Communication System (*PCS) phones and *Personal Digital Assistants* (PDA) such as the Palm Pilot and Hand Spring units.

WAP differs from other offerings by using UDP instead of TCP for lower overhead, in combination with the new *Wireless Transaction Protocol* (WTP) on the transport layer. A feature of WTP is a small session re-establishment protocol that allows a transmission to continue from the drop point without the overhead of the initial session.

WAP overcomes the limitations of being UDP based by the *WAP Gateway*. This service acts as a proxy server between the wireless device and the wired Internet. By eliminating the need for multiple routes, considerable overhead required for TCP is removed. Part of the functionality of the WAP Gateway is to act as a translator between the connection-oriented TCP protocol and WAP's connectionless UDP protocol.

A second meaning for WAP is Wired Access Point. This is the junction between 802.11 a/b/g wireless data traffic and the wired network. In fact 802.11 uses WAP with the Wireless Markup Language to communicate with WAP the Wired Access Point.

In some countries, WAP/Web enabled PDA and PCS devices are very popular.

2.6.4 Vulnerabilities

(This work is a cut/paste from our upcoming Reference Book for Security+ and was done before the Security+ objectives were released. Note that other vulnerabilities are covered in the preceding paragraph.)

Think back to the rock in the pond analogy. A larger rock will make stronger waves than a smaller rock. Minimizing transmission power reduces the changes your data will leak out of the intended area. Antenna

placement will also have an effect[74]. Attempt to place antennas as far from exterior walls as possible. Typically the interface between the wired network and the transceiver is placed in a corner in an effort to hide the electronics. That places the network signal outside and easy to intercept. In effect, you have put an Ethernet jack for your network in the parking lot. Besides controlling power output and antenna placement and configuration, consider shielding, the next topic.

Shielding

The building in which the wireless network is operating can be used as a shield for itself. The downside could be a negative impact on pagers and cellular phones. An additional plus is this reduces your risk of a wireless denial of service attack. Keep in mind that 802.11b operates at the same frequency of a microwave oven. Something as simple as disabling the safety interlock of a consumer microwave oven creates a DOS weapon with up to 1000 watts of 802.11b interference. A more determined black hat may invest in a 'heavy duty' antenna, capable of handling up to a 300W[75] input from the inner workings of a microwave antenna.

If you are lucky enough to be designing for a new construction, consider metal studs and run a bare copper ground wire from the studs to a grounding rod. Before installing dry wall attach very thin layers of aluminum to the metal studs.

More often retrofitting an existing site will be your task. Steps to attenuate the signal include metallic window treatments and metallic paint on wall surfaces.

At minimum, use a laptop with a wireless network card attached loaded with the vendor's software to reveal signal strength and perform both an interior and exterior 'walk through'. Document the authorized locations. This needs to be done on a regular basis. Wireless LANs can be bought at the store and plugged into a hub by users without the knowledge of the company. Gartner has estimated that at least 20% of companies have rogue Wireless LAN installations[76]. This issue is so widespread Microsoft CEO Steve Ballmer recently took advantage of it as told to Infoworld.[77]

---

[74] http://www.80211-planet.com/tutorials/article/0,4000,10724_1116311,00.html Article on running a Site survey by Jim Geier
[75] http://www.hyperlinktech.com/web/hg2415p.html
[76] http://www.eweek.com/article2/0,3959,7744,00.asp
[77] http://www.infoworld.com/articles/op/xml/02/07/22/020722opcurve.xml

"I was in a hotel in Sun Valley last week that was not wired," Ballmer recalls. "So I turned on my PC and XP tells me there is a wireless network available. So I connect to something called Mountaineer. "Well, I don't know what that is, but I VPN into Microsoft. It worked! I don't know whose broadband I used," he chuckles. "I didn't see it in Bill's room. I called him up and said, 'Hey, come over to my room.' So soon everyone is there and connecting to the Internet through my room."

Now suppose it wasn't Microsoft employees using the Mountaineer wireless LAN to tunnel into Microsoft. If this was a group who wished to cause the Redmond firm grief, they could have done so without worrying about covering their tracks since a search would go directly to the owner of Mountaineer. "Oops."

Networks that are large enough to make pre-enrolling MAC addresses impractical should look at a VPN (How a VPN works is described in Encryption) before the connection to a wireless network. Options such as the Cisco Aironet utilize the Cisco Lightweight Extensible Authentication Protocol (LEAP) to ensure mutual authentication between wireless clients. This includes a back end RADIUS server, dynamic WEP keys, and changes the Initialization Vector (IV) on a per packet basis. Vendors such as Fortresstech offer products such as Airfortress[78] that is placed between the Wired Access Point and the rest of the network. Other vendors have updated drivers for their wireless offerings to improve security on existing products.

Netstumbler is a free windows-based wireless sniffer that looks for a SSID being broadcast. While this is useful for finding the rogue access point that was set up without thinking of the ramifications, a Linux based sniffer called Kismet is totally passive and can find a wireless network without revealing itself in the process, as well as, being able to capture data from signals too weak to fully participate in a wireless LAN. Kismet was created to work hand-in-glove with the free protocol analyzers such as Ethereal[79] for Windows or tcpdump[80] for Linux or Windows. Ethereal does an incredible job in many ways. As an example, check out their site[81] to see how easy it is to see email messages.

[78] http://www.fortresstech.com/AirFortress%20Overview.htm
[79] http://www.ethereal.com
[80] http://www.tcpdump.org
[81] http://www.ethereal.com/introduction.html#screens

With so many sites running open wireless networks, it can be difficult for a person to remember all the sites they have found. Kismet solves this issue by adding support for GPSDrive[82] to map all the sites found. If you don't have the time to try all this yourself, the creator of Kismet has done a great job of showing how all this can work together. Just point your browser to the footnote[83] .

More 802.11b Lockdown Tips

While changing defaults such as the SSID, change the password on access points.

Turn off DHCP and assign static addresses, if possible.

Make sure hardware has upgradeable firmware.

Consider configurations that are "closed". Some vendors offer non-open options, such as turbo mode, if not, use an access point that does not broadcast an SSID.

The coverage on wireless security to this point has been a discussion on locking down the wireless LAN. 802.11b is also used for public access. Some sites are ad hoc, meaning that people who don't mind sharing a bit of their broadband access run them. One site created to help find these sites ishttp://www.freeap.com. The other approach is offered by firms such as Boingo[84] Wireless who offers paid subscriptions to 'hot spots'. Project Rainbow[85] is sure to make commercial hot spots very commonplace. Since public access is the point, WEP would not be running. Be sure to install a personal firewall[86] before joining a hot spot. If your business can take advantage of any ad hoc or commercial hot spot, run any web page containing internal corporate data via https, instead of http. One final option in the wireless LAN that is worth exploring is 802.11a, seen next.

[82] http://www.kraftvoll.at/software
[83] http://www.kismetwireless.net/screenshot.shtml
[84] http://www.boingo.com
[85] http://zdnet.com.com/2100-1105-944732.html
[86] http://www.theguardianangel.com/firewall_comparison.htm

802.11a

One other option (for the moment) is to spend a little more money and install 802.11a. While more expensive, it operates in a different frequency, reducing (for the moment) the available hardware that can be used against you. Another possible advantage is the fact that 802.11a has a much shorter range.

Bandwidth of 54Mpbs (up to 75Mbps in some vendors' unique implementation) is sufficient for real-time video. Encryption in some products goes up to 156bit encryption and the range is considerably shorter in the 5Ghtz spectrum. The same antenna site mentioned above offers 5Ghtz units.

In the real world, changes are happening in Internet Time, so keep your eyes and ears tuned in.

2.6.4.1 Site Surveys

More sensitive environments may wish to either hire a firm to perform a Technical Surveillance Counter Measures (TSCM) sweep on a periodic basis. Extremely sensitive sites would be wise to consider installing inplace monitoring in addition to periodic searches for light based Infra-Red (IR) or cellular phone based unauthorized equipment. With the coverage of security of 'wireless cable' complete, it is time to move up the OSI protocol stack and address the logic blocks in wireless networking.

.

Chapter 0011

Infrastructure Security (20%)

3.0 Infrastructure Security

In this section, we'll talk about the components of a network, and the security implications of them. We'll also discuss security-related processes such as intrusion detection and increasing the security of networks, operating systems and applications, which is known as "hardening". This section is a practical look at what you want to consider when securing your network -- everything from the bits and bytes traveling across a network cable to the server applications running on it.

"Why cover everything from hardware devices to applications in a single section?", you might ask. Well, to partly reinforce the point that a network is only as secure as its least secure component. You can have the most secure Internet gateway whatsoever, but, if one random user has installed a modem on her PC and uses that modem to connect to her personal Internet account, your internal network is exposed.

3.1 Devices

Networks include a variety of physical devices, and each type of device has its own security considerations. Some of these provide additional security for your network, and some are likely to be points of exposure. We'll cover some of the most common network devices in this section.

3.1.1 Firewalls

In the non-computer world, a firewall is a fireproof wall used as a barrier to prevent the spread of fire.[87] In the computer world, it's a barrier intended to prevent undesired access to computer and network resources, keeping those on the outside of your network out, and those on the inside of your network, compliant with company policies on network use. When a firewall is protecting an entire network, it is normally a separate system that is not used for any other task on the network. It may be an ordinary PC running specialized software, or perhaps, a customized hardware 'box' specifically manufactured to provide firewall functions.

There are two main types of network firewalls: application-level and network-level. Each has its advantages and disadvantages.

Application-level firewalls involve the use of one or more "proxy" programs on the firewall, which act as intermediaries between internal and Internet hosts. Usually, a separate proxy program handles each different protocol passing through the application-level firewall. The proxy program accepts a connection request from one side of the firewall, notes the desired destination address, and then creates a connection request of its own that is sent to the ultimate destination if it determines through its rule base that the connection should be allowed. The proxy then carries on two separate, simultaneous conversations. One between the network client talking to the firewall, while thinking it's talking to the server and one between the firewall and the server, which thinks it's talking directly to the client. It carefully passes each side's requests and responses to the other while keeping external systems from being able to play low-level TCP/IP games with internal systems, and trying to isolate each side from bad input that might exploit vulnerability in server or client software.

[87] American Heritage Dictionary of the English Language, Fourth Edition, Houghton Mifflin Company, 2000.

When determining whether or not to allow a connection, an application-level firewall can look at many criteria, including items such as packet source address, destination address, source and destination port numbers, and possibly other items such as user ID, user group, etc. Since the proxy programs have knowledge of the protocols, and control each conversation, it's possible to define rules based on subcommands within the protocol. For example, you could allow certain users to issue the FTP "put" command to save a file to your FTP server, but disallow "put" access to all others. This very flexibility is also a limitation, because if an application-level firewall doesn't have a proxy program for a protocol, the protocol can't pass through that firewall at all, as some users of Microsoft Proxy Server were unhappy to find out when trying to create connections to the Internet with proprietary client/server software.

This means that one thing to look at when evaluating application-level firewalls is, "Does it support all of the protocols I want to pass through the firewall?" Another limitation of application-level firewalls is that workstations may need to be configured to send traffic to the firewall proxy, for example, specifying a proxy server address in browser settings. (Fortunately some browser manufacturers make it possible to do this in a somewhat automated fashion). Because application-level firewalls do so much work to verify and maintain each connection, they're also the slowest type of firewall. If you're buying firewall software to install on your own computer, and plan to use application-level proxies, don't scrimp on the computer's processor or network cards!

Network-level firewalls work at the individual packet level, inspecting specific fields within the packets and comparing them with user-defined rules that determine whether or not should allow the traffic through the firewall. Typically the rules take into account any combination of the IP protocol number, packet's source address, destination address and source and destination TCP/UDP port numbers. Typically, you can either "allow" access to an address (or subnet), or "deny" it. For example, you might wish to "deny" incoming connections to your company's FTP server, by all IP addresses on the Internet except those belonging to certain collaborators. Or, you might want to "deny" all incoming connections to your network on the commonly used SQL Server port.

There are two varieties of network-level firewalls: packet filters, and stateful packet inspection firewalls. Packet filters are the simplest firewall technology. They examine each packet going across the firewall's network interface(s) individually, and compare it to its known rules.

Because they typically do little processing, they tend to be the fastest type of firewall. Stateful packet inspection firewalls take the idea of packet filtering one step further, by considering the "state" of the connection when a packet is inspected.

A stateful packet inspection firewall keeps track of all active and pending network connections through the firewall. It knows which side (external host or internal host) initiated a particular connection, the status of that connection, and possibly a bit about the expected packet contents for conversations using application-level protocols like SMTP or FTP, plus the standard packet-filtering details known by stateless network-level firewalls. By knowing connection status, a stateful packet inspection firewall is better able to protect the network from packets with spoofed addresses trying to masquerade as legitimate packets in that conversation. The tradeoff is that configuration of rules for this type of network-level firewall can be more complex than for simple packet filters.

Unlike application-level firewalls, network-level firewalls are transparent to workstations on the network. No configuration changes are necessary to hosts when implementing or fine-tuning the rules on a strictly network-level firewall. On the downside, unlike with application-level firewalls, direct TCP/IP connections between external Internet hosts and internal systems are permitted. This potentially means that your network's security could be compromised if a weakness in the lower levels of TCP/IP on any of your workstations was to be discovered and exploited by someone out on the Internet. With an application-level firewall, since packets are recreated on the firewall before being directed to internal hosts, only the firewall itself is likely to be susceptible to attacks that take advantage of flaws in the lowest levels of a TCP/IP stack.

Is there anything else a firewall can do for the network? In addition to providing access control based on protocol and source/destination address, firewalls can provide other functionality such as:

A. Access control based on time of day or an authenticated user ID
B. Session logging, useful for tracking connection utilization
C. Intrusion detection and notification

Additionally, some provide a facility known as Network Address Translation, or NAT. Most commonly, this feature allows computers on your internal, non-Internet-addressable network to gain access to the Internet, by automatically translating internal network addresses to external network addresses. The benefit of this is that your internal IP numbers are not known or accessed by Internet hosts. Any internal nodes for which NAT is not performed, is effectively isolated from the Internet (unless, of course, someone compromises one of your internal systems and uses that as a springboard to get to other internal systems).

In addition to firewalls used on enterprise networks, there is also personal firewall software, which protects a single system, or a small (generally home) network. Personal firewall packages such Zone Alarm[88] and Tiny Personal Firewall[89] offer a subset of features of larger firewalls, generally being lighter on logging and management capabilities may not needed by most home users, and add a few features intended to appear to home users, like "Winroute Pro" (which does packet filtering)

Which one should I get?

It depends, on many factors. Many people think Checkpoint is the market leader in network firewalls. Other companies, whose networks are full of Cisco's networking equipment, like Cisco's PIX. Still others, particularly those constrained by cost and those who like to inspect the source code for their security devices, like open source firewalls like the network-level iptables in Linux. Several independent organizations certify firewalls, including ITSEC, TCSEC and Common Criteria. Commercially, ICSA and West Coast Labs Check Mark provide somewhat more-limited certification.

If we buy it, will it protect us?

ICSA Labs reports that, "an alarming number of firewalls aren't functioning as intended."[90] This is largely due to the "people" component in the firewall configuration process, rather than shortcomings in the firewalls themselves. Many firewalls are simply improperly installed or configured.

---

[88] http://www.zonelabs.com/store/content/home.jsp

[89] http://www.tinysoftware.com/home/tiny2?s=8698768489789983685A1&la=EN&va=&pg=solo_download

[90] Firewalls FAQ, ICSA Labs, http://www.icsalabs.com/html/communities/firewalls/faqs/index.shtml, 2000.

So, it is important that staff is trained on proper firewall configuration and security techniques, and that firewall configurations and rules are documented. Again, security isn't a one-time action that is taken and then over with. It is a process! A single setting can sometimes turn on (and thus, OFF) packet inspection, turning a firewall with a well-designed rule base into a box that blindly passes along every packet it sees.

In addition to regular audits of your firewall rules and configuration, and perhaps an occasional penetration test by staff (or a consulting group) using an outside Internet connection, what else can be done to maximize the security of a firewall? Check with your firewall vendor regularly to ensure that you are running the most up-to-date software, which is likely to be the most resistant to known vulnerabilities. Follow popular mailing lists like BUGTRAQ, and Usenet newsgroups related to your firewall platform to keep up with potential issues. As with any network device, if the vendor has a security bulletins list, sign up for it and take the recommendations posted on it. For example, sometimes a vendor suggests temporarily disabling a feature until a patch for a security vulnerability involving it is tested and released. Ignore these (we know many people will, often based on, "I don't have the time") at your own peril, and don't say you weren't warned.

3.1.2 Routers

A router is a network device that connects networks, forwarding packets to and from them as needed. It determines how, and where to forward packets based on internal routing tables that may be hard-coded on the router and not subject to change, or may be dynamically updated by other routers on the network as the best routes to different destination networks change.

Generally, a router contains at least two network interfaces, and in larger networks, often more. Connected to each interface is a specific network or network subnet. Packets come into the router from each interface; the router compares each packet's destination address with its internal routing tables, and sends the packet out the appropriate interface, on its way to its destination. Often the only router on a small business network is one with only two interfaces, one for the internal network containing the organization's workstations and other devices, and one for the company's Internet connection.

Originally deciding which network to send a packet to, to move it toward its ultimate destination, was all the functionality that routers provided. In the case of the small business router described above, a router would often do little more than serve as an expensive connector between two networks, with very little in the way of routing decisions to make.

In larger networks, routers can be employed to segment traffic and regulate traffic among segments, helping to ensure that network performance is as optimal as possible and limiting the amount of traffic that can be spied on by those who install a network "sniffer" to inspect packets traveling along the wire.

As networks evolved, router manufacturers enhanced their products, adding a number of security-related features. One of the most notable is packet-filtering functionality similar to that found in many firewalls. In the Cisco world, the packet filtering rules are called Access Lists, or ACL's. There are simple ACL's which only allow or deny traffic based on a single IP address, and extended ACL's implementing a fuller set of packet-filtering criteria, including source address, destination address, protocol family (TCP, UDP, ICMP), and port. For example, the following command creates an extended ACL that denies access to the POP3 port on your internal mail server, 192.45.4.72, from all external systems:

 # Access list 101 deny TCP any 192.45.4.72 0.0.0.0 (eq telnet)

Other security features commonly found in routers are configuration options to protect malformed or possible vulnerability-exploiting packets from being forwarded to other network interfaces, and to prevent packets with spoofed source or destination addresses from entering or leaving the network. How does the router know a spoofed address when it sees it? If the router sees a packet coming into it from an external interface, and that packet's source address is set to an IP address within the internal network, odds are that the address was spoofed. Similarly, the router knows that packets coming into it from internal networks should not have external network source addresses – if they do, that may be a sign that someone is using one of your systems as part of a DOS or DDOS attack. Blocking such outgoing packets is just a matter of being a good net neighbor. If everyone did this, a number of DOS techniques would be much less effective, since they rely on the ability to send out packets with a spoofed source address equal to one on the target's internal network.

Routers communicate using special protocols known as "routing protocols", which include standard protocols like RIP, RIPv2, OSPF, BGP and vendor-defined protocols like Cisco's IGRP. These protocols, like many other Internet protocols, have security vulnerabilities. For example, RIP (v1) can be easily spoofed because its messages are not authenticated, so anyone can send update messages via RIP; RIPv2 relies on optional clear text authentication, transmitting passwords across the network where they can be intercepted and used in later spoofing attacks. Even OSPF can be spoofed if the protocol is not used in cryptographic authentication mode.[91] Once you've succeeded in spoofing a router message, you can do a number of things, like redirect switched network traffic to other segments so that it can be sniffed, create a "black hole" denial of service attack by advertising a non-existent router with a priority route for all traffic, etc. Generally, the stronger the protocol's authentication is, the less vulnerable it is to spoofing. Of course, this implies that those with access to router passwords maintain its confidentiality. In keeping with the evolution of the Internet, later protocols tend to incorporate better authentication than earlier ones. When you have a choice, opt for these newer protocols, and use the authentication features they provide.

Like firewalls, routers are implemented substantially in software, and from time to time, issues affecting security are found in that software. Therefore, the guidelines we offered for firewalls also apply to routers – watch for software updates and install them when they are available, subscribe to vendor security bulletin lists and other security discussion lists. Also, similar to firewalls, routers can be challenging to configure properly. Be sure that your network administrators with router configuration responsibility have been trained on proper techniques, and possibly even certified by your router's vendor as having satisfactory knowledge of router administration.

3.1.3 Switches

A switch is a network device that, like routers, forwards packets between LAN segments, providing somewhat of a subset of the functionality of a router, at a lower price. The simplest switches generally operate at the data link layer (OSI Layer 2), and base packet-forwarding decisions on physical device locations (for example, destination MAC address) rather than on supplied rules for packet routing, as in full routers.

---

[91] Russell, Chris. "Understanding Dynamic Route Protocol Vulnerabilities", Information Security Alliance, Rev 1, October 7, 2001.

Switches are useful in controlling access to network packets traveling on a wire. If a packet isn't sent over a particular segment of the network at all, it can't be "sniffed" by someone attempting to intercept information (like system administration passwords) on that segment.

