JAY BEALE'S OPEN SOURCE SECURITY SERIES

SYNGRESS®

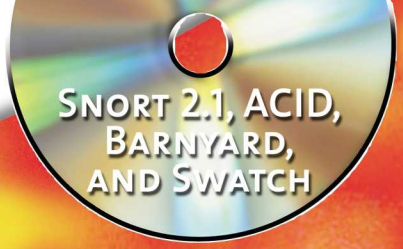
SECOND EDITION OF THE INTERNATIONAL BESTSELLER!

Snort 2.1 Intrusion Detection Second Edition

Featuring Jay Beale and the Snort Development Team
Andrew R. Baker
Brian Caswell
Mike Poor

Foreword by Stephen Northcutt,
Director of Training & Certification, The SANS Institute

Raven Alder • Jacob Babbin • Adam Doxtater
James C. Foster • Toby Kohlenberg • Michael Rash



About the First Edition of *Snort Intrusion Detection*

Overall, I found "Snort 2.0" enlightening. The authors have a powerful understanding of the workings of Snort, and apply it in novel ways.
—Richard Bejtlich, Top 500 Amazon Reviewer

Would I recommend this book to someone already running Snort? Yes! Would I recommend this book to someone considering deploying an IDS? Heck yes! If you attempt to deploy Snort on a production network without reading this book you should be instantly teleported out of your organization and into the "welcome to Walmart" greeter position at the nearest bigbox store of the world's largest corporation.
—Stephen Northcutt, Director, SANs Institute

First, Brian Caswell knows more about Snort than anyone on the planet and it shows here. Secondly, the book is over 500 pages long, and is full of configuration examples. It is the ONE Snort book you need if you're actually running a corporate IDS. This pig flies. Highly recommended.
—A Reader from Austin, TX

This book has proven to be a breath of fresh air. It provides detailed product specifics and is a reliable roadmap to actually rolling out an IDS. And I really appreciate the CD with Snort and the other IDS utilities. The author team is well connected with Snort.org and they obviously had cart blanche in writing this book.
—A Reader from Chestnut Hill, MA

"An awesome book by Snort gurus! This is an incredible book by the guys from snort.org and Sourcefire—this book is just great and covers everything I could ever have thought to ask about Snort 2.0.
—A Syngress customer

Register for Free Membership to

solutions@syngress.com

Over the last few years, Syngress has published many best-selling and critically acclaimed books, including Tom Shinder's *Configuring ISA Server 2000*, Brian Caswell and Jay Beale's *Snort 2.0 Intrusion Detection*, and Angela Orebaugh and Gilbert Ramirez's *Ethereal Packet Sniffing*. One of the reasons for the success of these books has been our unique **solutions@syngress.com** program. Through this site, we've been able to provide readers a real time extension to the printed book.

As a registered owner of this book, you will qualify for free access to our members-only solutions@syngress.com program. Once you have registered, you will enjoy several benefits, including:

- Four downloadable e-booklets on topics related to the book. Each booklet is approximately 20-30 pages in Adobe PDF format. They have been selected by our editors from other best-selling Syngress books as providing topic coverage that is directly related to the coverage in this book.
- A comprehensive FAQ page that consolidates all of the key points of this book into an easy to search web page, providing you with the concise, easy to access data you need to perform your job.
- A "From the Author" Forum that allows the authors of this book to post timely updates links to related sites, or additional topic coverage that may have been requested by readers.

Just visit us at **www.syngress.com/solutions** and follow the simple registration process. You will need to have this book with you when you register.

Thank you for giving us the opportunity to serve your needs. And be sure to let us know if there is anything else we can do to make your job easier.

SYNGRESS®

SYNGRESS®

SECOND EDITION OF THE INTERNATIONAL BESTSELLER!