Of course, a caveat to this is that switches have been designed primarily to enhance network performance, not network security. It is possible to barrage some switches with an excessive amount of traffic from unused MAC addresses, causing the switch to overflow its MAC address table and switch into hub mode. Once the switch is in hub mode, traffic is no longer restricted and sniffing is once again possible. With another switch, frames are always forwarded to all ports on the switch and the switch processor is relied upon to explicitly tell all ports except the correct one to drop the frame. If there is an excessive amount of processing on the switch (say, from handling bad packets) the processor may not get around to telling the other ports to drop the frames and the frames are sent out on all ports, as with a hub.[92] Another way to sniff switched traffic is to use spoofed ARP packets to misinform the switch of the MAC addresses for the hosts whose traffic you wish to sniff.[93]

Switches can be targets for attackers, who can gain access to administrative features of managed switches by using default passwords, or sniffing switch passwords sent in clear text via SNMP or telnet.

3.1.5 Modems

Modems allow users to use telephone lines to call in to a network, computer, or fax machine from a remote location. Most modems today use the V.90 standard, theoretically communicating at up to 56kb/sec, although they are limited to 53kb/sec (kilobits per second) by the telephone network. Older modems may communicate at 33.6kb/sec, 28.8kb/sec, 14.4kb/sec, or even 9600bits/sec or slower. Other devices that connect a home PC with a cable Internet connection or DSL connection are often referred to as "cable modems" or "DSL modems", but they more closely resemble routers than modems in the traditional sense.

[92] Turner, Aaron D., Network Insecurity with Switches, http://synfin.net/docs/switch_security.html.
[93] Switching and VLAN Security FAQ, http://www.fefe.de/switch

Modems are used in corporate settings. Most commonly for telecommuting by employees working at home, providing after-hours support, exchanging data among business partners, sending and receiving faxes, and accessing the network from off-site company locations that do not have permanent network connections.

Modems can be internal (on cards which plug into a PC or laptop) or external (separate boxes outside the computer). External modems typically connect to computers via serial or USB connections.

In the case of a serial connection, and probably also for USB, a "tap" can be inserted between the computer and modem, so that communications can be intercepted and recorded, compromising data privacy. For that matter, an appropriate device between computer and modem could even take over the conversation.

If your users have modems at home, and directly dial into your network, this means that there are modems waiting for incoming calls on your internal network, and therein, is the problem. Anyone who knows (or finds) the telephone number to these modems can call them and attempt to access your network. Relying on "security through obscurity" by selecting a modem telephone number that doesn't resemble any of your corporate phone numbers won't protect you. It just makes it a bit more difficult for those specifically targeting your organization. Assume that one way or another, people you don't want to have your modem phone numbers, eventually will. Given that, you need to protect your network by making sure that when personnel dial in to your network, they properly authenticate themselves. Non- or poor password dial-ins negates most (if not all) of the good done by a very well configured firewall between your internal network and the Internet.

One way to combat the dial-in security issue is to use only modems with "callback" capability. That is, when they receive an incoming call, they call back one of a set of pre-determined numbers, and let the user's PC answer, before offering a connection to the network. Unless someone resets the callback number, callers at unauthorized locations will not be permitted access. This is an example of redundant security measures – in this case employing both a password or certificate, and corporate knowledge of a user's location, for authentication. Of course, this mechanism isn't practical when your users are dialing in from unpredictable locations like hotels around the world.

Another security issue is that of “rogue modems.” These are modems on individual workstations whose users who wish to access their PC from home “easily” “without going through all that security stuff we have on the network dial-in”. They install remote access software on their PC at work, make sure the modem is turned on when they leave the office, go home and dial in directly to their office computer often without a password or any type of authorization gaining access to its files and network resources. Again, if THEY can dial in, ANYONE can dial in, if they find the phone number. So a very useful security precaution is to limit use of remote PC access software on your network, perhaps even regularly auditing for the presence of it on PC’s, and make sure when it IS used that strong passwords are employed.

3.1.6 RAS

RAS, an acronym for “Remote Access System” or “Remote Access Services”, is a function that authorizes users dialing into a network and then connects them with the network so that their PC becomes just another node on the network. In most corporate networks, it refers to the RAS function available in Windows. One or more Windows computers (or boxes implementing the same protocols) can be set up as “RAS servers” which accept modem connections from incoming telephone lines. Any user with a modem and the correct authentication information can access your RAS network, so you might want to consider additional levels of security such as utilizing the callback feature (in your modem, or in RAS itself). Typical RAS servers allow controlling access by user ID, time of day, and other factors. You can also set parameters such as the maximum number of incorrect logins per day per user ID.

Since RAS gives users access to the corporate LAN as if they were another local user, you might think that your entire network is wide open to any RAS users, but this is not entirely true. One interesting feature of RAS is that you can block certain protocols from use over RAS. So, for example, if there are certain applications you want to be run ONLY by users who are physically in the office, you might design them to run under a protocol that you don’t pass through RAS, such as IPX.

RAS can use a variety of authentication techniques, including Password Authentication Protool (PAP), Shiva PAP (SPAP), Challenge Handshake Authentication Protocol (CHAP), Microsoft CHAP (MS-CHAP).

Of these, CHAP and MS-CHAP are more secure than PAP and SPAP, because the challenge approach does not require an encrypted (SPAP) or unencrypted (PAP) password to be sent over the wire from client to server. The benefits of the challenge approach are covered in more detail in the section on CHAP. Also, Windows 2000 and later versions support EAP, the Extensible Authentication Protocol, which is an extension to PPP that enables the use of third-party modules to authenticate RAS users. For instance, smart cards, Kerberos or S/Key mechanisms can authenticate users, if the appropriate module is installed and configured.

3.1.7 Telecom/PBX

Your organization's telecommunications facilities are also part of its IT infrastructure. The PBX, or Private Branch Exchange, is the controller of the telephone network within an organization. It coordinates access to a limited number of "outside" telephone lines, each of which has its own telephone number, by a greater number of "inside" lines within the company. PBX's are often used by medium and large-sized companies because it's less expensive than connecting an outside line directly to each internal line. For example, a site might have 400 internal lines, or extensions, and 80 external lines. Another benefit of PBX's is that users within the organization can reach other users connected to the PBX by dialing their short (3 or 4 digit) "extension" rather than a full telephone number. Many PBX's contain what is known as a remote maintenance, or remote diagnostic, port which is used by the PBX vendor to reconfigure the phone system, upload and install software updates, download logs and perform other privileged operations. As you can see, a PBX is "phone system server" of sorts.

The same type of vulnerabilities faced by an organization's data network are also faced by its telephone network, including theft of service (through long distance toll fraud), compromise of data privacy or integrity, unauthorized access to privileged functions, denial of service, and opportunities for reconnaissance by an attacker interested in patterns of calls by one or more users. An example of the risks faced by PBX's is a situation in which an attacker gained access to a hospital's PBX, and then for the next two years, periodically blocked calls to and from the hospital, connected internal staff to outside numbers they did not intend to contact, and placed bogus emergency calls that appeared to come front the hospital itself. [94]

[94] Kuhn, Richard, "Security for Private Branch Exchange Systems",

Common pitfalls to watch for, in terms of the security of your traditional PBX include:

A. Default passwords on manufacturer-installed accounts (information on these can be found in the PBX switch documentation or from your vendor – you can be sure that potential intruders know what they are);
B. Features that can be accessed via the phone system, such as voicemail or switch reconfiguration functions (typically, the password security on these features is not significant, and can be defeated by brute force);
C. Leaving a modem connected to the remote maintenance port, even when you do not know of any scheduled maintenance that will require it, leaves open a path into the system for anyone who wants to try to use it;
D. Software updates sent from the switch vendor to the system administrator – which may have been intercepted and tampered with before the system administrator received them (ideally, your vendor would send the update as a digitally-signed message to guard against undetected tampering);
E. Hard copy of configuration information, possibly listing passwords and critical configuration details, being acquired by unauthorized personnel ("dumpster diving" can reveal this, and other, proprietary information).

Recently, PBX's incorporating Computer and Telephony functionality have become popular, some sticking to their traditional physical line "switching" routes, and others incorporating VoIP (voice over IP functionality). Many of these systems, such as the Altigen[95] communications server runs on Windows 2000 and other widely-available operating systems, integrating telephony with applications like SQL Server and Microsoft Exchange to perform functions like "screen pops" (popping up customer information, looked up from the incoming caller ID information, on a service rep's screen) or "call routing" (sending a call to an available tech support representative with the appropriate skills, based on the type of trouble call). Both these new computer telephony solutions and traditional PBX's may be network-enabled, to allow console access and transfer of PBX-collected information like Call Detail Records (CDR's) to another computer, perhaps for billing purposes.

[95] http://www.altigen.com/

If you're using one of these systems, and it is connected to your LAN, be aware that anyone breaking into your LAN may also eventually gain access to your communications server. So, the standard precautions, of setting extremely strong passwords, monitoring the server for configuration changes, following up on unusual logins, etc., apply. When your communications server runs on a widely available OS, also add automated virus checking to your list of security measures.

It can also use a variety of communication protocols, some of which offer encryption. If you are using the most recent versions of Windows (Windows 2000 SP2 or higher), you can configure your RAS server to require that connections use strong 128-bit encryption.[96]

A handy feature of RAS is that it can be configured to log incoming connections, giving you a record of when your network was accessed and by whom. If suddenly someone who never accesses the network via RAS hits it four times in the middle of the night, you might want to verify with that user that they did indeed call in, to make sure that an outsider didn't just guess their password and log in with false credentials.

3.1.8 VPN

A Virtual Private Network, or VPN, simulates a private network over a public network (or less secure private network), allowing multiple sites to communicate securely. In this way, a VPN supports traditional host-based remote access (dial-in from a PC), LAN-to-LAN access (wide area networking) and an extra level of communications security within an intranet (encrypting sensitive traffic so that it cannot be "sniffed" by personnel using your internal network).

Companies often establish VPNs which run over the Internet as a more cost effective, lower administrative overhead, more scalable alternative to a traditional (non-virtual) private network. VPNs are more cost effective because organizations can connect any physical locations together without requiring long distance data calls via modem or leasing expensive private communications lines between sites, equipping these sites with the appropriate communications hardware to support the private network. This saves the organization the monthly line and long distance costs.

[96] Worsham, Michael, "Beef up RAS security," http://searchsystemsmanagement.techtarget.com/tip/1,289483,sid20_gci788334,00.html, Dec 21, 2001.

VPNs offer lower administrative overhead and improved scalability because having users access the VPN through the Internet enables you to assign responsibility for that connectivity to the users' ISP's, rather than taking it on yourself. Additionally, your data center won't need to maintain the equipment that would be involved with the private lines, which could include high-speed CSU/DSUs and a router port for each private line, a modem bank for those network participants who do not connect via leased lines, etc. This frees technicians to concentrate on other issues and eliminates the time and cost involved in upgrading your communications hardware when technology advances or your network expands.

Primary security features offered by VPNs include:

A. User authentication and authorization, usually through digital certificates or passwords, combined with policies, to ensure that only authorized personnel can access appropriate portions of the VPN;
B. Secure communications, by way of an encrypted communication tunnel, which keeps communication private, and tamper-proof, as it is routed through the Internet;
C. Protocol encapsulation, allowing protocols outside the Internet's standard TCP/IP family, like IPX and AppleTalk, to be sent over the Internet;
D. Address space isolation, allowing use of private address ranges (and other internal Internet addresses to which your router may block access) within your VPN, even for hosts located outside the boundaries of your internal network;
E. Integration with firewall technology (with some VPN's), for ease of administration.

A number of popular protocols are used by VPNs, including PPTP, L2TP, Ipsec and SSH. Of these, IPsec is generally regarded as the most popular protocol for VPNs, with PPTP and L2TP following close behind, generally in Windows-based environments, and SSH popular in UNIX-based networks. For more information on these protocols, see the section on Remote Access, which discusses each in depth.

VPNs can be implemented in firewalls, dedicated hardware or software. For example, a popular firewall solution that includes VPN capability is the Pix, by Cisco Systems[97]. Popular dedicated hardware solutions include Sonicwall[98], Netscreen (for both small and enterprise VPN's)[99], and Nokia VPN[100]. There are also software-based solutions that integrate with firewalls, such as Checkpoint. Finally, some are software-based solutions such as SSH and SSL. The interesting thing about some of the software solutions (including SSH and SSL) started out as protocols that provided encryption for specific applications, such as remote terminal access (SSH) and SSL (Web browsing), but were later found to be effective VPN protocols as well.

One security issue with VPNs is that your communication is at the mercy of the Internet. While communication using a VPN is generally secure, it's not without potential problems. If an area of the net is slow one afternoon, your VPN communication speed for any communications traveling across that portion of the net, will be slow. If there's an outage at an ISP, one or more sites may be temporarily knocked off your VPN. If an attacker decides to DOS one or more routers at sites involved in your VPN, their communications will be impaired. If a script kiddie exploitable flaw that crashes your particular brand of VPN is found, prepare to stop what you're doing periodically and reset the VPN, until the vendor fixes the flaw.

Another issue with VPNs is that, depending on the technology you use, you may find that some information about your network and communications, such as packets' source and destination IP addresses, is not encrypted. If it is important to you to keep this information private, make sure that the VPN you select will do so, or perhaps incorporate the use of NAT in your network so that the only exposed address within your network is the NAT server.

On a related note, another issue with VPNs is their potential susceptibility to "man in the middle" attacks, which intercept the communication, take note of the IP addresses involved and then impersonate either the client or server side.

---

[97] http://www.cisco.com

[98]

[99] http://www.netscreen.com

[100] http://www.nokia.com/vpn

Another security issue with VPNs is related to encryption. If an attacker knows the information transmitted across your VPN is very valuable (perhaps credit card or wire transfer information), they may be willing to spend considerable amounts of time and money, to attempt to break the encryption being used. Practically speaking, most VPN encryption mechanisms are sufficiently secure and DES was considered not reasonably breakable, 30 years ago. Be aware that the passage of time may render certain encryption mechanisms ineffective.

Before implementing a VPN, consider whether interoperability is a factor, as it often is when your VPN includes an extranet consisting of your network and your business partners' networks. In that situation, it may not be feasible to merely dictate to your partners that they must use certain VPN hardware or software to communicate with your site. Your options may be limited by which VPNs the hardware or software they're already using supports.

For more information about VPNs, see Virtual Private Networking – a view from the trenches by Burce Perlmutter and Jonathan Zarkower, published by Prentice-Hall.

3.1.9 IDS

An Intrusion Detection System (IDS) is a system for detecting attempts to break into or misuse a system or network.[101] They are, in effect, a burglar alarm for you. What kind of activities can intrusion detection systems identify? It varies with the particular IDS, but can include detection of network scans (in which an attacker is sending traffic to your network to map it, or find systems with known vulnerabilities), spoofing attempts, denial of service attacks, attempts to connect to unsupported services, and receipt of network communication packets that violate standards.

You would position IDS anywhere on your network that you want to look for suspicious activity. Many home systems (and some sensitive network hosts) run IDS. In the case of a home system, it usually doesn't make sense to add a completely separate system for this task. In the case of a particularly sensitive business system, the administrators may want to build in as many layers of detection as possible, to enhance security.

[101] Network Intrusion Detection Systems FAQ, http://www.robertgraham.com/pubs/network-intrusion-detection.html#1.1, Version 0.8.3, March 21, 2000.

At other times, you might position IDS between your company's external (Internet-connected) firewall and its internal (internal network-connected) firewall, to detect any unwanted traffic that got through the first firewall, or within the internal network itself, if you're more concerned about monitoring for intrusions into your internal network. Think about this the way you'd think about burglar alarm sensors. Perhaps you want motion-detectors within your yard, which turn on outside lights when movement is detected. But you'd probably reserve the sensors which actually ring the alarm, for inside your home, near doors and windows lest you be awakened by loud beeping every time a dog runs across your front lawn, or you run down to the kitchen for a snack, in the middle of the night! Then, if you ran a "Bed and Breakfast" in your home and were concerned about security, you might place additional sensors in private areas of your home, much like a company might run an IDS on their internal network NOT for the purpose of catching those who are outside trying to get in, but to monitor for suspicious activities by in-house personnel.

The exact details of how IDS detects potential break-ins varies depending on the IDS in use. IDSs can employ a variety of techniques to determine what constitutes an "intrusion". These will be discussed in more detail later, when we get into the specifics of network and host based IDSs.

IDSs do more than just DETECT intrusions – they react to them as well. To go back to our burglar alarm analogy, an alarm would be of limited use if all it did was keep a counter each time a window is broken, without taking any other actions to inform responsible parties or discourage the intruder. Intrusion Detection Systems can take many different types of actions when a break-in is detected, including:

A. Logging the suspicious behavior, so that a record is available in the future (most IDSs use a standardized, documented log format such as Syslog, tcpdump, or Snort for ease of mining data from the logs);
B. Paging or emailing network administrators;
C. Interfacing with a local custom application to perform site-specific tasks like entering the data for the suspicious activity into a site trouble reporting system;
D. Taking corrective action to minimize exposure, such as killing server processes on the system which are being used by the intruder, disallowing access to the network from the attacker's IP address, altering traffic routing, etc;
E. Automatically performing reconnaissance on the attacker, trying to find out more about who they are

As the number of TCP network intrusions has increased over the years, more and more IDSs have been developed, both commercial and non-commercial. Examples of commercial intrusion detection systems you might want to research include:

A. RealSecure, by Internet Security Systems [102]
B. Dragon, by Enterasys [103]
C. NFR, by Network Flight Recorder [104] (also available in a free research version)

The most popular open source IDS, and possibly the most popular one period, is Snort[105]. SANS Intrusion Detection wizard Stephen Northcutt calls it, "the most advanced intrusion detection system money cannot buy."[106] A comprehensive list of public domain and shareware IDS software can be found at the COAST Intrusion Detection System.

Resources site[107]. If you don't require network-wide monitoring for suspicious activity, check out the following, which are only some of the packages in the growing category of freeware, sometimes open source, IDSs:

A. PortSentry, by Psionic Technologies[108]
B. Tripwire, by the Tripwire open source team[109] (also available in a commercial version)
C. Tcpwrappers, by Wietse Venema[110]

The different IDSs have subtly different capabilities, strengths and weaknesses, so before committing to one, do your research! Make sure that the one you've selected does in fact detect the kinds of intrusions you care about, and that the system is able to respond with the types of actions you need. For example, if you need for the system to dial a pager, make sure that it can do these or at least that you can find a pager dialing program out on the net (they're there…) and that the system can run it.

102

103

104

105 http://www.snort.org

106 Northcutt, Stephen, Donald McLachlan, Judy Novack, Network Intrusion Detection: An Analyst's Handbook (2nd Edition), New Riders.

107 http://www.cs.purdue.edu/coast/ids

108 .

109

110

Be cautious in reading reviews. This industry is evolving rapidly, what with new "cracking" techniques constantly being developed, and new detection measures being created to identify them, so review comments true about the last version of a package may or may not still apply to the current version. When in doubt, check with the vendor.

This is not the last you'll be hearing about IDSs, which show up again in more detail, later in this major section. For more information, do check out Robert Graham's excellent FAQ on this subject, including questions to ask an IDS vendor, further resources on the subject, and ways attackers attempt to avoid detection by IDSs.

3.1.10 Network Monitoring/Diagnostic

Most networks larger than small workgroups incorporate network monitoring or diagnostic tools, to assist with network management.

These may be devices to monitor and diagnose hardware issues or physical cable issues, such as TDR's (Time Domain Reflectometers). There are also software packages to monitor network traffic, such as the tcpdump network packet "sniffer" (which listens on the network for packets of interest and records them to a log), or diagnose network configuration glitches (such as the "dig" and "nslookup" tools to investigate name server issues, "traceroute" to check a packet's route across the Internet from source to destination, "netstat" to view current connections on a system, "ipconfig" to check which nameserver a system is using for name-to-IP-address resolution, etc.). A "port scanner" is a software-based monitoring tool which will inspect a network and report the hosts on it and which ports/services are available on each host. Some, like nmap[111], will report the OS's run by systems on the network, and even whether or not packet filters or other monitoring tools are present on the network.

Another type of software monitoring tool would be system/network security scanners like ISS and SATAN which will probe your network or systems for known security issues, and report any that are found. Intrusion Detection Systems, which we've just discussed, is also another form of network monitoring. From an attacker's point of view, these tools are interesting for the reconnaissance information they can provide, such as information about what applications are run on the network, user ID's and passwords, how well-connected a site is, and of course, proprietary data.

[111](http://www.insecure.org/nmap)

It's not enough to protect access to monitoring and diagnostic tools, when possible. You must also protect the information they collect from unauthorized access. If an intruder knows, or can determine, where your network monitoring logs are stored, and they gain access to them, they could view your logs or even remove all traces of their visit to your network. So, it's prudent to develop a policy for backing up important system logs to off-line storage on a regular basis, in the name of preserving potential evidence.

Since it's not feasible to guarantee that no one will ever run packet sniffer software on your network, remember that no packet on your network is immune from being captured by a sniffer, and keep the following in mind:

A. Do not send sensitive information across the network unencrypted (this includes email, files saved to servers, credit card information submitted from a web page to an application server, etc.);
B. Use challenge/response authentication techniques instead of those that send passwords in clear text or encryption, in order to minimize opportunities for playback attacks and password stealing;
C. Consider probing your network for the presence of unauthorized sniffers periodically, to at least limit the amount of information they gather before being discovered and disconnected.

The "state-of-the-art" in stealthy network monitoring involves monitoring to detect activities like port scanning and monitoring itself. Using techniques like looking for certain streams of packets, or examining packet delays and the responses of systems to specifically constructed packets, it is possible to determine that some sort of monitoring tool is in use on a network. If an attacker detects that monitoring is in use, he may try to "confuse" it by sending many meaningless packets in hopes that his traffic will get lost in the shuffle, or dropped before it is logged by a monitoring system struggling to keep up with the amount of network traffic, thus preventing a log of his activities.

Finally, we want to mention a network management/monitoring protocol (not a tool) that has been in the news lately, due to the security vulnerabilities inherent in it. The SNMP network management protocol is used to query network devices for information such as configuration, traffic and error counter values, etc.

However, it has also been used by hackers, who take advantage of the fact that the primary way an SNMP query authenticates itself to a device is by providing the right "community name" – which the majority of network installations never change from the default, "public", and those who do change it find that any password still leaves SNMP vulnerable to the same attacks that work on any password-access system. Various security flaws in SNMP have recently led vendors to issue software and firmware updates for devices from switches to routers, or to recommend that SNMP be disabled on devices which are not firmware-upgradeable. SNMPv2 will feature improvements to authentication.

3.1.11 Workstations

Workstations are the computers on your network that are most likely to be used by end users. (At least, we hope that your non-system-administrator users don't make a practice of doing their work on servers!) It's a fancy name for a desktop personal computer connected to a network, usually one which uses files, printers or applications located on one or more network server computers. Workstations are used for day-to-day "work" by users, such as creating documents, writing programs, updating spreadsheets, running graphics packages, etc. In "UNIX geek" circles, a distinction is sometimes made between "personal computers" and "workstations", which are considered to be higher-powered PC's, equipped with enhanced graphics capabilities, used for activities like animation or CAD/CAM. But in the corporate networking world, the terms "personal computer" and "workstation" are used interchangeably.

Workstations typically sit inside the corporate network, with access to internal servers, resources like printers, and possibly the Internet. Users' access to the Internet may be restricted by web filtering software, or security measures on corporate firewalls and routers. Unlike a desktop user, who typically has unrestricted access to the Internet from their PC, most workstation users find themselves behind a "guardrail" of some sort.

If you're thinking about security, and have done any work in the help desk field, the fact that these devices are accessed directly by end users should send up a red flag. Remember our point that security is largely a "people" problem? Users can do an absolutely endless variety of interesting things with their computers, which don't contribute to the security of a network. Some of the common security issues created by users sitting at their workstations include:

A. Opening email containing a virus, which infects the user's system and goes on to propagate itself around the network.
B. Downloading and installing unauthorized company applications such as audio/video players or IRC clients that might contain security holes that could be exploited by malicious users inside or outside the network.
C. Sharing their entire drive with no password, because Joe six cubes down wants access to their electronic library of customer support help documentation (this could be a huge issue if the organization has no firewall or router rules in place to block access to the drive from the Internet).
D. Starting unauthorized services such as Microsoft's IIS web server, which may be exploitable by others.
E. Not adequately protecting information stored on their computer (for example, by setting the computer up to auto-login so that anyone who turns the system on can access the contents of it and possibly the network as well, or by not password-protecting sensitive information).
F. Changing their network password to their first name and last initial.
G. Connecting a modem to their network and setting up a "remote access" application like Timbuktu or PC Anywhere, so that they can dial directly into their computer, and thus your network, from home – probably without having to provide a password.
H. Taking the CPU home for the weekend to finish some work (probably because they were read the riot act the last time they installed PC Anywhere).
I. Going home for the night and leaving themselves logged in, with a proprietary document being edited.

Protecting information stored on the computer is somewhat of a challenge, since any time there is physical access to a system (as with most workstations), getting access to its disk is usually no more complex than inserting just the right boot floppy, or at worst, disassembling the box and taking its disk out. An exception to this is if encryption is used, either on the system's most private files, or on the entire disk. Similarly, if they can remove the machine from the office, there's no telling where your physical asset or its proprietary data might end up.

Dot-bomb layoffs and “five-finger severance” have resulted in tens of thousands of dollars of missing computers (and copies of source code) at just two companies one of the authors knows about. And airports today are full of notebooks accidentally left at gates by hurried travelers.

Other workstation-based security issues can be more easily resolved. For example, to protect yourself against flaws in workstation OS and application software, keep all software installed on users’ machines up-to-date with the latest security fixes. Also, anti-virus software can scan all incoming email for potential viruses and can be employed on individual workstations to detect and contain a virus if one should appear. Windows system policies and careful assignment of user rights (no, every user does NOT have to be in the Power Users group) can prevent users from installing programs, controlling services, sharing drives to the network, etc. and set up screen savers to lock the workstation after 10 minutes without use. Restrictions can be set on user passwords to mandate an amount of time between password changes, strength of password, etc.

It’s also possible to address some potential security issues with changes in hardware configurations. You might consider removing floppy drives, if your systems don’t absolutely require them, in order to eliminate a way in which data can be copied from the network. (Really, when was the last time you needed to use a floppy on a network workstation? And when you did, would it have been that great an inconvenience to go visit one of the few machines that had one?) And to keep users from booting from unauthorized CD-ROM’s or floppies and gaining access to the machine’s disk, consider setting the boot order in the machine’s BIOS to exclude the CD-ROM and floppy, and password protect the BIOS so that the user can’t change the boot order back him or herself. Don’t want your users attaching modems? You can make the task more difficult by removing or disabling the workstation’s serial port.

Some of the security measures for workstations still come down to person-to-person communication. Let users know they’re responsible for their system, what happens on it and is stored in it. Inform users of your organization’s computer use policies so that users don’t unintentionally violate them and know the penalties if they deliberately do so.

3.1.12 Servers

In the context of hardware, a server is a device on the network that manages network resources[112] such as files, printers, email, Usenet News or databases.

Servers can be positioned on an organization's internal network or in the more public area of its network, called the DMZ that is accessible by Internet users. Of course, a high profile security issue is that Internet users known as "script kiddies" regularly probe any Internet-accessible servers for known holes. It is also possible for internal servers to be attacked inside the corporate network, for instance, by a disgruntled employee seeking access to salary data. No matter where the server is located on the network, it is potentially susceptible to unauthorized access. We mention this because sometimes organizations focus primarily on external threats and fail to consider the possibility of internal threats. The degree of security applied to internal servers depends both on corporate culture and the sensitivity and value of the information on those servers. It's legitimate to decide that too much security on internal servers would have an adverse affect on sharing of information, researcher freedom, or developer efficiency.

Servers can be general-purpose computers dedicated to managing one or more resources, or special-purpose "boxes" designed to do a few things well. General-purpose servers can be very similar to workstations, using similar CPU's, memory, and peripherals. However, there are also some key differences between servers and workstations. Servers typically offer faster I/O (such as SCSI instead of IDE disks), more disk storage, more RAM and support for more processors than a workstation. With a server that is a general-purpose computer, you can change the functionality of the server by changing the software installed on it. This week, it's a database server running Oracle on Windows. Next week, maybe it's a file server running Linux. Contrast this to a special-purpose, dedicated server, which will probably only ever do what it was designed to do, be that acting as a fax server, a file server, etc.

(Hackers, don't start. We know that where there's a will, there's a way. But in most cases, the vendor won't explicitly support it!)

[112] Definition of "server", http://www.webopedia.com

The same flexibility offered by general-purpose servers can be their security downfall. If you can install other software on a general-purpose server, like a Linux or Windows system, so can an intruder who gains the sufficient privileges. They can install Denial of Service agents, their own monitoring software to alert them of an administrator's presence on the system, Trojan horse programs in place of original system utilities, etc. On the other side of the customizability spectrum are the special-purpose, dedicated boxes.

Most likely, an intruder who gains access to one of these won't be installing his own "cracker" programs on it, because they either run a strange proprietary OS, or hide a normal one (like Linux) behind layers of menus and don't allow direct access to the OS. However, what this means is that if a security bug is identified in the firmware for this special-purpose server, you'll have to live with it (or do without some or all of the server's functionality) until the vendor supplies a patch. At least in the general-purpose server case, if a security flaw is found in one vendor's FTP server, you have the option of installing a different one.

Physically, servers are often kept in more secure physical environments than workstations, because they are generally considered more valuable (in terms of their function, data, or hardware itself), in need of closer monitoring, or in need of being located centrally for ease of performing tasks like backups which can require physical access to the machine if the backup is not performed over the net. This means that the systems aren't out in the middle of the office, where any user can access the keyboard of a server that a systems administrator didn't log out of, or boot it with a floppy disk or CD-ROM that would let them access the server's contents without a proper password.

Another consideration related to server hardware is uptime, or the amount of time the server is performing its job vs. the amount of time it's down for maintenance, repair, or due to software or hardware glitches. If a workstation goes down, one user is usually inconvenienced. If a server goes down, potentially hundreds (or more) of users can be inconvenienced by an inability to get their work done, resulting in missed deadlines, idle employees and time spent finding workarounds to the situation.

Or millions of dollars of revenue can be lost, as Ebay found out when a 22-hour outage over June 10 and 11, 1999, cost them an estimated $3 million to $5 million in listing fees refunded to sellers and caused a drop of nearly 30% in their stock price.[113] Since the stakes are high, there is often tremendous incentive to do whatever it takes, to increase uptime.

Some common measures to increase uptime include:

A. Connecting servers to UPS equipment … or even diesel generators.
B. Purchasing "high availability" servers that support redundant hardware, such as RAID arrays, multiple network cards, power supplies and fans, and hot swappable components that can be replaced without shutting the system down.
C. Clustering servers so that if one goes down, the others can take over its workload.

Specific security measures for different types of servers will be discussed in more detail in later sections.

3.1.13 Mobile Devices

Mobile Devices participating in your network include everything from Pocket PC and Palm Pilot handheld organizers, to notebook computers, to RF scanners used for managing inventory.

The key security vulnerabilities of mobile devices are that their portability can lead to easier theft or loss (thus loss of the data stored on the device) and that they usually communicate with the network via a wireless communication mechanism.

To guard against loss due to the disappearance of the device, you can take measures such as:

Ensure that data collected by the mobile device is uploaded to the network as quickly as possible, to minimize the amount of data lost if the device goes AWOL.

[113] Breymer, Lynn, "Just what I didn't need – a little downtime", http://home.techies.com/Common/Career/199907/Main/Verge070199_m.jsp

Password access to the device, if possible, to make it more difficult for an unauthorized person to view the data on it.

Install encryption software and make sure that any proprietary information stored on the device is encrypted.

When using a mobile device over the airwaves, make sure that any sensitive communications are encrypted. If you're communicating via 802.11 or other technologies that can be "broken" by curious attackers, you may want to employ an additional layer of encryption on the connection as well.

3.2 Media

Security+ looks at two types of media: network communications media that allows devices on the network to communicate with each other and removable media for data storage. Information can also be exchanged via wireless networking, which is not considered physical "media."

3.2.1 Coax

The initial cabling for Ethernet networks was coax. Thicknet, or 10Base5 is used for longer network distances up to 300 meters and up to 10mb/sec speeds. You can recognize it by its large cable diameter and typically orange or bright blue color. Thinnet, or 10Base2 is used for shorter network distances up to 183 meters and up to 10mb/sec speeds. Thinnet has a smaller diameter than Thicknet; is more flexible; and, is easier to work with than Thicknet. For example, to connect a computer to a Thicknet segment, you have to install a transceiver onto the Thicknet cable, and then connect the computer to the transceiver with a transceiver cable. In contrast, to connect a computer to a Thinnet segment, you only have to add another length of Thinnet cable to your existing network, put a "Tee" connector on it, then connect the T to the computer network card's 10Base2 port and possibly put a terminating resistor on one end of the T if the computer was at the end of the bus network. For the most part, coax network segments have been replaced by fiber and UTP/STP. However, coax may still be in use within some organizations.

The primary security consideration of coax-based network segments is reliability. Thinnet was used in many small networks in the late 1980's and early 1990's because it was the least expensive alternative at the time, and easier to work with than the "frozen garden hose"-like Thicknet. However, Thinnet is also less reliable than other alternatives. Because coax is used to implement bus topology networks, a fault could bring down the network. 10Base2 cabling, which looks very similar to cable TV wiring, is very flexible and susceptible to problems due to kinks, being run over by an office chair, crushed by a server box, etc. While 10Base2 cable looks like TV cable, it is NOT the same thing. If TV cable is used for a Thinnet network, it may appear to work (for a while at least), but your network will experience performance and packet error issues. In addition to issues with the cable itself, there could also be issues with connectors and terminators.

A loose connection or missing terminator could cause erratic network performance, with reduced speed, high frame transmission error counts, or even a lack of network connectivity as the result. Similarly, with Thicknet, a malfunctioning transceiver could also cause excessive packet transmissions, frame transmission errors, etc. Since a fault anywhere along the cable is likely to affect the entire segment, coax-based networks don't have the same level of fault tolerance as UTP/STP based networks. Usually time-consuming trial and error or an expensive diagnostic device like a TDR (Time Domain Reflectometer) is required to locate cable or connector errors on coax-based networks.

Another security consideration for 10Base2 coax-based network segments is that it is possible for anyone with access to the cable, to "tap" into it and add an unauthorized device to the network. It's easy to tap into a Thinnet-based segment (by adding another length of cable with a Tee connector, anywhere along the network), and also possible, though a bit more involved, to tap into Thicknet. Since the device connections on a bus network don't occur in a centralized location (as with hub-based systems), but rather at any convenient point along the cable, it is more difficult to physically monitor for these unauthorized devices. To compensate for this, if your network includes coax segments, you might want to regularly scan those network segments with a network scanner program like nmap (mentioned in an earlier section) and flag any unknown devices.

3.2.2 UTP/STP

In recent years, most workstation connectivity (and some server connectivity) has been accomplished with twisted pair cabling – either UTP (unshielded twisted pair) or STP (shielded twisted pair). It is easier to work with, generally less expensive today, and supports higher speeds (up to 1gbit/sec) than the coax Thicknet and Thinnet approaches.

STP is typically used for Token Ring network cabling and UTP for twisted-pair Ethernet.

In Ethernet networks, twisted pair cables with RJ-14 jacks (8-wire connectors that look like large RJ-11 telephone jacks) are used to connect computers with hubs or switches, which connect together into local area networks. Twisted pair can be used to wire 10BaseT (10mbit/sec), 100BaseTX (100mbit/sec) and 10000BaseT (1gbit/sec) networks.

Common physical cable standards are Cat3 (up to 10mbit/sec), Cat5 (generally 10mbit/sec, sometimes up to 100mbit/sec), Cat5e and Cat7 (up to 100mbit/sec)

In Token Ring networks typically found in organizations whose technology is IBM-based, Type 1 STP cables are normally used to connect computers to the network.

With twisted pair networks, the main security concern is not so much direct tapping into existing cable, but adding devices to the network by plugging easily-available cables into unused ports on existing hubs, switches and routers. You can physically inspect the hubs, switches and routers on your network, to ensure that only the ports you expect are being used, are in use. You can also employ software-based network monitoring techniques to detect unauthorized nodes, as on coax-based network segments. Another security vulnerability of twisted pair (and coax-based) networks is that since they transmit information via electric pulses, they emit small amounts of electromagnetic radiation that could conceivably be eavesdropped on, by someone with the right equipment and sufficient interest.

Networks built with twisted pair cable tend to have greater fault tolerance than those built with coax. Because devices are physically arranged in a "star" network configuration, a problem with the connection of one device to the network, or one port on a hub, typically does not interfere with the rest of the network. As with Thinnet coax, many kinds of twisted pair cables look alike, but do not work alike. Be sure that you are not using Cat3 cabling to wire a gigabit network, or that if you use UTP on your STP-based Token Ring, you install the appropriate media filters to compensate for the extra noise on a UTP cable.

When buying twisted pair cabling, you need to take into account WHERE you are installing it. When running twisted pair cable in certain locations (such as in a suspended ceiling or within conduit), you need to use a more expensive version, called "plenum cable", which has a different type of outer jacket to comply with building and fire codes.

3.2.3 Fiber

Before scientists ratcheted up the speed of twisted pair technologies, the main option available to those who wanted faster networking, was fiber. Fiber-optic cabling uses light pulses instead of electrons, as in copper-based technologies, to transmit information.

There are a variety of cable standards, supporting 100mbit/sec (100BaseFX) and 1gbit/sec (1000BaseSX, 1000BaseLX, 1000BaseLH, and 1000BaseZH) Ethernet. A number of connectors may be used as well, from the older SMA connector, to the newer, easier-to-use ST and SC connectors.[114]

Today fiber is often the cabling of choice within data center environments used to connect systems into fault-tolerant clusters, attach storage area networks, etc. It is also in use where an excess of electrical "noise" in the environment makes twisted pair impractical, such as a factory floor. Fiber also has some additional data-security advantages over copper-based technologies. The cable itself is difficult to "tap" into by someone who wants to add an unauthorized node to the network, and since fiber does not radiate electromagnetic energy, emissions cannot be intercepted.[115]

Fiber-optic cable can be challenging to install and maintain, depending on the precise technologies in use. Because fiber-optics involves passing light around a network, it is important for the connections to be as perfect as possible, to minimize the loss of light, and thus signal, at the connection. In earlier days, fiber optic cable breaks meant major efforts to polish the ends of the broken cable and reattach them, but fortunately, technology has advanced and vendors have introduced easier ways to deal with fiber-optic cable breaks.

As with twisted pair, if you are running fiber in certain locations like suspended ceilings or conduit, be sure that you are using a plenum-certified variety of cable.

Outside the data center, you may also see fiber-optic cable used instead of standard video cable in high-security theft or intrusion alarm systems.

3.2.4 Removable media

Removable media refers to data storage media that is somewhat portable, that is, it is not permanently fixed to a server or workstation. It can be a security consideration for the following reasons:

A. It enables data to be transported outside your physical network, without being filtered through a device like an internet gateway, possibly allowing unauthorized removal of data from the site.

---

[114] Houser, Tcat and Helen O'Boyle, Inside Scoop to Network+ Certification, BFQ Press, 2002.
[115] "Cable Construction," NetOptics, http://www.netoptics.com/5.html.

B. It is used to store data securely, for disaster recovery backup and archival purposes, with the expectation that the data can be retrieved from the media at a later date, if needed (you should consider the life of the media, and the ease with which data can be restored).

On the subject of data storage, depending on your security requirements, you may opt to encrypt data on removable media so that even if it is lost or stolen, unauthorized personnel still cannot read it.

3.2.4.1 Tape

Tape is the traditional high-volume backup media. It has historically had the advantage of providing the most storage for the least cost, at an acceptable (though not incredible) speed. Because of its ability to hold large amounts of material, it was also often used prior to the popularity of CD-ROM technology, for the distribution of software or data from one organization to another.

Common capacities and form factors of tape include:

A. 4mm DAT (Digital Audio Tape, which is used for more than just audio), holding 4, 8 (DDS-2), 24 (DDS-3), or 20/40 (DDS-4) GB
B. DLT (Digital Linear Tape), holding 10/15/20/30 (DLT III) 40/80 (DLT IV) GB, with the new Super DLT having 110/220GB capacities
C. 8mm (“Exabyte”), at many capacities from 2.5GB to 50GB
D. Travan (a standard for HP/Colorado tape drives), at many capacities up to 40GB
E. QIC (Quarter Inch Cartridge, an older standard), holding from 60MB to 2GB
F. 9-track (the large spools of “reel-to-reel” tape seen in all those Grade-B movies)

When choosing a type of media and removable media drive, consider the duty cycle of the equipment (how many hours’ use it can take per month/year), the life of the media (some media will last a couple years, others will last decades) and your backup needs so that you neither overbuy (going for an expensive solution that’s more than you need) or under buy (and end up replacing your drive every 3 months, and tending backups for hours each night).

The drive and media manufacturers are the best source of this information. Note that different brands of media, for the same type of drive, often have different lifetimes.

A characteristic that tape shares with most other magnetic media is that it can be rewritten. When you no longer need the information on the tape, you just reuse the tape and the old information is written over by new. As with other magnetic media, if you don't want to overwrite the tape, but do want to erase the data on it, a large magnet (sometimes called a bulk eraser or de-gasser) can be used to disrupt the magnetic storage mechanism, effectively "erasing" the tape.