Snort 2.1 Intrusion Detection Second Edition

Featuring the Snort Development Team
Andrew R. Baker
Brian Caswell
Mike Poor

Foreword by Stephen Northcutt

with
Raven Alder • Jacob Babbin • Jay Beale
Adam Doxtater • James C. Foster
Toby Kohlenberg • Michael Rash

KEY	SERIAL NUMBER
001	TCVGH39764
002	POFG398HB5
003	8NJH2GAWW2
004	HJIRTCV764
005	CVQ23MZX43
006	VB544DM78X
007	HJJ3EDC7NB
008	2WMKEE329N
009	62T7NC9MW5
010	IM6TGH62N5

PUBLISHED BY
Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Snort 2.1 Intrusion Detection, Second Edition

Printed in the United States of America
1 2 3 4 5 6 7 8 9 0
ISBN: 1-931836-04-3

Acquisitions Editor: Christine Kloiber
Technical Editors: Jay Beale, Brian Caswell, Toby Kohlenberg, and Mike Poor
Cover Designer: Michael Kavish
Copy Editor: Beth Roberts
Indexer: Nara Wood
Page Layout and Art: Patricia Lupien

Distributed by O'Reilly & Associates in the United States and Canada.

Acknowledgments

We would like to acknowledge the following people for their kindness and support in making this book possible.

A special thanks to Marty Roesch and the rest of the Snort developers for all their efforts to maintain Snort: Erek Adams, Andrew R. Baker, Brian Caswell, Roman D., Chris Green, Jed Haile, Jeremy Hewlett, Jeff Nathan, Marc Norton, Chris Reid, Daniel Roelker, Dragos Ruiu, JP Vossen, Daniel Wittenberg, and Fyodor Yarochkin.

Syngress books are now distributed in the United States and Canada by O'Reilly & Associates, Inc. The enthusiasm and work ethic at ORA is incredible and we would like to thank everyone there for their time and efforts to bring Syngress books to market: Tim O'Reilly, Laura Baldwin, Mark Brokering, Mike Leonard, Donna Selenko, Bonnie Sheehan, Cindy Davis, Grant Kikkert, Opol Matsutaro, Lynn Schwartz, Steve Hazelwood, Mark Wilson, Rick Brown, Leslie Becker, Jill Lothrop, Tim Hinton, Kyle Hart, Sara Winge, C. J. Rayhill, Peter Pardo, Leslie Crandell, Valerie Dow, Regina Aggio, Pascal Honscher, Preston Paull, Susan Thompson, Bruce Stewart, Laura Schmier, Sue Willing, Mark Jacobsen, Betsy Waliszewski, Dawn Mann, Kathryn Barrett, John Chodacki, and Rob Bullington.

The incredibly hard working team at Elsevier Science, including Jonathan Bunkell, Ian Seager, Duncan Enright, David Burton, Rosanna Ramacciotti, Robert Fairbrother, Miguel Sanchez, Klaus Beran, Emma Wyatt, Rosie Moss, Chris Hossack, and Krista Leppiko, for making certain that our vision remains worldwide in scope.

David Buckland, Daniel Loh, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, Pang Ai Hua, and Joseph Chan of STP Distributors for the enthusiasm with which they receive our books.

Kwon Sung June at Acorn Publishing for his support.

David Scott, Tricia Wilden, Marilla Burgess, Annette Scott, Geoff Ebbs, Hedley Partis, Bec Lowe, and Mark Langley of Woodslane for distributing our books throughout Australia, New Zealand, Papua New Guinea, Fiji Tonga, Solomon Islands, and the Cook Islands.

Winston Lim of Global Publishing for his help and support with distribution of Syngress books in the Philippines.

Series Editor, Technical Editor and Contributor

Jay Beale is a security specialist focused on host lockdown and security audits. He is the Lead Developer of the Bastille project, which creates a hardening script for Linux, HP-UX, and Mac OS X, a member of the Honeynet Project, and the Linux technical lead in the Center for Internet Security. A frequent conference speaker and trainer, Jay speaks and trains at the Black Hat and LinuxWorld conferences, among others. A senior research scientist with the George Washington University Cyber Security Policy and Research Institute, Jay makes his living as a security consultant through the MD-based firm Intelguardians, LLC, where he works on security architecture reviews, threat mitigation and penetration tests against Unix and Windows targets.