From a security standpoint, be aware of the difficulty of truly deleting information from magnetic media. Some types of disk and tape that are overwritten or bulk erased, such as a DAT or 8mm tape, may still be able to have data extracted through use of special hardware and software. How can this be? It's due to the magnetic properties of the media. When data is written to a specific location on magnetic media, the read/write heads do not always align with complete precision over the same exact spot, each time it is written to, and the strength of the magnetic field is not always completely uniform. According to Peter Gutmann, "The recovery of at least one or two layers of overwritten data isn't too hard to perform by reading the signal from the analog head electronics with a high-quality digital sampling oscilloscope, downloading the sampled waveform to a PC, and analyzing it in software to recover the previously recorded signal"[116], and other methods will recover data from even longer ago. For that matter, evidence suggests that some media may become more difficult to erase, thus physical destruction of media or use of a "secure delete" program is recommended to maintain confidentiality. "Secure delete" programs work by performing multiple overwrite passes with disk caching disabled, among other techniques.[117] Some options are SDel, available include:

Secure Delete[118] Windows
Macintosh[119]
Secure-delete UNIX/Linux[120]

[116] Gutmann, Peter, "Secure Deletion of Data from Magnetic and Solid-State Memory", Sixth Usenix Security Symposium Proceedings, San Jose, California, July 22-25, 1996.
[117] SDel product home page, JRTwine Software,
[118]http://www.jrtwine.com/Products/SDEL/
[119] http://www.aladdinsys.com
[120]

Magnetic media is also prone to damage by aging and extreme temperatures. Therefore, many organizations use temperature-controlled off-site data storage facilities (did you know that if you want to pony up the $$, you can store your data in Fort Knox?!), and have a policy of restoring and rewriting critical archival data before the useful life of the media is reached. If you need to read, write or store media in other than typical office conditions, check the removable media technology, and the specific brand of media you purchase, to make sure that it will function properly in your environment.

Another concern regarding the use of tapes and diskettes as backup or archival media is that unlike hard disks, tapes and diskettes do not have self-contained read/write heads (the part of the drive that translates between magnetic impulses on the media, and the 0's and 1's understood by computers). This matters when considering restorability because it is possible, if the data was written on a drive whose heads are out of alignment, and then the restore is attempted on a drive with properly aligned heads that the data will not be readable, and in fact, will only be readable by the drive on which it was written. More than a few notebook owners, one of your authors included, know of at least one road warrior notebook whose floppy drive wrote diskettes readable only by it! The time to find this out, so that the drive can be replaced or repaired, is BEFORE you need to read the data off the media at another site. Therefore, if using magnetic media for backup/archival purposes, build into your operational schedule regular "restore" tests, and use a different system and different drive to perform the test.

A final concern about all media used for archiving is that it's no use to have media that lasts for 50 years, if 50 years from now, you don't have a way to read the data off of it. For this reason, many companies also store a duplicate tape/disk/CD-ROM unit with the media itself – and sometimes an entire PC, if the device uses a non-standard interface like a special PCI card rather than SCSI, IDE, etc.

3.2.4.2 CDR

Recordable (CD-R, or Compact Disc Recordable) and Rewritable (CD-RW, or Compact Disk-Rewritable) CD-ROM technology has become popular for moderate-sized backups, due to the modest cost of drives and media. Other interesting uses have appeared as well[121].

[121][0]www.m4postcard.co.uk[0]

Note that CD-R allows for an area of the disc to be written once; CD-RW allows any area of the disc to be rewritten multiple times. Unlike tape and hard discs, CDR is based on optical technology with a laser used to create "pits" and "lands" in the media to store data.

CD has largely replaced both tape and diskette as the software distribution media of choice, because of the "Orange Book" physical standards which help ensure compatibility, capacity (usually 650meg, sometimes 700meg, per disk), adoption of standardized CD file system formats such as ISO and El Torito by vendors and the wide availability of CD-ROM drives. CD-R's and CD-RWs are usually the size of a typical audio CD, however, some manufacturers make them in "interesting" formats such as business-card-sized CD's, and "cd single" sized CD's which have a lower capacity than full-sized CD's.

A note to those buying media: CD-R's and CD-RW's are not quite created equal. There are "fast" discs (supporting fast write speeds like 24x) and "slow" discs (supporting slower write speeds like 4x), and varying levels of media quality. You generally get what you pay for here. Those bargain CD-R's may only be burnable at low speed (which increases the time it takes to write the info to the disc), or may have a very short life. For one-off temporary-use copies, bargain media may be sufficient. For backups, it almost certainly is not.

The composition of the media varies, depending on brand and quality. It consists of a reflective layer (either a silvery alloy, or 24K gold) and can include:

A. Cyanine dye (blue)
B. phthalocyanine dye (aqua)
C. Metallized azo dye (dark blue)
D. Formazan dye (light green)

The type of reflective layer and dye may have a bearing on how long the media lasts (for example, phthalocyanine dye based CD's are less sensitive to sunlight and UV radiation, while cyanine dye based CD's are more forgiving of read/write power variations between drives). Additionally, some CD's have an extra coating to help the CD resist scratches. Since scratching is one of the easiest ways to damage a CD-R or CD-RW, it is recommended that full-disc labels be used, to cover the top of the disc, to protect it from scratches and prolong its life.

How long will a CD-R or CD-RW last? This is a matter for some debate, since the claims now being made for some types of discs, such as 75 years for cyanine dye, 100 years for phthalocyanine dye on gold discs or even 200 years for phthalocyanine dye on "platinum" discs. Unrecorded (blank) discs are estimated to have a shelf life of 5-10 years. On the care and keeping of CD-R's and CD-RWs, CD-R FAQ recommends, "keep them in a cool, dark, dry place and they will probably live longer than you do," because aside from scratching, the main enemies of CD-R's are heat, humidity and direct sunlight (CDs, not being magnetic media, are not sensitive to magnetic fields).[122]

Along these lines, what if you have CD-Rs or CD-RWs containing proprietary data, which you want to render unreadable before disposing of? Again, the CD-R FAQ has suggestions, such as scraping off the reflective layer with something sharp (warning: this may still leave data on the dye residue), running the disc through a sander, etc. Although not a mil-spec procedure for rendering all data on a CD unreadable, a combination of scratching the surface and breaking the CD into pieces seems to be popular. Since data is redundantly written to multiple locations on a CD, if you do this, make sure you've done more than make a few token scratches on the media.

3.2.4.3 Hard drives

Recently, with the dropping prices of high-capacity hard disks (100 GB and up), they've become a viable off-line as well as on-line (via RAID) backup option. Want to send data off-site? Many servers are already equipped with removable hard drive bays. You can often identify them by the row of disc-sized panels with handles, on the front of the server case. For under $100, you can acquire a removable-drive bay, and a caddy that fits inside it, to make your own hard disk-based removable media system. Buy a hard disk of the appropriate capacity, install it in the caddy, and you're set. However, this "homebrew" approach may not give you all of the benefits of a server-removable drive system. For example, many servers feature "hot swappable" hard drive subsystems allowing drives to be removed and replaced on the system while it is running without affecting system operation.

[122] McFadden, Andy, CD-Recordable FAQ, http://www.cdfaq.org

In contrast with the other media solutions we've looked at so far, a hard drive contains both media and the read/write mechanism (tapes and CD-R's are just media … the drive contains the read/write mechanism). Because of the mechanical nature of the read/write mechanism, with heads hovering just barely above the surface of the media platters within the drive, hard drives are very sensitive to shock. For instance, dropping a hard drive is very likely to result in a non-working drive (though some of the data may still be able to be recovered), but dropping a CD or tape is likely to be a non-event.

The most common types of hard drives today are IDE (used on most network workstations), and SCSI (used on most network servers), with IDE as the low-cost leader and SCSI as the performance leader. Again, as with tape, hard drives are magnetic media, and are sensitive to magnetic fields, thus care should be taken to keep them away from large magnets – including the ones in your big, bad computer speakers!

Hard drives are available in a variety of form factors (from PCMCIA-card sized drives used in notebooks, to 3.5" drives often used in servers), capacities and speeds. In general, the faster the drive, the more expensive, and the more heat it generates. The heat generated by a hard drive can be a concern because it can affect the reliability and lifetime of any part of the computer that is temperature-sensitive (that is, most of them). Therefore, many users who want fast hard drives opt for larger cases with better airflow, and may add extra fans to the computer case, to help cool the system.

Unlike CD-R's, which cannot be erased once written, and tapes and diskettes which have mechanical protection against accidental overwrite (if the user wishes to employ it), most hard disks cannot be "write-protected" in hardware, so there's always a risk that a critical piece of data will be overwritten or erased with a "format c:" or "rm *".

Many hard drives have a MTBF (mean time between failure) rating, to allow you to compare the useful life of the drive with other similar drives since part of building a reliable system is simply choosing parts that are less likely to fail. The MTBF is usually measured in hours of operation. One interesting fact about this is that notebook drives often have a much lower MTBF than workstation or server drives. The manufacturers assume that a notebook drive will not be in use as much per day, as a typical desktop or server machine's drive would be.

Because every day people lose data on hard drives that weren't backed up recently (or at all), the data recovery industry is thriving. Did your disk stop working? Was there a "head crash" in which the read/write heads of the disk contact the media surface with some force, damaging it? Got errant fingers "rm –r /"? There are plenty of "data recovery" companies, who specialize in getting the data off a disk, no matter what's happened to it, that are ready to help you, for a price. Just search the Internet for "data recovery". In many cases, they'll have to totally take apart the drive, rendering it unusable, and you'll have to replace it, but chances are, that's less important to you than getting your data back. For lesser accidents, you can purchase disk recovery software from companies such as OnTrack and Norton, which may help you do the job without enlisting expensive professionals and/or risking sending the disk two states away via a delivery service (ever see what those overnight delivery guys do with boxes when they think no one's looking?).

3.2.4.4 Diskettes

Diskettes are still sometimes used for small software/patch/configuration distribution, and emergency system boots, in some environments. Although not usually positioned as a backup solution in corporate environments, diskettes were at one time the personal backup solution of choice, since every computer was already equipped with a floppy drive and everything else was generally outside the price range of non-business users.

The primary type of diskette in use today is the 3.5" 1.44meg diskette. The disc is covered by a hard, square, plastic shell. To protect the data on the disc, you can adjust the "write tab" on the disc. When inserted in a drive, the drive will detect the position of the write tab, and not allow the disk to be written to (or reformatted) if the user has set the tab appropriately.

Like tape and hard drives, diskettes are also magnetic media, and thus can be affected by EMF, so they should be stored appropriately.

3.2.4.5 Flashcards

A media technology that has been rapidly gaining in popularity for electronic devices like PDA's, MP3 players and digital cameras, as well as for data exchange among computers, is the flash memory card (not to be confused with the very-low-capacity media you might have used in 2nd grade to learn math).

Flashcards offer "persistent" memory, sort of a cross between the no-moving-parts attribute of RAM and the ability to maintain data even without a continuous supply of power offered by magnetic and optical media. There are a number of different types of flashcards in common use today, including:

A. Compact Flash (CF)
B. Smart Media (SM)
C. Memory Stick (MS)
D. Multi-Media Card (MMC)
E. Secure Digital (SD)
F. XD-Picture Card

They are a convenient way to transport large amounts of data in a very small space, usually without worrying about format issues. For example, a CF card written by a digital camera can usually be popped into an adapter on a desktop, and read, as if it were a CD or floppy. Additionally, flash memory cards are more durable than most media, even diskettes, thus making them useful for transporting data. Since they are solid-state devices, there's no mechanical wear and tear, and unlike optical media, no concern about scratching. The main disadvantage to this type of media is high cost (hundreds of dollars for a card which stores less than a CD-R, for example).

Capacities offered by these cards are constantly growing. For example, CF cards are up to 1 GB. In terms of CF, be aware that there are both Type I and Type II cards. If you have an older device that accepts CF, you may not be able to use Type II devices such as the "micro-drive", which is really a miniature hard drive in a CF form factor, not a true CF technology device. CF is very similar to PCMCIA, so a CF card can be used in a standard notebook PCMCIA slot with the simple addition of a 50-pin to 68-pin adapter.

In contrast to CF, which contains both memory and the controller used to access it, the smaller Smart Media cards contain only the memory, laminated onto a plastic card. Because the controller used to access the memory sits outside the card itself, there are occasionally compatibility problems between newer SM cards and older SM devices like digital cameras. While for some time, CF and SD media sizes ran neck-to-neck currently CF cards provide larger storage capacities.

This in combination with SM's less durable construction (it is a very thin card) makes CF a more attractive option when using a device that supports both.

It is anticipated that Smart Media is gradually being replaced by the new XD-Picture Card technology, which allows storage of up to 8 GB in a card the size of a postage stamp. This is interesting from a security point of view because of the sheer volume of information that can be carried around (or out of an environment) very unobtrusively.

Memory Stick is a Sony technology. MS slots are built into many Sony cameras and desktop/notebook computers. This makes MS technology sometimes more convenient than other types since an adapter is often not required as long as you stay within the Sony brand family. Of course, any time you exchange with non-Sony technology, an adapter probably will be required. A newer version, called the Memory Stick Duo, is smaller than the original Memory Stick. While most Memory Sticks sold are of the generic type, there is also a hard-to-find copyright-protected Memory Stick available, which protects the contents from illegal copying.

Secure Digital and Multi Media Cards are closely related. Most devices that support one, also support the other (although there are some devices which are compatible only with one or the other; check the device specs for the final word). SD, currently popular in Pocket PC's, is an evolution of the original MMC. The word "Secure" conjures up images of built-in encryption, passwords and other great security features, but alas, it refers to compatibility with the Secure Digital Music Initiative specifications, which allows the copyright owner to specify if the data can be transferred, and if so, how many times. So, the primary security benefit available from these cards is that of creating software or data repositories that can be used anywhere, but not "pirated" for use in multiple locations at once.[123]

All of these types of cards (except the Memory Stick) usually require an inexpensive adapter when they're used in a computer (whether it's a PCMCIA adapter, or a USB or serial reader device), so when transferring data, make sure that you have an appropriate adapter that can be used to read the card, on the destination computer.

[123] Flash Memory Card Primer, http://www.onepc.net

Note that flashcards are most often used in combination with electronic devices such as digital cameras and MP3 player/recorders, which do not support reading or writing encrypted files. Because of this, you should be aware that data exchanged with these devices via flashcard will be stored unencrypted, and is thus available for access by anyone who obtains the card. If you are using a flash card as storage for a Palm or Pocket PC, there are numerous programs that will encrypt data, so that loss of the device or card does not necessarily mean that the data on it is accessible by unauthorized personnel.

Flashcards tend to allow a more limited number of write/read cycles before failure, than hard discs and RAM. If you plan to use flash memory for frequent data transfer, it's a good idea to estimate the useful life of the media, and make sure that you have spare cards on hand toward the end of its anticipated life so that operations are not interrupted due to bad media.

For more information about flash memory cards, see the Flash Memory Card Primer[124]

3.2.4.6 Smartcards

A smart card is "a small device, about the size of a credit card, which contains electronic memory and possibly an embedded integrated circuit (IC), which allows them to do a small amount of processing. Smart cards containing an IC (aka microprocessor) can cost 3-6 times more than non-IC cards, and are sometimes called Integrated Circuit Cards (ICC's)."[125] Much like flash memory, to access the data on a smart card, you insert the card into a device known as a smart card reader. In contrast to flash memory, smart card storage capacity ranges from 8K to 32K. No, this isn't much at all! And this is complicated by the fact that if you have a programmable card, the available storage often must hold both your program code and the data you wish to store.

Smart cards can be used for a variety of purposes, including storage of personal data like medical records, newer cell phone configuration information, management of passwords and digital certificate data, electronic "cash" (school dining credits, department store gift cards, etc), access control (the widely used "card key") and generating network authentication information (instead of using a fixed password every time you login).

---

[124] http://www.onepc.net.
[125] "Smart Card", http://www.webopedia.com

It is estimated that as of the end of 1999, more than 1.5 billion smart cards were in use worldwide. They are managed by smart card life cycle management software, which is generally obtained from a different source than the smart cards themselves, such as Bellid[126] or Litronic[127]. Because the capabilities of smart cards differ so substantially, you need to look at a variety of areas when evaluating smart cards, including security features, supported applications, storage capacity, and standards supported, data access controls, processor support, programming methods and algorithms supported.

Security features on smart cards, in addition to the embedded memory and possible IC, may include pictures, biometric data storage, bar codes, a magnetic stripe (as in credit cards) and even a very small antenna (for wireless communication that doesn't require the card surface actually contact the reader). Additionally, the chip, by virtue of being embedded in the card, is tamper-resistant. Typically each card has its own serial number[128].

An interesting aspect of smart cards is that they may contain more than one type of memory, for security purposes. For example, memory is either read only, or read/write. And within each of these types, are other types: memory which can be accessed without providing credentials ("public"), memory that can be accessed without a PIN but with other access controls ("scratch pad"), memory that can only be accessed with a PIN ("private"), "emergency" memory that can be written to or read from even after the card's preset expiration date.

Most smart cards support at least one standard encryption algorithm, since without encryption, anyone who can read the card, can retrieve the data off it. Encryption algorithms that may be available on smart cards include DES, 3DES, Diffie-Hellman, MD5, SHA-1, the NIST-approved DSA, Rinjdael, KEA, RSA, and ECDSA.

It has been said that "the nice thing about standards is that there are so many to choose from," and in the case of smart cards, that's definitely true. There are at least 25 standards a smart card can choose to comply with, including ISO/IEC, FIPS, ANSI, X.509 (certificate) and EMV standards, which define physical, electronic, algorithmic and formatting standards.

---

[126] http://www.bellid.com/
[127] http://www.sspsolutions.com/solutions/government
[128] "Smart Card Basics", http://www.gemplus.com/basics/index.html

Important from a security standpoint would be the FIPS-140, Level 3 standards level, an NIST security requirement for cryptographic modules.

A de-facto standard for programmable cards that is gaining in popularity is that of Java support, with other options being Visual Basic or MULTOS. Most new smart card applications are written in java, so using a Java-based smart card will allow you the greatest choice of applications, as well as comply with the GlobalPlatforms.org standards. Federal Government users choosing a Java-programmable smart card should make sure that the one they select is FIPS certified, such as Schlumberger's Java smartcard.

The software running on the card itself is not the only area of compatibility to consider when selecting a smart card. You should also make sure that the smart card reader and smart card lifecycle management software you choose works with the operating systems and OS versions on which you plan to use them.

If you are interested in implementing smart cards in your organization, check out the current offerings from ActivCard[129], Datakey[130], GemPlus[131], Oberthur[132] and Schlumberge[133]. A great book to start with is Smart Cards From Scratch[134].

[129] http://www.activcard.com
[130] http://www.datakey.com
[131] http://www.gemplus.com
[132] http://www.oberthur.com
[133] http://www.schlumberger.com
[134] Taylor, Laura, "Smart Cards from Scratch", http://www.intranetjournal.com/articles/200205/se_05_08_02a.html

3.3 Security Topologies

Security topologies deal with the organization of devices on a network, from a security perspective. They can specify the way devices within different zones of your network communicate with each other, and which types of zones are used within the network.

3.3.1 Security Zones

Security zones are areas of your network with specific security-related attributes and requirements. We will look at DMZs, intranets and extranets.

3.3.1.1 DMZ

The DMZ, or Demilitarized Zone, on a network is "that portion of a company's network which sits between the Internet and an internal network's line of defense, usually some combination of firewalls and bastion hosts"[135]. Firewalls were described earlier in this section. Bastion hosts are gateways between inside and outside networks, designed to defend the internal network from attacks aimed at it from outside. Much like in the non-computerized world, a network DMZ, sometimes called a "perimeter network" is a "neutral zone" that attempts to keep external users and the internal network apart. It is strongly advised that there be an external firewall (or at least a well-configured filtering router) between the DMZ and the Internet as well, for protection of the DMZ, although that is not required for that portion of the network to be considered a DMZ. Usually a DMZ is a separate subnet from your internal network, to minimize opportunities for compromises due to traffic sniffing.

The DMZ typically contains those devices which used for services which need Internet access (inbound, outbound or both), such as DNS, email (SMTP), FTP and Web (HTTP) servers. It lets an organization provide access over the Internet to a small portion of its network that offers services useful by those outside its internal network, without making all hosts on the organization's network similarly available to random Internet surfers. The hosts and other devices on the DMZ are more "exposed" to the Internet and are thus more vulnerable to attack. Therefore, care should be taken when planning which services and data will reside there, and on administering machines that reside there.

[135] "DMZ", http://www.webopedia.com

For example, your company's internal web server, not intended for use by those who don't have access to the internal network, should NOT be placed on the DMZ. Similarly, an internal mail server, like an Exchange Server, which stores individual mailboxes, should be located on the internal network, not in the DMZ. The idea is that if your DMZ is compromised, the information that is most likely to get corrupted or taken is that which is available on the DMZ – information on your internal network will remain private unless access to the DMZ allows attackers to break into the internal network as well.

A general guideline is to run only the most essential services on your DMZ hosts, because as with security in general, the more applications, ports, services and the like that are available, the greater the number of potentially-exploitable objects there are. Separating the services into ones which must accept connections from Internet hosts and those which need only accept connections from your DMZ host (for example, by having your DMZ-based web server access a database on an internal host) enables you to locate those services which only need to be accessed by DMZ hosts, on your internal network instead of in the DMZ.

While organizations usually implement only one DMZ, it is possible to separate your network into multiple DMZ's, each with a particular purpose (for example, Web server, Database server accessed by Web server, etc.) to further isolate traffic and increase the security of your network. More information on advanced DMZ design can be found in the paper, "Three Tiered DMZs" by Chris Mahn[136].

Because hosts in the DMZ tend to make good targets (not only are they relatively easily accessible, but also, they may be VERY visible, such as a Fortune 500 company's main external web server), it's a good idea to pay special attention to locking them down (to minimize security exposures), keeping up-to-date with updates and patches for the OS and application software used on those systems, and to formulate backup and recovery plans which will allow you to restore the system in the event of the almost-inevitable break-in to at least one of the hosts in your DMZ.

[136] http://www.sans.org

A DMZ is not without costs, including the cost of additional hardware and software, (possibly) a second firewall, a slight decrease in performance since traffic will not go directly between the Internet and internal hosts and the cost of maintaining and administering the DMZ (including additional security auditing).

Nevertheless, it is almost always well worth the expense, since a single break-in when a DMZ is not present can often result in data and staff time losses far in excess of the cost of setting up a DMZ.

3.3.1.2 Intranet

An intranet is a logical (not physical) network that is specific to a single organization. It may reside wholly within the organization's boundaries, and be contained within the company's internal network, or span across public networks such as the Internet. The traffic that goes across the intranet includes proprietary data that should not be exposed to those outside the organization. Because of the security implications of sending private data across public network links, when planning an intranet that uses these public links, you should also plan to implement VPN technology to authorize users and encrypt the traffic that crosses outside your internal network's boundaries.[137]

3.3.1.3 Extranet

An extranet is another type of logical network, which allows a business to connect with suppliers, vendors, customers, stockholders or others related to its business. An extranet is usually run over the Internet, often using a VPN for security. Alternatively, some companies limit their extranet functionality to data exchange via SSL-protected web pages that can only be accessed by authorized parties, and do not implement a VPN. The extranet's primary components are normally situated within the company's internal network, allowing limited access to some corporate resources to those outside the organization on a need-to-know basis, and allowing corporate staff to access limited resources on other organizations' networks as needed.

If you are employing a VPN for your intranet, you may wish to employ a second VPN for your extranet, to best separate this "restricted public access" traffic from your completely internal traffic.

---

[137] Houser, Tcat, and Helen O'Boyle, Inside Scoop to CompTIA Network+ Technology, BFQ Press, 2002.

3.3.2 VLANs

A VLAN, or Virtual LAN, is a logical subnet created through configuration of networking switches. It may be part of a larger LAN or WAN.

One benefit of VLAN is that you can get the benefits of a subnet without requiring hosts to be in physical proximity to each other, or connected to the network using the same physical technology (such as 100BaseT UTP vs. fiber). Switches and other network devices can be configured to pass data that would not normally be passed between subnets (such as broadcast packets) so that it is shared among multiple physical subnets, via a trunking protocol such as the emerging 802.1q standard, or the more secure 802.10 standard. Conversely, you can also use VLAN technology to break a single physical subnet into multiple logical subnets, reducing collisions and broadcast overhead.

An investigation into security vulnerabilities of VLANs reveals that it is not wise to assume that partitioning your network into VLANs provides the same level of protection as subnetting it or carefully designing a routed network the directs traffic appropriately. Researchers discovered through experimentation that it is possible to get the 802.1q trunking frames to hop into a switch's non-trunk ports and be delivered to their destination, and that it is possible to get 802.1q frames to hop from one VLAN to another if the frames are sent through a switch port attached to the native LAN of the trunk port. While an attacker requires some network knowledge (such as the MAC address of the target machine, and VLAN trunk configuration data) and access (to a switch port on the same VLAN that the trunk port is assigned to) to pull this off, it's often not impossible, depending on the configuration of the VLAN.[138]

[138] Taylor, David, "Are There Vulnerabilities in VLAN Implementations?", http://www.sans.orghttp://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci540868,00.html

3.3.3 NAT

NAT, or Network Address Translation, allows devices on private networks to communicate with outside networks by "translating" between the network address conventions used by each.

Typically, NAT is used by an organization connecting its internal network, using a private IP address range, to the Internet. Recall that there is a private IP address range for each Class of TCP/IP network.

Since these private IP address ranges cannot be used directly on the Internet, NAT was developed to act as a go-between, mapping internal host, port and connection information, to external connections.

NAT can work in a variety of ways. For example, a router may allow you to set up mappings to internal hosts, based on the port number of incoming traffic. So, if your one public IP address is 199.245.111.7 (it won't be, because that belongs to my vintage-1990's ISP startup class C– Helen), you can define port 80 connections to 199.245.111.7 as going to perhaps a private host 10.0.0.201, and port 110 connections to 199.245.111.7 as going to host 10.0.0.205, on your internal network. Similarly, NAT might work like a higher-level "TCP/IP switch" for outbound connections, accepting outbound connections from your internal hosts, recording the connection information (source and destination addresses, source and destination port numbers) in a table on the NAT box, and creating separate new connections used for actually contacting the external hosts. When the external hosts reply (to the NAT box), the NAT box checks its table to find out which internal host requested that connection, and forwards the inbound packet over the correct internal connection to the proper host.

Sometimes organizations use NAT by choice, to help limit the direct connectivity that is possible between internal network hosts and the outside and make it more difficult for outside attackers to "map" the target's internal network. Sometimes, but not always, NAT is combined with a proxy service which ensures that any outside connection inbound to your network terminates at the machine providing the NAT, and is proxied to the appropriate internal service with a second, separate connection – making sure that there is never a direct connection from an outside host to an internal server. This isolates your internal network from certain low-level attacks and exploits that might otherwise be possible.

Some organizations and individuals use NAT by necessity. ARIN (the American Registry for Internet Numbers) has long since stopped giving out permanent Internet network addresses, even for small Class C networks, because they were getting close to the point of running out of them. Also, the Internet had expanded to such a point that routing became increasingly difficult with random network addresses scattered all over the Internet – and could be simplified considerably if things were set up so that certain super blocks of net addresses, consisting of multiple Class (n) addresses, were all assigned to the same ISP.

Although some organizations and individuals who've been around for a while (like me, Helen) continue to use their old "portable" class C network addresses, nowadays, it is usually the case that you make do with whatever network numbers you get from your ISP – and as with most everything, the more you want, the more it costs.

For example, if your ISP provides you with 1 external address, but you have a small LAN with a half dozen hosts, you may use NAT to direct incoming traffic to your web server port to the machine handling your web server, incoming traffic to your SMTP port to your email server, etc. Additionally, NAT will multiplex your internal machines' outbound connections through the NAT box, making it appear to the Internet as if all traffic from your network is originating from that single IP address. On an individual basis, many users are familiar with NAT in terms of Windows' Internet Connection Sharing (ICS), which lets them share one Internet connection such as a dial-up modem, with all computers on a LAN. For slightly larger organizational networks, NAT is often implemented in DSL and other routers, as well as firewalls and proxy servers. The main characteristic is that the host implementing NAT will have two network addresses. They may both be Ethernet connections, as with some organizational routers, or may have one Ethernet connection and one dial-up, cable or DSL connection.

What other advantages are there to NAT, besides conserving Internet addresses and providing some additional security? Making do with what your ISP provides you also means that your external IP address(es) will change when and if you change ISP's, thus showing another benefit to NAT – generally the only device requiring reconfiguration when this happens, is the box (router, computer, etc.) in charge of the NAT mapping.

Any internal routers that understand the internal network address conventions will probably not require updating unless you make changes to your internal network addressing at the same time. Conversely, if your internal network addresses change, outside hosts don't need to be aware of this, since the only thing they need to know about your network is what your NAT box's external address is (and this is handled through DNS).

For more information about NAT, see the Network Address Translation FAQ at http://www.vicomsoft.com/knowledge/reference/nat.html

3.3.4 Tunneling

Tunneling is the process of encapsulating a protocol within packets carried by a lower-level network, to provide a virtual point-to-point connection. For example, many VPNs run over TCP/IP networks, embedding another TCP/IP network's traffic, sometimes using a completely different host addressing scheme such as NAT, within the outer network's packets.

In many cases, the complete packet traveling on the "virtual network" is placed into the data area of the physical network's packets, just as with any other application. The lower layers of the outer network are not even aware that another network's traffic is piggybacking over them.

Tunneling can provide an authenticated, encrypted, tamper-resistant channel of communication between two points. Tunneling software takes care of encapsulating the packets for transmission, and then receiving and decoding them at the other end of the tunnel and placing them onto the local network as if they had originated there. It can exist at various network layers, such as OSI layers 2 and 3, or higher application layers.

For example, layer 2 tunneling is often used to carry PPP traffic between VPN endpoints. When dialing in to a corporate VPN, there are two options – originating the tunnel at the dial-up user's host (voluntary mode) or intercepting the user's dial-in at the ISP and originating the tunnel to the corporate network from the ISP forward (compulsory mode). PPTP is an example of a layer 2 protocol that provides encrypted, authenticated tunneling. L2TP is a layer 2 protocol that provides authenticated tunnels, which can be encrypted using the layer 3 IPsec technology mentioned below.

Layer 3 tunneling provides virtual IP connections at the network layer. It is often implemented via the IPsec protocol extensions and IKE (Internet Key Exchange, an authenticated key exchange protocol). It supports a wide variety of encryption options, such as DES, 3DES, MD5, SHA1 and is often used in "security gateway" products such as IPsec-enabled routers, which provide dial-up or Internet users access to the internal network behind the gateway. Note that IPsec itself doesn't provide for authentication, which is why it is often paired with other technologies like L2TP, or used in full site-to-site links where the organization considers individual authentication to be overkill.

Higher level tunneling, when you wish to tunnel traffic related to some applications, but not all traffic on the network, is most often accomplished via Secure Shell (SSH), or Secure Sockets Layer (SSL). SSH seems to be the protocol of choice when tunneling login connections (providing a more secure remote connection than the Berkeley UNIX "r-"commands it was developed to replace). And SSL is, of course, the protocol used to implement a secure version of HTTP communication used between web browsers and servers.

Interestingly, the use of both of these originally special[purpose tunneling protocols has been expanded to other applications as well. For instance, many companies now use SSH as an inexpensive way to provide general-purpose security tunnels between remote clients and all sorts of applications, including web servers and POP3 or AMTP email connections. And SSL has evolved into the IETR-standard Transport Layer Security (TLS), which uses digital certificates for authentication and confidentiality. [139]

[139] Phifer, Lisa, "VPNs: Virtually Anything?", http://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci540868,00.html, 2001.

3.4 Intrusion Detection

Earlier in this section, we looked at the IDS as a component of a network. In an ideal world, every system would run flawless software (with no known or soon-to-be discovered bugs which can be exploited), administrators would never make careless mistakes and users… well… what can I say? Since this is the real world, not the ideal world, new security bugs are discovered daily, administrators do sometimes set permissions incorrectly and users load software from virus-infected diskettes. Thus, organizations need the capability to detect and respond appropriately to suspicious activity, and deploying IDS is one way to help automate this process.

Standards that apply to IDS, which you might wish to investigate, include CIDF (Common Intrusion Detection Framework, by DARPA) and IDWG (ID Working Group by IETF).

In this section we will delve into specific types and features of IDSs. There are two primary types, network based and host based. Some IDSs make use of only one or the other, and some make use of both.

3.4.1 Network Based

Network-based intrusion detection systems (NIDSs) monitor network traffic, looking for "interesting" events. When examining traffic, they can detect either patterns in individual packets that indicate suspicious traffic such as data streams from popular exploit tools (often referred to as "signatures"), or violations of algorithmic rules that indicate out-of-the-ordinary traffic (often referred to as "heuristics", such as more than 100 incoming FTP connections to a single host within 10 seconds). More advanced systems rank the level of threat of different events, and are able to correlate a variety of suspicious activities in order to determine if a more significant threat is present.

Some NIDSs also monitor SNMP, syslog logging communications and other network-event-reporting mechanisms for interesting network-related events. Some, but not all, NIDSs allow the administrator to create custom rules and algorithms to search for traffic of local interest. For example, one site has a custom rule that searches the network for SNMP traffic containing the default community strings of 'public' or 'private'.[140]

[140] Saoutine, Greg, et. al., "Barbarians at the Gate", http://mcpmag.com/Features/article.asp?EditorialsID=294

Because NIDSs generally run in real time, they are likely to detect potential attacks more quickly than host based IDSs. Another benefit of NIDSs over host based IDSs is that some attacks, such as a DoS or Teardrop attack, can only be detected by looking at the packet headers.

When selecting and deploying a NIDS, be aware of a few key points:

A. A NIDS can only monitor what it can see. If switches or routers prevent the NIDS from seeing traffic, it can't review that traffic for potential problems, so use span ports on switches or place the NIDS agent on a machine off a hub on the uplink port of the switch. Then TEST your configuration to make sure that the NIDS really is seeing the traffic you want it to see
B. If you use a VPN or other traffic encrypting technology, such as SSL for web transactions, on your network, the IDS may not be able to look inside those packets for potential threats, and will miss any attacks that occur through those channels
C. Most malicious attacks on a network occur on the internal network (given the increasing presence of third parties like vendor field engineers and consultants, on internal networks), and internal networks are full of well-meaning users who every so often open an email containing a virus, so consider using a NIDS to monitor the internal network as well as your DMZ
D. Some NIDS do not process fast enough to examine all traffic on gigabit networks, and may let traffic through without scanning it, if overwhelmed, so if you have (or plan to soon have) a gigabit network installed, make sure that your vendor certifies that the software and hardware you select for your NIDS can keep up with it
E. Make sure that the vendor regularly updates their rule database in response to new threats, and then make sure that someone in your organization is tasked with keeping the NIDS rule base up-to-date, by promptly installing vendor updates. If a signature-based NIDS doesn't know about a suspicious pattern, it can't look for it and alert you to its presence

F. NIDS which focus on detection of statistical behavioral anomalies (unusual patterns of network activity, as compared with normal network traffic) can often detect attacks unanticipated by simpler pattern-matching systems that looks for known threat signatures. The downside to the statistical approach is that benign but unusual traffic patterns often trigger the NIDS into action, resulting in a higher degree of "false positives" than with the pattern-matching approach, and thus require more administrative attention in exchange for a (possible) increase in detection ability
G. Tools such as "Stick", popular in black hat circles for flooding a NIDS with traffic and causing it to drop packets, can also be used for NIDS evaluation, to stress-test an NIDS before purchasing it
H. Not all NIDSs reassemble fragmented packets before comparing packets with signatures; those that do not, may miss detecting attacks if the attackers obfuscate them via excessive fragmentation

Typically, a NIDS consists of several components:

A. Agents deployed on hosts around the network to collect information and forward relevant information to the Director
B. Director which combines information from agents and analyzes it to find potential threats
C. Notifier which handles responding to threats identified by the Director[141]

For more information on NIDS, check the "Barbarians at the Gate" article referenced earlier in this subsection.

[141] Memon, Nasir, "CS 392 Network Security – Module 5 Intrusion Detection", http://isis.poly.edu/courses/cs393/lectures/module-5.pdf

3.4.1.1 Active Detection

NIDSs are increasingly focusing on active detection. This does not refer to how the NIDS detects potential problems, as you might expect. Instead it refers to how the NIDS responds to the situation once it is detected. (Sorry folks, this is the industry standard terminology. We didn't create it.) With active detection, the NIDS takes some action to mitigate the detected threat. Such options can include:

A. Reconfigure a firewall or router to route traffic around a problem such as a DOS attack;
B. Break suspicious network connections;
C. Send a message to a host-based agent to shut down a vulnerable service on a particular host.

3.4.1.2 Passive Detection

Most early NIDSs focused on passive detection, which involves alerting someone to the detected threat so that they can take action, if desired. IDSs that use only passive detection do not take action against the threat themselves. Passive methods include:

A. Logging the event;
B. Emailing or instant messaging an administrator;
C. Paging an on-call administrator;
D. Sending an SNMP trap to flag the event.

3.4.2 Host Based

In contrast to network based IDSs, host based IDSs focus on monitoring for unauthorized activities occurring on hosts themselves, including both incoming/outgoing network traffic involving the host running the IDS and other events on the host which do not involve the network. Host based IDSs have been around for a long time (since the 1980's, says ISS[142]). They range from simple accounting record auditing packages (which may or may not run in real time), to more advanced detection systems that not only check system, security and event logs, but also periodically verify system file checksums to ensure that critical configuration files and executables have not changed, and/or monitor certain ports for connection activity.

---

[142] "Network- vs. Host-based Intrusion Detection", Internet Security Systems, http://documents.ids.net/whitepapers/nvh_ids/

To secure your network using host based IDS technology, you would have to install a copy of the host based IDS on each system you want to monitor. Unlike NIDS, a single installation can only monitor a single machine – the one on which it has been installed.

While host-based IDSs typically do not address the variety of attacks that can be detected by NIDSs, there are some benefits to host-based technology, such as:

A. Ability to detect attacks which occur within a system, without traffic traveling across the network, such as someone sitting down at a serial-wired "dumb terminal" and replacing a key system file;
B. Fewer false positives (reporting something as a threat, which isn't really a threat), since host-based IDS technology usually looks at logs of what has already happened as opposed to what it looks like someone might be trying to do;
C. Not affected by switched environments or network-based encryption, because the system must decrypt the network traffic destined for it in order to respond to it. For example, the actual contents of an SSL transaction are known on the web server to which it is sent, so a host-based IDS on the web server may be able to detect anomalies in that transaction which could not be detected by a NIDS.

3.4.2.1 Active Detection

As with network based IDSs, host based IDS technology can feature either active or passive detection. Active methods used by host-based systems can include:

A. Terminating suspicious network connections;
B. Shutting down services which are being attacked;
C. Shutting off user accounts that are being used for suspicious activities.

3.4.2.2 Passive Detection

Passive methods used by host based systems mirror those available on NIDSs, including:

A. Logging the event;
B. Emailing or instant messaging an administrator;
C. Paging an on-call administrator;
D. Sending an SNMP trap to flag the even.t

3.4.3 Honey pots

"Mmmmmm, honey pot"…. The name is supposed to sound tempting… to potential attackers, that is. Honey pots are decoy systems or networks set up to look like interesting targets to attackers. The idea is that attackers will spend their time and resources on this (to you non-business-critical) system rather than interfering with the operation of more important systems on your network. Alternately, the honey pot can be purely a research tool, to lure the 3133t (that's "elite" in script kiddie speak) into attacking, so that those observing the honey pot can learn about their techniques.

Honey pots typically mimic a real system or network (ideally one particularly attractive to hackers, such as one that purports to contain interesting data or runs a service That is a known "easy target" like (sorry Microsoft!) IIS). Each system can be set up to run one or more services that any other server on your network would run. The difference is that a honey pot is not normally in use (at least, not in use doing anything other than pretending to be a great target). It is merely a target lying in wait, and any amount of interaction with it can be interpreted as an attempt at intrusion, reconnaissance, or other type of abuse. Honey pots are normally well-isolated from the rest of the network (due to obvious concerns about traffic sniffing), and feature good logging, often sending their log results across the network to a different machine so that compromise of the honey pot doesn't allow the attacker to "cover his tracks" by tampering with the logs.

Entire Internet projects such as Lance Spitzner's Honeynet Project[143] revolves around using honey pots to study black hats in their native habitat (out in what looks like "the wild").

[143] http://www.honeynet.org

While most honey pots, including those at the Honeynet Project, have traditionally been UNIX-based, there are also tools for setting up honey pots on Windows systems, which even include servers such as the free-for-personal-use BackOfficer Friendly from NFR[144]) which simulate popular Trojan servers like BackOrifice, but log instead of act on their requests.

"Luring? Deliberately going out of your way to create a target to attract ne'er-do-well's? What's our legal department going to say?" The only answer we can give is: we don't know. Some have claimed that honey pots are a form of entrapment. Others have pointed out that entrapment can only be committed by law enforcement, so random net administrators and security researchers aren't affected by that regulation. Additionally, if your honey pot is compromised, and the attacker does make off with information you wish he hadn't what then? **[Fragment…..not a complete sentence]** Does the fact that you set the honey pot up specifically for use by such individuals imply that they might have been (in some legally-defensible way) authorized users of that system and thus broke no privacy rules? It's tough to say, since we're still in the early stages of legal precedents in this area. There are questions here that before you set up a honey pot of your own, you would be wise to confer with your legal counsel to determine potential legal ramifications.[145]

For more information on honey pots, check the paper by Lance Spitzner mentioned in the footnotes, as well as the http://www.honeynet.org site.

3.4.4 Incident Response

An intrusion has been detected… Now what? Unless you are restricting yourself to observation only, as an academic exercise, the next step is to respond to it.

As seen above, sometimes part of that response is accomplished FOR you, by the IDS doing what it can to stop the attack, determine the extent of the damage, and safeguard the system or your network from further attacks by taking actions like shutting down services.

---

[144] http://www.nfr.net/products/bof
[145] Spitzner, Lance, "Honeypots: Definitions and Value of Honeypots", http://www.enteract.com/~lspitz/honeypot.html

At other times, it's up to you to get the email, page, or alert on your computer screen, and race into action. Do you immediately shut the attacker off? Or do you let them continue for a while, and try to determine the source of the attack so that you might have a better lead on who could be prosecuted later? (Take note: few prosecutions of this sort are successful.) Do you yank your site's Internet connection? All of these are potentially valid actions that the authors have seen network administrators take more than once.