Jay wrote the Center for Internet Security's Unix host security tool, currently in use worldwide by organizations from the Fortune 500 to the Department of Defense. He leads the Center's Linux Security benchmark team and, as a core participant in the non-profit Center's Unix teams, is working with private enterprises and US agencies to develop Unix security standards for industry and government.

Aside from his CIS work, Jay has written a number of articles and book chapters on operating system security. He is a columnist for Information Security Magazine and previously wrote a number of articles for SecurityPortal.com and SecurityFocus.com. He co-authored the Syngress international best-seller *Snort 2.0 Intrusion Detection* (ISBN: 1-931836-74-4) and serves as the series and technical editor of the Syngress Open Source Security series. He is also co-author of *Stealing the Network: How to Own a Continent* (Syngress ISBN: 1-931836-05-1). Jay's long-term writing goals include finishing a Linux hardening book focused on Bastille called, *Locking Down Linux*. Formerly, Jay served as the Security Team Director for MandrakeSoft, helping set company strategy, design security products, and pushing security into the third largest retail Linux distribution.

Technical Editors and Contributors

Brian Caswell is a member of the Snort core team, where he is the primary author for the world's most widely used intrusion detection rulesets. He is a member of the Shmoo group, an international not-for-profit, non-milindustrial independent private think tank. He was also a technical editor for *Snort 2.0 Intrusion Detection* (Syngress, ISBN: 1-931836-74-4). Currently, Brian is a Research Engineer within the Vulnerability Research Team for Sourcefire, a provider of one of the world's most advanced and flexible Intrusion Management solutions. Before Sourcefire, Brian was the IDS team leader and all around supergeek for MITRE, a government sponsored think tank. Not only can Brian do IDS, he was a Pokémon Master Trainer for both Nintendo and Wizards of the Coast, working throughout the infamous Pokémon Training League tours. In his free time, Brian likes to teach his young son Patrick to write perl, reverse engineer network protocols, and autocross at the local SCCA events.

Toby Kohlenberg is a Senior Information Security Specialist for Intel Corporation. He does penetration testing, incident response, malware analysis, architecture design and review, intrusion analysis, and various other things that paranoid geeks are likely to spend time dealing with. In the last two years he has been responsible for developing security architectures for world-wide deployments of IDS technologies, secure WLANs, Windows 2000/Active Directory, as well as implementing and training a security operations center. He is also a handler for the Internet Storm Center, which provides plenty of opportunity to practice his analysis skills. He holds the CISSP, GCFW, GCIH, and GCIA certifications. He currently resides in Oregon with his wife and daughters, where he enjoys the 9 months of the year that it rains much more than the 3 months where it's too hot.

Mike Poor is a Founder and Senior Security Analyst for the DC firm Intelgardians Network Intelligence. In his recent past life he has worked for Sourcefire, as a research engineer, and for the SANS Institute as a member of the technical staff. As a consultant, Mike conducts penetration tests, vulnerability assessments, security audits and architecture reviews. His primary job focus however is in intrusion detection, response, and mitigation. Mike currently holds both GSEC and GCIA certifications and is an expert in network engineering and systems, network and web administration. Mike is an Incident Handler for the Internet Storm Center.

Contributors

Raven Alder is a Senior Security Engineer for True North Solutions, a consulting firm specializing in network security design and implementation. She specializes in scalable enterprise-level security, with an emphasis on defense in depth. She designs large-scale firewall and IDS systems, and then performs vulnerability assessments and penetration tests to make sure they are performing optimally. In her copious spare time, she teaches network security for LinuxChix.org and checks cryptographic vulnerabilities for the Open Source Vulnerability Database. Raven lives in the Washington DC area.

Jacob Babbin works as a contractor with a government agency filling the role of Intrusion Detection Team Lead. He has worked in both private industry as a security professional and in government space in a variety of IT security roles. He is a speaker at several IT security conferences and is a frequent assistant in SANS Security Essentials Bootcamp, Incident Handling and Forensics courses. He lives in Virginia.