The NSWC Dahlgren Computer Security Incident Handling Guidelines[146] describe incident response as a six-step process:

A. Preparation (setting up systems to detect threats and policies for dealing with them, including identifying roles staff will play in incident response, and creating emergency contact lists);
B. Identification (identifying what the threat is, and/or the effects it is having on your systems/networks, including keeping records of the time/systems involved/what was observed, and making a full system backup as soon after the intrusion was observed, as possible, to preserve as much information about the attack as you can);
C. Containment (limiting the effects of an incident by confining the problem to as few systems as possible, freezing the scene so that nothing further happens to the compromised system(s) by disconnecting its network connections and possibly console keyboard)
D. Eradication (getting rid of whatever the attacker might have compromised by deleting files or doing a complete system reinstall – we cannot stress enough that you should err on the side of deleting MORE rather than less in order to restore a system to production, since the intruder may have left Trojan binaries around the system, to be activated once the system is reconnected to the Internet);
E. Recovery (getting back into business, by putting the system back into normal operations, reconnecting it to the network, restoring from backups if necessary, etc.);

[146] "NSWC Dahlgren Computer Security Incident Handling Guidelines", http://www.nswc.navy.mil/ISSEC/Docs/Ref/GeneralInfo/incident.handle.html, February, 2002.

F. Follow up (if possible tightening security so that the intrusion cannot happen again, determining the "cost" of the intrusion based on staff time/lost data/lost user work time (don't skip this! It may help justify security expenditures in the future), considering which, if any, additional tools might have helped handle the incident better than it may have been handled, reflecting on "lessons learned" from both the intrusion and the organization's response to it and tweaking policies as required).

SANS offers an incident response publication dealing with these 6 major phases, in detail[147].

Additionally, you may wish to take advantage of some third-party channels. For example, if you seem to have a new vulnerability on your hands, you might want to contact your application, OS or security product (firewall, IDS, etc.) vendors to see what information you might be able to share with them, to help them protect against (or respond to) this vulnerability. You might want to contact CERT[148] or the BUGTRAQ mailing list, to report the flaw so that other white hats find out about it and can take appropriate action (if it's been aimed at you, at least one black hat's already got the exploit).

[147] http://www.sans.org/newlook/publications/incident_handling.htm
[148] http://www.cert.org

3.5 Security Baselines

Security Baselines are standards that specify a minimum (that is, "baseline") set of security controls that are suitable for most organizations under normal circumstances. They typically address both technical issues (such as software configuration) and operational issues (such as keeping applications up to date with vendor patches). The idea of security baselines is that for any particular platform (hardware, OS, network, application), there is a minimum set of security recommendations which, if followed, will significantly decrease its vulnerability to security threats, and that it shouldn't take an expensive consultant doing an extensive risk analysis of your environment to determine a reasonable set of security controls for you to implement. In this way, even a small mom-and-pop business without access to a major IT consulting firm can have some assurance that they are taking at least some worthwhile steps to computer security.

There are multiple schools of thought on the use of security baselines. Some think adopting a common set of security baselines across the industry is the way to go – a kind of set it and forget it approach that ignores the risk analysis step. Others think that baselines are just a starting point for the bare minimum acceptable level of security and those organizations that can, should expand upon them to further increase the security of their system as time, knowledge and budget permits and their particular risk situation requires.

When establishing Security Baselines, you may consider:

A. Any existing security baseline documents for the hardware and software you use;
B. Any "best practices" guides that exist for hardening the hardware/software you use, which may exceed the recommendations in any proposed baselines for that hardware/software;
C. Specific issues you may have run into the past which deserve extra attention (suppose your web server has historically been a favorite target of hackers);
D. What other administrators are saying and doing (do you really want to run the easiest FTP server for "warez" folks to take over, on the whole Internet? If not, take the same steps other administrators customarily take to secure their servers);

E. Unique characteristics of your environment (in terms of security risks faced, how much collaboration takes place, management's views on the security requirements vs. ease of use tradeoff, etc.).

Some security auditing products take the idea of baselines one step further by allowing you to specify rules for your desired system configurations (and other security controls) within the auditing product, so that the tools can automatically scan for deviations from those baselines and report them to you.

3.5.1 OS/NOS Hardening

OS hardening deals with the actions that can be taken to secure an OS.

In the UNIX world there are many tools that help checks for compliance with organizationally set baselines. In the Windows world, there are both written guidelines such as the SANS "Gold Standard" and automated tools which attempt to automate some baseline security measures. The "Gold Standard" is a recent development by the Center for Internet Security, DISA, NSA, NIST, SANS and GSA. The US National Security Agency, after review of successful system compromises of Windows 2000 during the past eighteen months, found that more than 85% of them would have been blocked had the owners been using the Gold Standard. Heard enough? Go get it[149].

Types of things that are usually taken into account when hardening an OS include:

A. User accounts (removal of unused accounts, enforcement of password security, enforcement of account lockout on unsuccessful passwords, etc.);
B. Installed options (don't install options that you don't use … that just means there are more potential vulnerabilities for someone to take advantage of);
C. Available services (if you don't need a service, like Windows File Sharing, running on a computer, by all means, leave it disabled … as with installed options, enabling it just increases your potential vulnerabilities);
D. OS configuration (settings for logging, etc.).

[149] http://csrc.nist.gov/itsec/guidance_W2Kpro.html#NSA_W2K

3.5.1.1 File System

File system issues are important to consider when hardening an OS. These relate to both the type of file system chosen, and the access controls on information stored in them.

In most modern operating systems, an administrator can choose to format a disk in any number of standardized formats, called file systems. For example, in the Windows world, there is NTFS, FAT-32, FAT-16, etc. In the UNIX world, there are MS-DOS compatible file systems, JFS, extfs, ReiserFS, etc. The best way to make sense of these is to study your documentation, as complete coverage of the attributes of these file systems is beyond the scope of this document.

Several security-related aspects of file systems are important to consider when choosing a file system:

A. What kind of access controls does the file system provide? (Some, like the FAT-32, provide none);
B. What kind of encryption/data privacy features does the file system provide? (Again, some may provide none, requiring you to use application-level encryption);
C. How resistant is the file system to loss of data as a result of a system crash? (Some, like compressed file systems sometimes used in days of old to conserve disk space, were notoriously bad; others, like the journaling file system for UNIX, are generally good).

Additionally, one should visit the topic of proper file access control configuration. For example:

A. Don't allow users write privilege into system directories, or shared data directories they don't need to update;
B. Provide each user with their own home directory, whose file access control settings fit your organization (for some organizations, this might mean the directory is unreadable by all users except the directory's owner; for others, it might mean everyone can read it but only the owner can write to it);
C. Make sure that sensitive shared data directories are unreadable by everyone except those authorized to use them.

3.5.1.2 Updates

As if we haven't mentioned this enough already, you need to watch for them and install them. Vendors produce updates for all sorts of reasons – not just for fixing security issues. Because installing updates

3.5.2 Network Hardening

Network hardening deals with the steps taken to secure a network and the devices on it.

3.5.2.1 Updates (Firmware)

As always, stay on top of the latest versions of firmware available for your networking products. These often address security-related vulnerabilities.

3.5.2.2 Configuration

Configuration deals with setting options on the devices. One key issue here would be device passwords used to access administrative features of switches, routers, etc. Devices typically ship with a default password used for initial device configuration. Please make sure you change it ASAP. Entire web pages are dedicated to lists of default passwords for common network devices, so that intruders can walk right in your front door after turning the key.

3.5.2.2.1 Enabling and Disabling Services and Protocols

As stated above when discussing OS hardening, TURN OFF anything you're not using. Not using IMAP? Don't allow traffic on port 143 through your firewalls and routers. Someone installed a UNIX IRC server "just for test purposes" to see if it enhanced internal communication, and left it running, forgotten, after the test was complete? Get rid of it. Got a database sitting in your internal network that your DMZ-based web server uses to look up customer information? For the love of security, make sure there's a network filter in place that doesn't allow any random IP address on the Internet to connect directly to the database server! (It happens more often than you might think, and the cost can be tremendous – compromised customer name and address information, credit cards, inventory data, etc.)

3.5.2.2.2 Access control lists

Access control lists, sometimes called filters, are used to determine which traffic is permitted to pass through a network interface, in which direction. Routers typically manage access control via a text file of access control rules; OSs and home networking devices including this functionality generally wrap access control lists in a GUI for ease of use. Typical communication attributes taken into account in access control lists include a packet's:

A. Source IP address;
B. Destination IP address;
C. Source port number;
D. Destination port number;
E. IP protocol number (this is not the same thing as the application port number);
F. Direction of travel (incoming to or outgoing from the interface).

You should set up access control lists to enforce your security policies, such as those which specify which Internet services are, and are not, made available from your network to the Internet. Two "no brainer" access control list rules to implement are:

A. Do not allow into your network, any traffic from the outside whose Source IP address is set to an address inside your network (it's a red flag that the incoming traffic was spoofed – to keep the attacker out, don't let the traffic in);
B. Do not allow out of your network, any traffic from the inside whose Source IP address is set to an address outside your network (another red flag that the traffic is spoofed – although this time, you've got more problems, because the spoofer is somewhere on your network, at least you're not aiding and abetting him in attacking someone on another network).

3.5.3 Application Hardening

Application hardening can be a major issue simply because there are so many applications in use on the average network. It's simply difficult to keep track of what's installed where, let alone what the latest security baseline recommendations are for each product. Nevertheless, it's a task that needs to be performed. There's really MUCH more that can be said in this section, than we've said, but time pressures required that we be brief. **[Do we want to add here since this is the real deal???? No substantial changes to what was released PRE-BETA. Adding links for further research for areas that were just missed (such a Worms).]**

3.5.3.1 Updates

Keep up to date with the latest fixes and patches for the applications used on your network.

3.5.3.2 Web Servers

Sorry. Not done in time. Frozen. Are we filling in the holes here Nope.

3.5.3.3 Email Servers

Sorry. Not done in time. Frozen. Same as above Same answer as above ;-)

3.5.3.4 FTP Servers

There are two primary aspects of an FTP server to consider when hardening it: user authentication and file access permissions.

FTP servers accept connections in either authenticated mode or unauthenticated mode. Authenticated mode sends, in the classic Internet style, the user and password across the network, and assumes no one's running a sniffer on your network. Because this is a bad thing, newer FTP servers feature a Secure/FTP protocol that handles authentication in a more secure manner.

Anonymous FTP is a whole 'nother barrel of laughs. Presumably you've heard of the concept of "warez" – pirated software. Well, these warez folks need lots of disk space and bandwidth to store and distribute their software… and they often get it from random anonymous FTP sites around the net which have at least one directory writable by the anonymous FTP user. If you have to allow anonymous FTP access for one reason or another, OK – but make sure that you don't offer anonymous users a writable directory.

Note that if a warez geek does find and use your server, and then you shut off his/her access, you've just dramatically upped your network's chances of being attacked by a group of software "traders" annoyed at losing "their" site.

File access permissions refer to which FTP users have access (and what type of access – read, write, delete, etc.) to which resources on the server. Some FTP servers rely strictly on OS security to set up these permissions. If the OS would allow that user to have access to that file/directory normally, the FTP server lets them have it. Other FTP servers start with that level of security and then add onto it an additional file access control file that modifies those permissions when files are accessed in the context of an FTP server. Do make use of file access permissions.

3.5.3.5 DNS Servers

Sorry. Not done in time. Frozen. Are we filling in here also….or leaving as is Door #2!

3.5.3.6 NNTP Servers

NNTP Servers handle the distribution of Usenet News.

Some require users to authenticate themselves before access is provided, and some don't. If possible, run a news server that requires authentication and takes steps to ensure that articles are not submitted with forged identity information.

Permissions can be set up to allow users to read news, post news, and xfer (transfer) news (the permission to copy news articles in bulk – to enable other news servers to transfer news to and from the server).

NNTP software tends to be fairly complex, and security holes are discovered in various implementations from time to time, with results including an attacker obtaining system administrator access to the news server, creating a denial of service situation, reconfiguration of the news server, etc.

3.5.3.7 File/Print Servers

The most important point here is to watch configuration details. Make sure you haven't made any directories or devices available to the world that you didn't want to be available to the world.

3.5.3.8 DHCP Servers

Since DHCP servers don't require authentication of either client or server, they are vulnerable to exploits by attackers. For example, any client can request a network address – if enough spurious requests are aimed at a DHCP server, its pool of available addresses can be exhausted, depriving legitimate users of access to the network. Therefore, it is recommended that your DHCP server be configured to hand out addresses only to those hosts that are "known" to you (for example, those hosts whose MAC addresses appear in a file on your DHCP server).

On the other side, anyone can run their own DHCP server on a network, and if that server is faster at responding to DHCP queries than the "real", authorized server, clients will accept the data from the rogue DHCP server. Among the problems this can lead to is the rogue DHCP server providing incorrect DNS nameserver addresses, which might allow the attacker to redirect traffic originating at that client, and destined for legitimate sites, to other sites, by "faking" bogus DNS information for the legitimate site. Therefore, it is recommended that DNS information be configured statically on each client rather than provided by the DHCP server.

(There is a new version of DHCP that will feature authentication, but is it out for your platform yet? I didn't think so.)

3.5.3.9 Data Repositories

Data Repositories are locations that hold data – about your network or about its business. For obvious reasons, you should care about protecting repositories from unauthorized reading and modification.

3.5.3.9.1 Directory Services

Some directory services don't make use of authentication, instead allowing anyone to query the directory for information about network resources and users. Still other directory services support multiple forms of authentication – allowing the administrator to choose the most appropriate mechanism (hint: challenge/response or PKI based authentication schemes are more secure than those which transmit a password in encrypted or clear text forms).

Information provided by directory services can include sensitive information about the enterprise and its network configuration – types of data that you wouldn't want an attacker with a sniffer to have. Therefore, many directory services can make use of encryption when sending data back and forth between directory service client and server. If your directory service supports an encrypted communication path, use it. If you're using vanilla LDAP, consider moving to LDAP over TLS, which provides such encryption.

3.5.3.9.2 Databases

Databases are quite useful tools from a hacker's perspective. Not only can they contain valuable data, but also they're often a handy portal into command line access to the OS – sometimes with system administrator privileges – due to programming errors in the database software itself, or in applications written by others, attached to the database server. Some databases feature the concept of a "stored procedure" – small programs stored within the database server to do a series of things (like, often, run certain command line commands) when they are invoked by database users. The stored procedures often take user-provided data as parameters, to determine exactly what the stored procedure will do. As with CGI script exploits and buffer overflows, it's sometimes possible to creatively manipulate this data to do something other than what you might expect.

Along the same lines, many web applications are written to build database commands (in the SQL language used by most databases) from certain keywords like "SELECT" and input that is provided by a user via a web form. Often the provided input is simply copied into the database command as it is being built, without checking to see that the web form data would actually be valid, and then submitted to the database to be run. As with stored procedures, it is possible to creatively construct web form data so that it actually embeds additional attacker-specified database commands into the original command. The database sees the additional database commands and not knowing they're not legitimate requests, executes them – and gives the hacker a map of the database, deletes your customer records, changes item prices, etc. This particular attack is known as SQL injection.

Of course, the most straightforward issue with databases is simply configuration for appropriate levels of data privacy and integrity. Your database administrator should be responsible for maintaining the necessary security on the sets of data stored in the database. Many databases can be configured to accept individual user logins as well as general connections without authentication, and then match the user login with access rules for the data in the database, to determine what kind of access (delete records, add new records, change records, read only) the user has to each type of data in the database.

One final note: Remember default passwords? Some databases have them, too. Make sure you change the password of any account installed with your database installation. EVERYONE knows the default password for older versions of SQL Server; surprisingly few people change it. Combine this with a network configuration that allows the database server to be accessed from the Internet and it's not a pretty sight.

Chapter 0100

Basics of Cryptography (15%)

4.0 Basics of Cryptography

Sorry. Frozen. **Are we filling in the hole**

4.1 Algorithms

Keep in mind what an algorithm really is: a sequence of operators and numbers that define an outcome with unknown, but anticipated anomalies. In other words, the sequence wants to produce a desired end result, but doesn't always know which parameters are going to be entered into the equation.

4.1.1 Hashing

Hashing is the process of creating a long alphanumeric string that is a sum of a file. It is considered computationally infeasible to modify the content of a message or program while retaining the same sum. It is impossible to ascertain the contents of a program or message from the sum alone. The most popular type of integrity check in this genre is MD5. The footnote takes you to a freeware version know as MD5Sums[150]

4.1.2 Symmetric

Symmetric algorithms come in either the block or stream cipher (coding methods). In a block cipher data is broken into blocks and encrypted (and later decrypted) with the same key. Stream ciphers work the same way but work on a bit-by-bit basis. In both cases, the key is the same and must be protected. (Obviously you cannot send the secret key along with the data.) This is frequently known as Secret Key encryption.

4.1.3 Asymmetric

This uses one key to encrypt data and a different key to decrypt the data. The most popular known example of the Public/Private key system is PGP (Pretty Good Privacy)

[150]http://www.pc-tools.net/win32/freeware/console/

4.2 Concepts of using cryptography

"Crypto" is the catchall term for cryptography, crypto-analysis and cryptology. If you're looking for a break, and you are into creative avoidance several folks have suggested the book Cryptonomicon[151] as a nice fiction book to study this art. The term is based in the Greek language that translates to secret writing.

4.2.1 Confidentiality

The theory of cryptography is to hide the message, creating confidentiality. This works provided the key is not detected or broken. Keys come in two flavors, secret key Symmetrical encryption) or asymmetrical encryption (public/private key).

4.2.2 Integrity

Both public private keys and symmetrical (secret) keys are acceptable for confidentiality. Other methods must be employed to ensure data integrity. The most commonly used is the message digest (MD) function. Message digests are one-way hashes. The hash output varies with the standard selected. The most common is MD5 which produces a 128-bit that is mathematically unfeasible to alter the message without creating a different hash output.

4.2.2.1 Digital Signatures

Sorry. Frozen. **Are we filling in the hole Nope. Readers can dig on there own. The wise reader would realize 1) The original release said we ran out of time, and felt this area was weak. They were warned! And 2) not making substantial changes now that one of us has seen the test. Could be construed as an illegal revealing of the data on the beta test.**

4.2.3 Authentication

Establishes and confirms that communicating parties are who they say there are. This can be accomplished with secret or public keys.

[151] http://www.cryptonomicon.com/beginning.html

4.2.4 Non-Repudiation

Cryptography can be used to disallow the possibility that a message was forged. In other words, you cannot deny you sent a particular message. In the paper world this is done via a Notary Public who witnesses the signing and puts their seal that they witnessed the signing. In the digital world the same idea of a trusted third-party is used. The third party is called a Certificate Authority.

Non-Repudiation cannot be achieved with a symmetrical or Secret Key. It must be accomplished with Message Digest & Digital Signatures.

4.2.4.1 Digital Signatures

Digital Signatures[152] require that the sender determine what is to be signed (so the hash can be created – refer to 4.2.2.Integrity) then it must be signed by the owner's private key, which is opened by the distributed public key. A receiver may wish the verification of a public key. This is done thorough the Public Key Infrastructure.

4.2.5 Access Control

When using a Secret Key (symmetrical key) all parties need to insure the key is secured. This can be impractical when working with a large group. In a public/private key arrangement, the private key must remain private. Should a compromise of the private key occur, it must be destroyed. (Refer to 4.3.2)

4.3 PKI

Public Key Infrastructure is generally referred to as Certificate Authorities who feature:

A. Creation and distribution of Public/Private keys.
B. Publishes public keys in open directories.
C. Secures private keys.
D. Provides revocation in the event a private key is compromised.
E. Acts as a digital notary for the holders of public/private keys.
F. Has a Registration Authority that is typically publicly available (subject to DOS attacks).

Note that PKI is based on the X.509 standard.

[152] http://www.abanet.org/scitech/ec/isc/dsg-tutorial.html

4.3.1.1 Certificates

Certificates bind a public key to a specific person, business, document, software etc. Certificates are given an expiration data when created.

4.3.1.2 Certificate Policies

The certificate policy is a set of rules indicating the applicability of a certificate to a class of application(s) within common security requirements

4.3.1.3 Certificate Practice Statements

Certificate Practice Statements[153] (CPS) is state of the practices that the certificate authority employs in managing the certificates it issues. It should describe how the certificate policies are interpreted in the context of operating procedures and system architecture within the organization.

4.3.2 Revocation

Several methods of revocation are currently in place. The older method employs a list of keys that cannot be trusted being held in a CRL (Certificate Revocation List). A newer method is the OCSP[154] (Online Certificate Status Protocol)

4.3.3 Trust Models

Different trust models exist to create a "chain of trust[155]".

A Web-of-Trust is the simplest model. Each user creates and signs certificates for the people they know. This is the basis for PGP and has no central authority.

In the Single CA Model each person (document/software/business/computer) is given a public key out-of-band (not sent the same way a message is). A single point is used to check against revocation.

The Hierarchical Model is Multiple CA s with a **Root CA** at the top, using lower level CAs whose public keys are signed by the Root CA. Higher overall assurance than other models however it may not work in a peer-to-peer role. Performs well in a large hierarchical environment IE. Military.

---

[153] http://www.entrust.com/resources/pdf/cps.pdf
[154] http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci784421,00.html
[155] http://www.e-government.govt.nz/docs/see-pki-paper-4/chapter3.html

The Browser Trust-List Model is sometimes called the CA list. Each user has a list of the public keys for all the CAs the user trusts. The good news is different CA can be used for each application. The less than stellar news is there is no way to discern the strength of the PKI class. Verisign has 4 different classes.

The Policy Trust List Model restricts access based on the policy under which the certificate is issued. Recommended in X.509 V3 Certificate RFC but not adopted to date.

The Policy Trust List Model with OU is similar to Policy Trust but using Organizational Units for policy.

In the Cross-Certificate Model, each CA creates a certificate for the CA that has been confirmed to have an equivalent strength. Only one root public Key, however, the key is the local CA not the Root CA.

The Bridge CA is a trust bride that is built with cross-certificate pairs is an emerging concept.

4.4 Standards and Protocols

The IEEE has a working group for Public-Key Cryptography[156]

Sorry. Frozen. **Are we filling in the hole**

[156]http://grouper.ieee.org/groups/1363/

4.5 Key Management/Certificate Lifecycle

4.5.1 Centralized vs. Decentralized

A model such as PGP is decentralized, therefore, would not scale well at the enterprise level. Centralized is done at by a firm, such as Verisign, or within an organization as seen in the Liberty Alliance.

4.5.2 Storage

As private keys are in the digital sense, the “keys to the city”, improper protection of private keys would prove a major disruption of business and costly in terms of revenue. This does not even begin to address the security aspects between compromising of private keys and the time when the compromise was discovered.

4.5.2.1 Hardware vs. Software

Hard to figure out where the SMEs are going here. Do they mean put private keys on something like a USB memory stick and put it in a safe vs. encryption on a hard/soft drive with the stuff locked up? ***[What do we want to do here???? Let the discussion boards fight it out.]***

4.5.2.2 Private Key Protection

Another WTF. I see the transition to escrow, but I'm concerned with so many sub-domain points being called out without explanation.

4.5.3 Escrow

Key Escrow[157] is the procedure of keeping a copy of a users private Key in a centralized location which is only accessible to security administrators. This can be done either after the key has been generated or by pre-compiling public/private keys before assignment.

(Editor note:”I'm still not sure about that definition of escrow. A quick google showed 50% of sites say it's a 3rd, separate key held by the admin, the other 50% say it's a copy of the private key. Hm). ***[Do we want to leave this or further the it with facts? Allow the discussion boards to data-mine the confusion.]***

[157] http://www.networkmagazine.com/article/NMG20001004S0015/2

4.5.4 Expiration

At the time a digital certificate is created it comes with an expiration data. A new certificate must be created by the time the previous certificate expires. This is different from revocation

4.5.5 Revocation

Verisign states the following reasons for revocation:

"There has been a loss, theft, modification, unauthorized disclosure, or other compromise of the private key of the certificate's subject.

The certificate's subject (whether an IA<*Information Assurance*> or a subscriber) has breached a material obligation under this CPS (*Certification Practice Statement*), or

The performance of a person's obligations under this CPS is delayed or prevented by an act God; natural disaster; computer or communications failure; change in statute, regulation, or other law; official government action, including but not limited to acts by agencies responsible for export control administration; or other cause beyond the person's reasonable control, and as a result another person's information is materially threatened or compromised."

4.5.5.1 Status Checking[158]

The concept of status checking is to use a relaying party to confirm in real time, not unlike a credit card authorization when presented for payment.

Note: *The footnote is a link to a white paper published by Microsoft. As Microsoft is a Cornerstone member of Security+, it does not take a great leap of faith to pay extra attention to official Microsoft documents when preparing for Security+. Since at the time of this writing (20 August 2002) this work is partly an outline for our reference work with some of our work in progress from said reference work, the reader is advised to "dig deeper". While we could have done this at this juncture, it seemed more important to addresses all the beta objectives as published before day 1 of the Security+ beta exam.*

[158] http://www.farcaster.com/papers/fc99/fc99.htm

4.5.6 Suspension

Suspension of a certificate places a hold without causing a revocation. Suspension comes with a reason code.

4.5.6.1 Status Checking

Status checking is under suspension. Could this be referring to the reason code? ???????

4.5.7 Recovery

Signature Recovery is a process that lets you import a private key when upgrading a device (such as a PC), a hard disk has died, or a signature (pass) phrase is forgotten.

4.5.7.1 M of N Control

*Yet another WTF. What does computer math concepts for Artificial Intelligence play in a certification testing for 2 years of practical experience? **Anybody** who can explain this send email with high importance to Tcat@tcat.net.* **[Will we leave this like it is or actually insert facts???6 Leave it as a great topic for the discussion boards. I'm still baffled over this.]**

4.5.8 Renewal

According to Verisign[159]:

1. Renew existing certificate (using your current public key from your public/private key pair from the existing certificate).
2. Renewing the existing certificate by creating and using a new public key that you provide from a newly created public/private key pair. This is essentially a new certificate. This is the suggested option from VeriSign.

4.5.9 Destruction

Unclear where this is going. ***Need more here…….if possible also Great fodder for the discussion boards.***

Sorry. Frozen. **Are we filling in the hole**

[159] http://support.aventail.com/akb/article00626.html

4.5.10 Key Usage

Unclear. Are the SMEs looking for legal acceptance of Digital signatures? If so[160] ***[Did we find out where they were going with this Nope.]***

Editor Note: Maybe it's to do with certificate templates/roles? Eg, MS example, Smartcard User -v- EMail Signing only etc.)

4.5.10.1 Multiple Key Pairs (Single, Dual)

Conventional hierarchical PKI uses a single key. According to VeriSign, "Key Pairs are used for one or more of three basic purposes: encryption, authentication and non-repudiation." A single key used for multiple purposes violates non-repudiation[161].

[160] http://www.ilpf.org/groups/analysis_IEDSII.htm
[161] http://www.verisign.com/resources/wp/enterprise/management/management.pdf

Chapter 0101

Operational/Organizational Security (15%)

5.0 Operational/Organizational Security

Sorry. Frozen. **Are we filling in the hole**

5.1 Physical Security

In one documented case, precautions were taken, and the server was placed in a locked, windowless room. The security aspects seemed satisfied. The hapless firm discovered otherwise when the intruder simply pushed the ceiling tiles aside outside the server room, and went over the wall, which did not go all the way to the ceiling. Once inside, the intruder unplugged the server, opened the door from the inside of the server room, and walked out with the server containing the data.

Consider:

A. Physical barriers from 6 sides
B. Alarm System
C. Camera fee to long term videotape (time-laspe)

5.1.1 Access Control

Sign in/out logs to provide an audit trail.

Video surveillance for investigation and evidence.

5.1.1.1 Physical Barriers

Good defense employs *concentric rings of security*. This term involves two thoughts in one term. The first is the process of effective physical security creating layered perimeters in circles that get tighter with each circle. In addition to tighter rings, as the core is approached, certain (mind & emotion) principles are applied. Once again, three points are to be considered when applying physical security to a sensitive area. For security zones they are:

1. The outermost ring or perimeter should be the first line of defense and should contain the public activities. Continuing inward, more security measures should be incorporated as one approaches the private or high-risk areas.
2. Prominent use of signage (symbolic language), architectural elements and natural and man-made barriers should be utilized to signify the transition from public to semi-private, and finally, private space.
3. Clear border definition should be provided for the controlled space.

Further, there are territorial behavior strategies that contribute to physical security. There are guidelines for this aspect as well. These include:

A. Create elements to reinforce the feeling of proprietorship within an area. Use signage, reflecting the purpose of the area.
B. The physical area should support legitimate users. Architectural amenities should not attract undesirable behavior or provide concealment points.
C. Provide natural barriers for controlling activities that conflict with the purpose of the area.

5.1.1.2 Biometrics

Biometrics adds an additional layer of security. Passwords coupled with biometrics involves authentication with something you are. Biometrics should not be used as the single means for authentication. Tsutomu Matsumoto of Yokohama National University has demonstrated two different techniques for circumventing biometrics with an 80 percent accuracy rate[162].

[162]http://www.itu.int/itudoc/itu-t/workshop/security/present/s5p4.pdf

While other biometric options are possible, such as facial or iris scanning, fingerprint scanning is currently the most popular biometric because it is the lowest cost option with the highest accuracy rate.

5.1.2 Social Engineering

Security's biggest challenge is people. The art of dealing with people is called social engineering. Refer back to the first chapter (titled README) for further information.

5.1.3 Environment

Policies regarding protection of corporate assets and employee safety need to be established by senior management. Committees involving stakeholders need to create recommendations so intelligent choices can be made balancing issues of implementation costs, and security.

5.1.3.1 Wireless Cells

While not as 'cool' as the relative high-speed nature of 802.11x, other wireless technologies have been around for upwards of 10 years. Generically, we refer to these as cellular phones. While voice is the major use of cellular, data cable adapters have been available for cellular phones for a long time. It is a very rural area that does not have cellular phone coverage. Before jumping to the table of the speeds available, consider how much damage could be done if email from select accounts was secretly transmitted. The email text does not consume a great deal of bandwidth.

Now consider the following table.

1G	Mobitex, Moitent	9.6Kbps
2G	CDMA, CDPD, TDMA-IDEN	19.6 Kbps
2.5G	GPRS	128+/- Kpbs
3G	CDMA2001x	384Kpbs

The table illustrates the total bandwidth, and like Ethernet, about ½ of this speed is the real throughput. Now even with 1st generation wireless, which means over 400 characters per second of email, could be leaked without anyone knowing. Then think about the 2.5 G PCS phone plans that offer unlimited data usage for about $100 a month.

Given an AC adaptor to a PCS phone on one of these plans and 16,000 characters per second of email could be sent to virtually any location in the world.

Nextel uses IDEN (Integrated Digital Enhanced Network) who is based in TDMA, but offers private 2-way radio. Nextel is specifically mentioned because we know that setting an IP address on a Nextel phone is very easy.

5.1.3.2 Location

Minimizing transmission power reduces the changes your data will leak out of the intended area. Antenna placement will also have an effect[163]. Attempt to place antennas as far from exterior walls as possible. Typically the interface between the wired network and the transceiver is placed in a corner in an effort to hide the electronics. That places the network signal outside and easy to intercept. In effect, you have put an Ethernet jack for your network in the parking lot. Beyond controlling power output and antenna placement and configuration, consider shielding, the next topic.

5.1.3.3 Shielding

The building in which the wireless network is operating can used as a shield for itself. The downside could be a negative impact on pagers and cellular phones. An additional plus is this reduces your risk of a wireless denial of service attack. Keep in mind that 802.11b operates at the same frequency of a microwave oven. Something as simple as disabling the safety interlock of a consumer microwave oven creates a DOS weapon with up to 1000 watts of 802.11b interference. A more determined black hat may invest in a 'heavy duty' antenna, capable of handling up to a 300W[164] input from the inner workings of a microwave antenna.

5.1.3.4 Fire Suppression

Computers do not mix well water. Fire suppression for computers used to be setup with Halon 1211 gas. Due to the nature of halon 1211being destructive to the ozone layer, it has been replaced with FE-36 in both portable and fixed system fire extinguishers. FE-36 is "much safer" from inhalation point of view, according to the manufacturer, DuPont.

---

[163] http://www.80211-planet.com/tutorials/article/0,4000,10724_1116311,00.html Article on running a Site survey by Jim Geier

[164] http://www.hyperlinktech.com/web/hg2415p.html

FE-13 can be used to prevent explosions. The inert concentration for FE-13 Dupont sates the inerting point for methane/air is 20.5%. While safer than Halon 12111, the Dupont FAQ sates regarding human safety, the following:

"Where possible, evacuate before discharge. Cardio toxicity is what usually LIMITS the quantity of halocarbon agent to which humans can be exposed. This level is determined by challenging dogs with epinephrine in the presence of an air/agent mixture. No cardiac sensitization was observed at 30% FE-13^{TM} in air. DuPont has established a fire emergency exposure limit (FEEL) for FE-13^{TM} of 20% by volume for up to 15 minutes with a 1 minute not-to-exceed ceiling of 23%"

The manufactures requirements are that a fixed fire suppression system brings the contained area up to 16% concentration within 10 seconds with this heaver than air fire suppression gas.

5.2 Disaster Recovery

This topic within security is so broad, the full scope involves many departments within a large firm. For the purposes of the Information Technology department, four areas should be addressed. They are:

A. A 'battle plan'
B. Hot sites
C. Backup
D. Vital records
E. Planning

Justifying the cost of disaster recovery can quickly be ascertained by:

A. How much revenue is lost for each minute of downtime?
B. How many dollars in salaries are being lost per minute due to lack of system availability?

5.2.1 Backups

A keystone in disaster planning is, have *recent backups, and keep them off site*. Smaller sites can do this by utilizing a key person taking backups home. This action step should be firmly planted in your thought process as job.

5.2.1.1 Off Site Storage

Off site storage can be found in a number of different options. A trusted backup operator can take home CD-RW or tape. There are a number of firms that offer backup via remote access by either direct dial with a modem or over the Internet[165]. Companies that want media picked up for safe storage can enlist the services of firms specializing in off site storage[166].

[165]http://www.networkbuyersguide.com/search/105582.htm
[166]http://www.fedsysgrp.com/mediastorage/About%20Datasafe.pdf

5.2.2 Secure Recovery

Integrity is required to have a security guarantee. Typically, this is not found as part of the design of a computer system's architecture. The University of Pennsylvania has developed a system they call AEGIS or Automated Recovery in a Secure Bootstrap Process.[167] In brief, AGEIS assumes:

A. The system board has not been compromised.
B. A certificate authority is being used.
C. A trusted host exists for recovery purposes.

With these assumptions, there are six goals for AGEIS. They are:

A. Allow the AGEIS client and the trusted repository to mutually authenticate their identities with limited or no prior contact.
B. Prevent man in the middle attacks.
C. Prevent replay attacks.
D. Mitigate certain classes of denial of service attacks.
E. Allow the participating parties to agree upon a shared secret and secure manner in order to optimize future of message authentication.
F. Be as simple as possible: Complexity breeds design in an application .

5.2.2.1 Alternate Sites

If the firm is small enough, a *cold site* may suffice. A cold site may be rented storage in a hotel, containing minimal equipment. When major system crash occurs (for whatever reason), renting a room in the hotel can provide an alternate site with minimal downtime.

A *hot site* is fully equipped, with technology including the phone lines, chairs, etc. The site does not have to be sitting idle. Consider using a branch office, or if the firm is too small to have other locations, create a reciprocal relationship with another company. Ideally, the site is of some geographical distance from the original site. Consider how the damage area in a flood, earthquake or other natural occurrences would impact when planning an alternate site.

[167] http://www.cs.columbia.edu/~angelos/Papers/reco.pdf

Vital records should be part of a site, hot or cold. This data should include how to reach employees, contact information for water, power, and telecommunications companies that provide logistical support to your firm.

5.2.3 Disaster Recovery Plan

Once a solid backup procedure is in place, a strategy for what to do when Old Man Murphy jumps from his hiding place and really messes with your company is the next step. When detailing a plan, a good set of guidelines is to pretend you are cub reporter for a news agency. The outline for the cub reporter asks the questions, *Who? What? When? Where? Why?* Loosely that translates into something like:

A. (Who) is responsible for X?
B. (What) does *Who* need to accomplish X?
C. In what sequence (When) will *Who* perform the *What* steps.
D. (Where) will *Who* find the parts to accomplish X?
E. The (Why) typically comes in the form of cost justification.

5.3 Business Continuity

Continuing business as usual, in the face of unusual conditions requires forethought. The reality is there is a mountain of work in putting together a plan of response. The question is where to start? The answer is ISO 17799. The Industry Standards Association is the same international Organization who brought us the OSI (Open Standards Interconnect) model that we use in networking. Specifically ISO 17799is a comprehensive set of controls created with best practices for information security in mind. The first version it did not live long however with version two published in May 1999 this standard is being used at least as a guideline. Note that ISO 17799 has several sub sections including:

A. Security Risk Analysis
B. Compliance Management
C. Auditing

As stated above, even using ISO 17700 as a guideline is not a small effort. A sample of what is involved is available as a PDF found in the footnote[168].

[168]http://www.iso17799software.com/policies.exe

5.3.1 Utilities

Utilities such as electricity in many countries have an amazing 'uptime'. So much so that we tend to take it for granted that it is there. As part of an overall security plan, the assumption must be made that it may not be there. Phone companies use large batteries that are backed up by generators. Larger ISPs follow the same technique. In your firm, make sure you check the manufacturer's rating for battery life of your UPS systems. PDA's lose all data if they completely discharge. Sealed lead acid batteries that need to be swapped out according to the makers' recommendations also run security alarms.

5.3.2 High Availability / Fault Tolerance

High Availability in computers generally refers to another computer that is capable of continuing should one fail. More modern fail over designs are capable of load sharing when both systems are functioning normally and pick up the load on 'fail-over'.

Fault Tolerance is the generally accepted term for disk sub-systems. For example, RAID 1, also known as disk mirroring because it has a complete byte-for-byte copy of the other disk. Disk Duplexing adds to the idea of a second disk controller preventing the violent failure of a single disk controller from trashing both copies. If this does not sound like old ground to you, refer to a Network+ reference[169]

5.3.3 Backups

Backup systems employ either magnetic tape or Erasable Optical media such as DVD-RW or CD-RW.

Full backups make a copy of all data

Differential backup all data that has changed since the full backup.

Incremental backups backup only what has changed since the last backup.

This means if you are using incremental backup, and you have 4 of these since the last full backup, you must restore the full backup first, then the 4 incremental backups in the order they were make.

[169]Your authors have ISBN 159095222-7 covering Network+

Differential back-ups copy everything since the full and the disadvantage is each differential backup takes longer each time.

A computer system knows what to backup and what to pass over via the archive bit.

Be sure to:

A. Test backups for integrity
B. Have multiple copies
C. Store in an off-site secure location
D. Replace backup media often
E. Re-format and reuse tapes. (Tape stretches)

Three different backup schemes are frequently sited. They are:

A. Round Robin One tape for each day of the week.
B. Tower of Hanoi each tape is used a different number of times in a rotation set.
C. Grandfather, father, son tape sets for each daily backup, with a tape for each end of week, followed by 12 tapes for a monthly backup, with one tape for an annual backup.

Grandfather, father, son is the most common scheme; however it wears out the daily tapes quickly.

5.4 Policy and Procedures

Security policy (or collection of policies) can be regarded as the strategy and practices concerning confidentiality, integrity and availability of data[170]. A policy or set of policies cannot be created or purchased until the company philosophy has been clearly defined.

Policies define what is to be protected. Once policies are defined procedures are created to insure the policies that have be decided upon are implemented. Procedures determine how that protection happens. Procedures should also be in place for step-by-step instructions for abnormal events. Just as virtually every public place has an EXIT sign as a guide in the event of a emergency, a procedure should be written in a step-by-step manner for "what to do & how to do it" in the event of negative occurrences.

This work continues with some brief thoughts. All readers are strongly advised to refer to refer to RFC2196[171]. Readers of this document who do or are considering employment in the computer industry should consider studying carefully RFC2196 mandatory.

5.4.1 Security Policy

When creating Security Policy, the first order of business is determining who needs access. The principle of least access is the commonly accepted practice. Management has concerns about data protection. Legal cares about keeping the company out of court. The technical folks have to implement the policies, and users fear policies will impact their ability to get work done.

5.4.1.1 Acceptable Use

There is no single template that applies to all for Acceptable Use Policy. The Electronic Freedom Foundation has a large repository that serves as guidelines by industry[172]. As noted in the general FAQ from the site

[170] http://online.securityfocus.com/infocus/1193
[171] http://www.faqs.org/rfcs/rfc2196.html
[172] http://www.eff.org/CAF/faq/policyy

General guidelines include:

As much as possible use existing legislation and law enforcement mechanisms rather than creating your own.

A. Cite statutes or ordinances based upon which the authority to make this policy is based.
B. Make policies reasonable and narrow
C. Have legal counsel check policy
D. “Common sense, reason and sensitivity should be used to resolve issues in a constructive and positive manner without escalation.”
E. Train staff. Include empathy training.
F. Consider any policy that the limits access carefully
G. Provide a clear description of the behavior that is prohibited so that a reasonably intelligent person will have fair warning

Policies should be clear on a number of specific topics. Examples include:

A. Responsibility of users to protect the data they are using
B. Modifying database entries
C. Providing passwords or sharing user accounts with other workers
D. Copying software
E. Installing software or hardware
F. Policies regarding email and web access
G. Password requirements including how often they must be changed
H. Remote access capabilities
I. Auditing of computer accounts

5.4.1.2 Due Care

As with much of this domain of Security+ the phrase “common sense” comes into play. This may be a poor choice of a phrase. It appears to your authors to be a key term for saying “after considering the situation, has a balance been struck between protecting assets and the privacy of both workers and customers?” Failure of anyone within a company to perform “due care” could result in liability for the company, which could be grounds for termination of an employee.