Andrew R. Baker is a Senior Software Engineer for Sourcefire, Inc. His work experience includes the development and use of intrusion detection systems, security event correlation, as well as vulnerability scanning software, network intrusion analysis and network infrastructure management. Andrew has been involved in the Snort project since 2000. He is the primary developer for Barnyard, which he started working on in 2001 to address performance problems with the existing output plugins. He currently also serves as the mailing list administrator for the Snort project. Andrew has instructed and developed material for the SANS Institute, known for providing information security training and GIAC certifications. He has a bachelors of science in computer science is from the University of Alabama at Birmingham and he is presently attending the R.H. Smith School of Business at the University of Maryland, where he is completing his MBA.

Adam Doxtater (CUSA, MCSE) is a computer engineer for MGM MIRAGE in Las Vegas, NV. Prior to MGM MIRAGE, he was employed as a computer consultant in the greater Las Vegas area. With over 8 years of network administration, he is a very capable and diverse individual. Adam has contributed to the Open Sound System digital audio architecture, allowing it to be ported to a larger UNIX/Linux audience. His Linux-related efforts and columns have been featured in such magazines as eWeek and Network World Fusion, as well as on Web sites such as Slashdot, Linux.com, NewsForge.com, and LinuxWorld.com. Adam is responsible for the launch of the MadPenguin.org Linux portal, which is currently in the top 100,000 sites on the Internet. In the year since its inception, Mad Penguin has become one of the highest-ranking Linux sites, and gathered an impressive and dedicated following. Over the past two and a half years, Adam has contributed to several Syngress books, including *Snort 2.0 Intrusion Detection* (ISBN: 1-931836-74-4) and is truly thankful for the opportunity to reach an audience of that magnitude. Adam owes his accomplishments to his wife, Cristy, and daughter, Amber Michelle. He would also like to thank his entire family for providing the support necessary to make it through some of the hardest times he has ever endured.

James C. Foster, is the Deputy Director, Global Security Development for Computer Sciences Corporation where he is leading the task of developing and delivering managed, educational, informational, consulting, and outsourcing security services. Prior to joining CSC, Foster was the Director of Research and Development for Foundstone Inc. and was responsible for all aspects of product and corporate R&D including corporate strategy and international market expansion. Preceding Foundstone, Foster was a Senior Advisor and Research Scientist with Guardent Inc. (acquired by Verisign in 2004 for $135 Million) and an adjunct author at Information Security Magazine (acquired for an undisclosed amount by TechTarget in 2003.) He is commonly asked to comment on pertinent security issues and has been sited in USAToday, Information Security Magazine, Baseline, Computer World, Secure Computing, and the MIT Technologist. James has co-authored or contributed to *Snort 2.0 Intrusion Detection* (Syngress, ISBN: 1931836744), *Hacking the Code:ASP.NET Web Application Security* (Syngress, ISBN: 1-932266-65-8), and *Special Ops Host and Network Security for Microsoft, Unix, and Oracle* (Syngress, ISBN: 1931836698) as well as *Hacking Exposed, Fourth Edition, Advanced Intrusion Detection, Anti-Hacker Toolkit Second Edition*, and *Anti-Spam Toolkit*. James has attended Yale, Harvard, and the University of Maryland and has an AS, BS, MBA and is currently a Fellow at the University of Pennsylvania's Wharton School of Business.

Michael Rash works as a Security Research Engineer in Columbia, MD for Enterasys Networks, Inc. He is a frequent contributor to Open Source endeavors such as Bastille-Linux and the Netfilter Project, and has written security articles for publications such as Sys Admin Magazine, the Linux Journal, and Information Security Magazine. Michael is the author of Fwsnort and PSAD; two open source security tools designed to blur the boundaries between Iptables firewalls and the Snort Intrusion Detection System. He holds a master's degree in applied mathematics with a concentration in computer security from the University of Maryland, and resides in Maryland with his wife, Katie.

About the CD

The CD-ROM accompanying this book is an archive of many open-source security tools including Snort, Nmap, Nessus, Ethereal, Tcpdump, Ettercap, Nikto, Psad, Iptables, Ebtables, ACID, Barnyard, libnet, and libpcap. Most files are included as a gzip-compressed tar archive, but in some cases .zip compressed files for use on Windows systems are included. Although the latest version of each piece of software at the time of this writing was placed on the CD-ROM, it should be noted that many of the open source projects contained therein have active development cycles and so newer software versions may have been released since publication. An excellent place to find links to the latest releases of each piece of software is by checking on www.freshmeat.net.