5.4.1.3 Privacy

Privacy laws exist in most countries. Legal infractions can be found in a variety of forms. For example, without policies in place that state the company has a duty to inspect the companies' computers for unapproved software or to examine systems for Trojan horses could allow for a case where an employee can make a case against the company for invasion of privacy.

Clients of the company have a legal right to expect that their business with a firm is kept private. Some sectors such as the medical industry have additional demands placed on them.

5.4.1.4 Separation of duties

Sensitive operations operate in what is called dual control. This means two people are tasked to a job. For example a bank teller informs a supervisor of a transaction, and the supervisor initials that they reviewed the transaction. The assumption is that an additional person requires collusion. It was Ben Franklin who said, "Two people can keep a secret if one of them is dead."

Specific separation of duties is part of a security policy. A typical procedure would be to have different employees for accounts payable and accounts receivables. For specific examples click on the footnote[173].

5.4.1.5 Need to Know

Need-to-know is a fundamental security principle. When doing a job, information requests must have a genuine need-to-know. Be prepared to justify your request for information.

5.4.1.6 Password Management

Security+ could be looking for two different meanings. Don't know what they want, so both are quickly reviewed.

The topic may mean addressing policy for software programs such as ZDNet's Password Pro 32 (Freeware) to keep track of username, password and system information. This could be a security flaw.

[173] http://www.uh.edu/infotech/pnp/security/rotation.html

This topic could be about the policies for changing passwords, frequency and length, using ALT along with the number keypad to use high-order characters in passwords. May be looking for policies on how administrator passwords are stored. For example: written down, sealed in an envelope and put in a safe.

5.4.1.7 SLA

Service Level Agreements

Delineate services on which On-Line Administrative Systems users can depend.

A. Describe responsibilities of On-Line Administrative System users.
B. Identify roles and specify responsibilities of service providers who support users.
C. Detail problem resolution paths for users and service providers.
D. Describe service levels users should experience when problems or questions arise

5.4.1.8 Disposal / Destruction

The term used for finding discarded information is called dumpster diving. Silicon Valley dumpsters were famous for what surfaced in the trash.

For tape and floppies, use a degasser on magnetic media.

Send hard copy through a shredder.

Secure erasing is a procedure of writing random byte patterns to change the magnetic information to prevent “un-erasing” of data. It is possible to recover data unless this is done. The command FDISK destroys the index of file structures. Think of a library with a manual card file for locating books. Destroying the card file does not make the library go away, just more difficult to find things.

The only way to be really sure that a hard drive can no longer reveal data is to use a file on the surface of the platters. Low level formatting should make a drive reasonably safe.

5.4.1.9 HR Policy

To the typical IT person, human resources do not make a great deal of sense. In brief, the Human Resource department has a wide range of duties, and one area includes legal issues. There are so many governing bodies that must be complied with. To get an idea of the complexity go to the footnote[174] and while at the web site, enter the word security in the search box and look at how many different papers return from the system library.

The HR department creates the handbook that each employee gets with the policies defined. They must also insure that paperwork is in place acknowledging that the employee has read the book, and understands the provisions

It may be part of your duties to help Human Resources understand the different technical issues. For example, Senior Management is encouraging the use of Instant Messaging. In this example a balancing act may need to be spelled out with you installing security software[175] to protect against worm or viruses while the HR department includes polices instructing staff to not reveal sensitive information via Instant Messaging because data is sent in clear text (human readable form).

5.4.1.9.1 Termination

Completely missing in first version. Please use the footnotes to also study on this sub-sub-sub-sub domain. Other sections (ghost workers has good thoughts).

5.4.1.9.2 Hiring

It is the duty of the IT department to work with HR to add/revoke passwords, privileges, etc. for both temporary and permanent staff. This is not being handled properly according to NetworkWorld[176].

[174] http://www.hrnext.com/
[175] http://www.instantmessagingplanet.com/security/article/0,,10818_1379731,00.html
[176] http://www.nwfusion.com/archive/2001/124370_08-27-2001.html

5.4.1.9.3 Code of Ethics

The Information System Security Association[177] is a non-profit organization for security professionals. The association has a code of ethics for its members and looks like good guidelines for anyone working in the computer industry.

The code of ethics from ISSA follows.

A. Perform all professional activities and duties in accordance with the law and the highest ethical principles;
B. Promote good information security concepts and practices;
C. Maintain the confidentiality of all proprietary or otherwise sensitive information encountered in the course of professional activities;
D. Discharge professional responsibilities with diligence and honesty;
E. Refrain from any activities which might constitute a conflict of interest or otherwise damage the reputation of employers, the information security profession, or the Association; and
F. Not intentionally injure or impugn the professional reputation or practice of colleagues, clients, or employers.

5.4.2 Incident Response Policy

Incident Response Policy will vary with the particular needs of an organization. For example, it may be acceptable to disconnect the router that connects to the Internet in one firm. This could lead to serious liability in another firm, such as an ISP. It is beyond this work to detail all forms of policy regarding Incident Response. The footnote[178] will take you to a page supported by Fred Cohen who has more than a half-dozen links to specific policies from the Naval Research Lab to generic templates to be filled in.5.4.2.Incident Response Policy

[177] http://www.issa.org/
[178] http://www.all.net/books/ir/

5.5 Privilege Management

Policies are defined for persons, groups of persons or objects such as data files or hardware such as printers. The policies provide for what is and is not allowed. The terms rights management, access control, and authorization are frequently used interchangeably.

Two terms used are sometimes confused. They are Authorization and Authentication.

Authorization allows a user, Alice, to access a resource. It does not prove Alice's identity. Authentication proves that Alice is really Alice. At least that is how it is suppose to work.

5.5.1 User/Group/Role Management

Authorization can be broken down into different categories.

A. Users are associated with Roles (many-to-many)
B. A Group contains users (one-to-many)

5.5.2 Single Sign-on

Network Operating Systems were designed to behave independently of other systems. If a user wanted a resource in a domain that was different than the domain they are held in the administrator had to tell the second domain to be trusting of the fact that the first domain provided proper security. The alternative was to have that user also have user and group membership on the second domain, this increased administration overhead.

With Single Sign-on, instead of using a password to verify a user when accessing data protected by an Access Control List (ACL), the client is checked using a certificate.

5.5.3 Centralized vs. Decentralized

In a centralized single sign-on data is held at a certain location where authorization takes place. Today, Microsoft Passport is the most talked about centralized single sign-on service. The Liberty Alliance approach is a company implements its own account service, conforming to the Liberty Alliance specifications for interoperability with other companies. Interesting to note that Verisign supports both Passport and Liberty.

5.5.4 Auditing

Sorry. Frozen

5.5.5 MAC/DAC/RBAC

Role Based Access Control

The National Institute of Standards and Technology (NIST) states:

“With role-based access control, access decisions are based on the roles that individual users have as part of an organization. Users take on assigned roles (such as doctor, nurse, teller, and manager). The process of defining roles should be based on a thorough analysis of how an organization operates and should include input from a wide spectrum of users in an organization.

Access rights are grouped by role name, and the use of resources is restricted to individuals authorized to assume the associated role. For example, within a hospital system the role of doctor can include operations to perform diagnosis, prescribe medication, and order laboratory tests; and the role of researcher can be limited to gathering anonymous clinical information for studies”

The NIST has a draft standard with a 51 page PDF that is available to study[179]

RBAC is Policy oriented, yet policy neutral.

Examples of RBAC can be found in Microsoft's Active Directory and Novels Directory Services.

Highlights of RBAC include:

A. Least Privilege
B. Separation of Duties
C. Abstract Permissions
D. Separation of administration and access

Discretionary Access Controls

DAC uses an access policy that restricts access based on the identity of users and/or groups. DAC is identity based. Strict DAC does not allow ownership transfer. For example, Bob can create an object (say a spreadsheet). Bob grants access to Ted. Ted cannot grant access to Carol. Being discretionary means you can choose to implement access control, or not IE. Assign permissions and level of access (Read-Write-Execute) to someone else. Contrast this with Mandatory Access Control.

Mandatory Access Controls

MAC is the most stringent of the security controls. Unlike DAC, you don't have a choice.

In a MAC, 'everything' and everybody gets a label. This label is called a sensitivity or classification label. This allows for multi-level security policies. That is the ability to handle different clearance levels on a single system.

[179] http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf

Labels can be created for levels of trust such as:

Administrator
Power User
User
Guest

And, another set of labels such as:

Accounting
R&D
Sales

These labels can be combined. For example a User and Sales may be allowed to access another label set such as specifications.

While higher authority exists with Accounting and Power User, these labels could be combined to only allow this Labeled Person with an Accounting Label to print to the Labeled Printer Secure Printer from the Labeled File Accounts Receivable. For more information refer to DOD 5200.28-STD[180] (Orange Book)

5.6 Computer forensics

Computer forensics involves the application of investigation and analysis techniques that comply with a legal system The U.S. Department of Justice working in conjunction with a number of groups including the Technical Working Group for Electronic Crime Scene Investigation has created a 93 page PDF that appears to be accepted internationally[181]. A large percentage of the PDF is resources, which are handy. It will not take you a great deal of time to read what to do in the first moments of responding without destroying evidence.

The Cyber crime scene is no different than a physical crime scene in the sense that from a legal standpoint the protection of evidence is critical[182].

[180] http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html
[181] http://www.iwar.org.uk/ecoespionage/resources/cybercrime/ecrime-scene-investigation.pdf Electronic Crime Scene Investigation: A Guide for First Responders
[182] Computer Forensics Incident Response Essentials ISBN 0-201-70719-5

The basic elements include the following:

A. Insure that no possible evidence is destroyed, tampered with, damaged or compromised in any way by the procedures during the investigation.
B. Insure that all possibly relevant evidence is properly handled and protected from electromagnetic or mechanical damage.
C. Insure interruption to the normal process of business as minimal as possible
D. Insure that information acquired with respect to a client-attorney relationship is not divulged.
E. Take care to not violate provisions of law, i.e. Electronic Communications Privacy Act.
F. Contact senior management.
G. A continuing chain of custody is established and maintained.

The chain of custody will be discussed in the next heading.

At the first sign of an issue:

A. Begin a journal with accurate notes, including date and times.
B. Contact management.
C. Take pictures including:
D. Scene
E. Computer Screen

Local laws may require Polaroid as valid evidence.

5.6.1 Chain of Custody

Our research suggests that evidence is frequently not legally valid due to improper handling of evidence. Findings will be examined both for content and breakdowns in the chain of custody.

In short, documentation is **everything**, and you cannot be too careful. Successful legal defense will look under every rock for a hole where evidence tampering could have occurred. The basic tenant for a criminal court case in many countries is, “beyond reasonable doubt.” Civil law is less restrictive going by “based on the preponderance of evidence. And if it appears that evidence was not properly documented, there is no evidence to ponder.”

5.6.2 Preservation of evidence

This sub domain follows very closely to the previous domain, 5.6.1 Chain of Custody. For detailed guidelines follow the footnote to Army Regulation 195.5, Evidence Procedures[183]

Preserving evidence in a legally defensible manner requires that:

A. The handling, labeling and storage of evidence are critical.
B. “Everything” must be 'bagged and tagged'.
C. Stored in an evidence room that has:
D. Low traffic
E. Restricted Access
F. Camera monitoring with the output captured on long play video recorders.
G. Sign In & Sign Out for Chain of Custody

[183] http://www.usapa.army.mil/pdffiles/r195_5.pdf

5.6.3 Collection of evidence

You are encouraged to follow the footnote to Electronic Crime Scene Investigation: A Guide for First Responders. Keep in mind the following points:

A. Do not power down or reboot the system.
B. Do not open files
C. Do unplug the system from the network
D. Do capture running processes and open files
E. If possible, do document current memory and swap files.
F. Do capture mail, DNS and other network service logs supporting hosts.
G. Do a complete port scan of external TCP and UDP port scans of the host.
H. Do contact senior management.
I. Where it is practical to make byte for byte copies of the physical disk without a re-boot, do so.

    If you are making byte for byte (bit stream) copies, it is preferable to use new drives. If you must use existing drives "sanitize" the drives first (low-level format) to eliminate the possibility of a virus.

J. Take pictures of internal components.
K. Document make/model/serial numbers, cable configuration and type.
L. Label evidence "bag and tag".
M. Repeat photographic process with labels on evidence.
N. Document who, what, when (with precise time), how, and why.
O. Have evidence custodian initial each item at the scene, along with initials of worker.
P. Photograph/videotape above procedures through process to the evidence room.
Q. Include hardware for specialized media, i.e., zip disks.
R. Be extra careful with battery powered devices i.e., laptops.

5.7 Risk Identification

1. Identification of risks comes in several forms184.
   Natural disasters
   A. 1.1 Flood
   B. 1.2 Fire
   C. 1.3 Power Failure
   D. 1.4 Equipment Failure

2. External Threats
   A. 2.1 DOS/DDOS attacks
   B. 2.2 Other outside attacks (Man in the Middle, SMURF, etc.)

3. Internal Threats
   A. 3.1 Embezzlement
   B. 3.2 Theft of data/equipment

4. Loss of Credibility
   A. 4.1 Lost revenue due to lack of confidence

5.7.1 Asset Identification

Asset Identification can be a physical label (frequently with a bar code), a tag with RFID (Radio Frequency IDentification) that derives its power from the reader, therefore not requiring a power source at the tag (which is how the exits work at some retail stores).

5.7.2 Risk Assessment

Two factors are computed when considering Risk Assessment. The cost of an event, should it occur and the probability of it occurring. Let's look at an example.

[184] http://www.ustreas.gov/tigta/fy99-strategic_plans/att1-threats-example.html

If a firm has 100 employees who use email have a bottom line cost to the firm (salary+taxes+benefits) of $25 dollars an hour times 100 employees, you have a $2500 per hour cost to the firm. Select a threat, such as viruses. Select a probability that a virus is likely to occur that affects email. In this example we will say 90%. Further let's say it will take you 3 hours to repair the damage. Given this: (3 * $2500) = $7500. * .9 = $6750. That is your cost of the risk

Use this figure against the cost of purchasing a site license for Anti-Virus software. Be careful to make an apple-to-apple comparison. If you estimate that without anti-virus software you would be repairing once a month for a year then it is $6750 * 12 = $81000 verses the annual fee for the site license of the anti-virus software. (Who says you cannot estimate risk? Insurance companies live by formulas like this (only more complex) for determining rates. The example used for anti-virus is known as quantitative risk analysis. A more accurate model may be the qualitative risk analysis. Both are described in more detail at the web site mentioned in the footnote[185].

A great article for software developers can be found in the footnote[186].

5.7.3 Threat Identification

Not clear on what CompTIA intends as a differentiation between risk and threat identification. At this point, your author is questioning how you can determine risk without determining threats. Please review 5.7.Risk Assessment. If you are willing clarify this for me, drop me an email, Tcat@tcat.net ***[Did we find out any more here??? Still vague.]***

5.7.4 Vulnerabilities

NIST states, Vulnerability analysis is an assessment to determine whether vulnerabilities identified during the evaluation of the construction and anticipated operation of the TOE (Target Of Evaluation) or by other methods (e.g. by flaw hypotheses) could allow users to violate the TSP (TOE Security Policy)"[187].

[185] http://www.security-risk-analysis.com/introduction.htm
[186] http://www.processimpact.com/articles/risk_mgmt.html
[187] http://www.niap.nist.gov/tools/CCTB60f-Documentation/CCManual/CCCOVER.HTM

5.8 Education

To be successful, several aspects must be considered in user training. The one of paramount importance is getting users to 'buy into' the fact that the need for security is critical. This requires overcoming the mindset of “this is not my concern". This is accomplished by demonstrating to the end users that only through the continued health and safety of the company will they continue to be employed. Even if an employee is unconcerned because she/he plans to quit anyway, a new perspective employer, in some cases, cannot contact this present firm regarding his/her performance, due to legal issues.

Users must be educated on the fact that corporate data is the most valuable asset the corporate entity has. This step is the groundwork which training builds upon. Without this policy concept firmly in place, there's no foundation to create structure.

It is the administrator's task to convince (and ensure) users, that steps must be taken on an ongoing basis, such as changing passwords. A password taped to a monitor or under a desk blotter renders the password useless. A would-be intruder simply gets a job in maintenance and cleans the office of passwords at night.

Some firms put into place rigorous policies, ranging from no copying of work (files) to diskettes, or only allowing new work to be created outside of the office and then brought into work. Often all disks transported by employees in a particularly diligent company, will have those disks leaving the premises scanned. Security policies require being flexible enough to allow employees to get their job done. Reasonable security combined with convincing employees that protection of the corporate assets is in their best interests is the best possible approach.

5.8.1 Communication

One school of thought is the carrot and stick approach. The carrot is motivational slogans such as “SEC_RITY is not complete without U”

5.8.2 User Awareness

"Ghosts of millions of former workers populate the databases of corporate America. The workers have moved on, but their ghosts linger, awaiting a hacker intent on using the ghost's identity to damage the company's network systems." Says Brian Hook in his Tech Republic article188.

The outdated access account can either be used by a black hat or by an employee whose layoff was handled with less than excellence. From a user awareness viewpoint:

Educate users on the value of strong passwords that are not written down.

Educate users on the value in mandatory frequent changing of passwords. This will disable forgotten accounts.

Give personnel that are being laid off a specific amount of time to download personal data. Failure to do so only creates a hostile former worker.

Create and enforce non-disclosure policies. Non-disclosure is not the same as non-compete clauses. The former employee has a right to work, making non-compete clauses impractical. It is reasonable to expect them to not take with them sensitive supplier or customer data.

5.8.3 Education

Education can come from in-house training based on data from groups such as http://www.sans.org and the reading room.

5.8.4 Online Resources

http://softwaredev.earthweb.com is a great place to start for software developers. Another source is http://www.processimpact.com NIST (National Institute of Standards and Technology) has developed a great deal of material and since we already paid for it with our tax dollars, it is free. Specifically in education, NIST has many hosted programs listed in the footnote[189].

[188] http://www.techrepublic.com/article.jhtml?id=r00620020717hoo01.htm&fromtm=e106-4&_requestid=22643
[189] http://www.nist.gov/public_affairs/edguide.htm

5.9 Documentation

As you have already seen in Forensics, document every move and why you did it. Document your infrastructure and keep these documents secured. Obsolete data should be shredded before disposal.

5.9.1 Standards and Guidelines

Creating a working system requires a roadmap, beyond this, your author is not clear on the difference between this domain (5.9.1) and 5.4.1.x thru 5.4.2 . If you care to enlighten me, drop me your thoughts to Tcat@tcat.net **[Anything more here??? Released to the world so 10,000 eyes can piece this together.]**

5.9.2 Systems Architecture

Hardware and software combine to become a system. Both maintaining and securing a system requires detailing how a system was designed with sub-components detailing hardware and software. MIT has a white paper titled "A fractal Representation for Systems[190]. A template for software project management is listed in the footnote[191]

5.9.3 Change Documentation

Change documentation is a decision memorandum specifying the changes and justification with the accompanying IT spreadsheet reflecting the changes. The memorandum will also state that the changes to the program are approved and should be reflected in a control log.

5.9.4 Logs and Inventories

Logs and inventory documents the maintenance of the institution's computer systems and is used to insure compliance with any warranties or service contracts, schedule regular maintenance and diagnose system or component problems, and document system backups. Records may include: computer equipment inventories; hardware performance reports; component maintenance records (invoices, warranties, maintenance logs, correspondence, maintenance reports, and related records); system backup reports; and backup tape inventories

---

[190]

[191] http://cs.wwc.edu/~aabyan/435/Forms/SPMP.html

Access to secured areas, for example a server room or router should be entered into a logging system that is tamper proof.

5.9.5 Classification

Classification can follow a military classification system such as confidential, secret, and top secret. Categories may be further divided such as “eyes only” which prohibits copies being made. Business may break down into items such as public (marketing), confidential, (sales volume), private (payroll data) and trade secret (formula for Coke).

5.9.5.1 Notification

Notification can be the policies for an unexpected event such as an Intrusion Detection System revealing a possible attack or a planned change such as an upgrade to a piece of equipment which is critical to operations.

5.9.6 Retention/Storage

Procedures for the securing of records both physical and electronic must be spelled out for proper retention, accessibility and security. It is difficult to expect an employee to “do the right thing” when they haven't been told what that is. Depending on industry, different legal requirement come into play

5.9.7 Destruction

Destruction of obsolete information may involve shredding of paper-based data. Higher security may require burning. For electronic media, formatting with a secure erase (overwriting with a random pattern of zeros and ones) may be acceptable. Other security measures may involve using a strong electromagnet such as a bulk tape eraser. Higher security (for example some DOD requirements) may demand the opening of a hard drive and filing the metal on the platters.

Original version has a brief appendix on TCP vs. UDP. Ian thought it created more confusion that reminder, so it is gone. Study, do well, work your favorite discussion board, go pass this test. Best, from all of us.

Special Offer:

We will accept PayPal donations because while this work is free, food, etc. is not. If you do choose to send us $20 USD or more, we will provide you a PDF of our reference work, which will be done when Security+ goes live.

Http://www.alphageekproductions.com