Chapter 3 contains the Snort-2.1.2 intrusion detection system, along with an archive of the latest Snort rules. Chapter 5 contains a smorgasbord of tools for offense (Nmap, Nikto, and Nessus), and packet analysis (Ethereal and Tcpdump). Chapter 6 is an archive of the latest release of Ettercap, which definitely falls into the offense category with its capability of performing "man in the middle" attacks on a LAN. Chapters 7 and 8 provide copies of ACID (Analysis Console for Intrusion Databases), Barnyard, and swatch. Chapters 9 and 10 contain copies of the IDS testing/evasion tools Stick and Snot. Chapter 12 is an archive of three active response systems, Snortsam, Fwsnort, and Snort_inline, which automate the process of responding to attacks in real time.

Contents

Foreword

Snort, Information Security Magazine's pick for Open Source Product of the year 2003, is one of the best examples of the IT community working together to build a capability. Please notice I did not say a tool, but rather, a capability. Snort's extensible architecture and open source distribution has long made it an ideal choice for intrusion detection. Snort is amazingly flexible with its plug-in architecture and all its supporting tools such as: ACID, barnyard, and swatch. Snort runs on a large number of hardware platforms and OS configurations, and is one of the most widely ported pieces of security software in the world. Analysts with expensive commercial intrusion detection systems still turn to Snort to fill in the gaps.

The creator of Snort, Marty Roesch, originally envisioned Snort as a lightweight intrusion detection system, and it was initially designed as a network packet sniffer. You can run Snort without specifying a ruleset and view all of the traffic traversing a network on the same network segment. As Snort has continually grown, with enhancements from Marty, as well as with a lot of community-contributed code, it has become a full-featured, real-time IP traffic analysis and packet logging system. And though this is a book about Snort, not about intrusion detection per se, you will learn about all the parts of Snort from how to write a rule to becoming familiar with the numerous auxiliary tools used. For example, Barnyard, Andrew Baker's contribution to Snort, solves one of the hardest problems in intrusion detection: You want the data the IDS collects to end up in a database to facilitate advanced analysis, but databases are slow. If you are running Snort on a busy network a slow database will eventually lead to dropping packets and that is a bad thing, but Barnyard addresses this problem. In short, you will benefit from this book whether you are already running Snort or if you are a beginner.

The years of support for the Snort rule set are an incredible gift to the community. The ruleset and processor bring Snort to life. The Snort rule language is easy to learn and flexible, while the powerful rules and supports enable an advanced analysis capability of all network traffic. You will learn to write rules to determine how to handle any packet you are interested in; you can ignore packets, record them, cause Snort to send an alert, you can do whatever needs to be done. The rule set allows you to specify a number of logging or

alerting methods, Syslog, plain text or XML files are common, but there are a number of additional options. As a new exploit begins to make its way around the Internet, you can be sure that in a matter of hours a new rule specific to the exploit will be published. In fact, the authoring team is a veritable who's who of the intrusion detection community. Brian Caswell, and also James C. Foster have contributed countless hours to making the rule set the lingua franca for intrusion detection. A number of commercial IDS systems can either use Snort rules directly or have a translation function and the Tiny personal firewall uses them as well. Perhaps you have heard of the infamous Gartner Inc. report claming "Intrusion Detection is Dead" and suggesting we all switch to intrusion prevention devices. Amazingly, several of the IPSes I have examined run a subset of the Snort rule set. IDS is not dead: the Snort community is very much alive, kicking and producing.

These folks and the rest of the writing and edit team including: Raven Alder, Jake Babbin, Jay Beale, Adam Doxtater, Toby Kohlenberg, Mike Poor and Michael Rash bring extraordinary capability to the community which is reflected in the book. The authors of this *Snort 2.1 Intrusion Detection, Second Edition* have produced a book with a simple focus, to teach you how to use Snort, from the basics of getting started to advanced rule configuration, they cover all aspects of using Snort, including basic installation, preprocessor configuration, and optimization of your Snort system. I hope you can begin to see why I say Snort is one of the best examples of the IT community working together to build a capability. I am very thankful to have a front row seat to watch the enormously talented security analysts of the Snort community continue to refine and improve the capability of the tools we use. While you are reading though the book, I would encourage you to keep an eye out for the little nuggets that can only come from in-the-trenches experience. My hope is that you will do far more than simply read a book. I would challenge you to make this a step and become an active participant in the defensive information community. Master the material in this book, get your Snort tuned up and running, write a filter and share it, participate in the Snort mailing list, SANS Incidents list, or Security Focus IDS list. I will be looking for you to be part of the author team for Snort 3.0.

— Stephen Northcutt
Director of Training and Certification,
The SANS Institute

Chapter 1

Intrusion Detection Systems

Solutions in this Chapter:

- Introducing Intrusion Detection Systems
- Answering Common IDS Questions
- Fitting Snort into Your Security Architecture
- Determining Your IDS Design and Configuration
- Defining IDS Terminology

☑ Summary

☑ Solutions Fast Track

☑ Frequently Asked Questions

Introducing Intrusion Detection Systems

It's three o'clock in the morning, and Andy Attacker is hard at work. With the results from the latest round of portscans at hand, Andy targets the servers that appear vulnerable. Service by service, Andy fires off exploits, attempting to overflow buffers and overwrite pointers, aiming at taking over other peoples' servers. Some of these attempts are successful. Encouraged, Andy quickly installs rootkits on the compromised machines, opening backdoor access mechanisms, securing the machines enough to lock other attackers out, and consolidating control. Once that is accomplished, Andy begins the next round of scan-and-exploit, from the newly compromised machines.

It's three o'clock in the morning, and a shrill insistent beeping rouses Jennifer Sysadmin from her bed. Blearily, she finds her pager on the nightstand and stares at the message it displays. A customized message alerts her to a Secure Shell overflow attempt… outbound from one of her servers. She is startled into wakefulness. Throwing back the covers and grumbling about the tendency of network malefactors to attack during prime sleeping hours, she grabs her cell phone and heads purposefully for the nearest computer.

It's three o'clock in the morning, and across town, Bob Sysadmin is sleeping peacefully. No pager or cell phone disturbs his rest.

Is Bob's security that much better than Jennifer's, so that he can sleep soundly while she cusses and does damage control? Or has he also been compromised and just doesn't know it yet? With only this information, we don't know. And if he doesn't have an Intrusion Detection System (IDS), neither does Bob. IDSs are a weapon in the arsenal of system administrators, network administrators, and security professionals, allowing real-time reporting of suspicious and malicious system and network activity. While they are not perfect and will not show you every possible attack, IDSs can provide much-needed intelligence about what's really going on on your hosts and your network.

What Is an Intrusion?

To understand what "intrusion detection" does, it is first necessary to understand what an intrusion is. Webster's dictionary defines an intrusion as "the act of thrusting in, or of entering into a place or state without invitation, right, or welcome." For our purposes, an intrusion is simply unauthorized system or network activity on one of your computers or networks. This can take the form of a legitimate user of a system trying to escalate his privileges and gain greater access to

the system than he has been allowed, a remote and unauthenticated user trying to compromise a running service in order to create an account on a system, a virus running rampant through your e-mail system, or many other similar scenarios. Intrusions can come from the deliberately malicious Andy Attackers of the world, or from the terribly clueless Archibald Endusers of the world, who will click on every e-mail attachment sent to them, despite repeated admonitions not to do so. Intrusions can come from a total stranger three continents away, from a disgruntled ex-employee across town, or from your own trusted staff.

OINK!

Detecting malicious activity when it comes from your own employees or users is one of the most important purposes for IDSs in many environments. In fact, a properly implemented IDS that is watched by someone besides your system administrators may be one of the few methods that can actually catch a system administrator when she is doing something malicious. This is one of the main reasons why you should have network security personnel analyzing IDS events and system administrators managing systems.

Legal Definitions

Legally, there are not clear and universal standards for what constitutes an intrusion. There are federal laws about computer crime in many countries, such as the United States and Australia, but none in others. There are various state laws, and regional statutes in place, but not everywhere. Jurisdiction for computer crime cases can be unclear, especially when the laws of the attacker's location are vastly different from the laws in place in the compromised machine's region. To add to this confusion even if an intrusion is clearly within the legal definitions, many law enforcement agencies will not spend time working on it unless there is a clear dollar cost that is greater than some fixed amount. Some agencies use US$10,000 for their guideline, while others use US$100,000—this number varies from place to place.

Another legal concern when using IDSs is privacy. Technically, an IDS is a full content wiretap. In the United States, full content wiretaps are regulated by federal laws, including Title III of the Omnibus Crime Control and Safe Streets Act of 1968 (Title III), 18 U.S.C. §§ 2510–2522 and the Electronic

Communications Privacy Act of 1986. They are also subject to less stringent laws governing Pen Registers or Trap and Trace situations, such as the Pen Register, Trap and Trace Statue "Provider Exception," 18 U.S.C. § 2511(2)(h). These generally involve tapping the characteristics and patterns of traffic without examining the data payload. Under these laws, intercepting network data may be illegal, particularly if it is not done by the network operator in the pursuit of his normal duties or in direct support of an ongoing criminal investigation of a computer trespasser. We strongly advise that you consult your legal department about your particular jurisdiction's laws and the ramifications of deploying an IDS on your network.

Some enterprises rely on the status of their data as "protected trade secrets" under local common uniform trade secrets statutes. Such laws usually require the data to not be known to the public at large, and for some efforts to have been made to secure the data. Therefore, if you're relying on such laws to save you when your data is stolen, you may be in for a nasty shock if the court deems your security measures insufficient. However, the U.S. Economic Espionage Act of 1996 (viewable at www.cybercrime.gov/eea.html) can make such activity a federal crime.

The type and scope of the activity can affect this as well. In computer security forums, there are often arguments about whether portscanning is legal. The answer depends on your jurisdiction. In 1998, Norway ruled that portscanning was not illegal. Michigan law, however, states that unauthorized use or access of a computer is illegal unless you have reason to think the system is designed for public access. Lawyers are still arguing about whether portscanning is "unauthorized use." In some jurisdictions, login banners explicitly prohibiting access are required to prove that a given use of the system was unauthorized. Privacy expectations can play into the equation, too—if the user has an expectation that her system activity may be private, logging and prosecuting her for that activity may be difficult even if it is obviously malicious.

The best practices solution to this legal morass is usually to secure your systems as much as possible, clearly label all accessible services with login banners stating the terms of use, and know your local and federal computer crime laws, if there are any. That will help you protect your systems and identify what is considered an intrusion in your jurisdiction.

Scanning vs. Compromise

When watching network activity, one of the first things that usually jumps out is scanning activity; specifically, lots of scanning activity. Whether it's scanning for particular vulnerabilities or just scanning for open ports, this type of activity is very common on the unfiltered Internet, and on many private networks. Many IDSs are configured to flag scanning activity, and it's not uncommon to see the bulk of your alerts be caused by some form of scanning. While scanning is not necessarily malicious activity in and of itself, and may have legitimate causes (a local system administrator checking his own network for vulnerabilities prior to patching, for example, or a third-party company hired to perform a security audit of your systems), very often scanning is the prelude to an attempted attack. As such, many administrators want to be alerted when they are being scanned. Tracking scanning activity can also be useful for correlation in case of later attack.

Many popular network scanning tools are free, and freely available. A quick Google search will turn up everything from the ping and File Transfer Protocol (FTP) "Grim's Ping" to the full-featured portscanner Nmap, from the commercially available SolarWinds scanner to the vulnerability scanner Nessus. Since scanning tools are so easily accessible, it's not that surprising that they are so widely used.

However, it is important to realize that scanning is not an attempted compromise in and of itself, and should not be treated with the same level of escalated response that an actual attempted attack would merit. There are people who just scan systems out of curiosity and do not intend to attack them. A fellow that we met at a security conference once confided that before he engages in online financial transactions with any business, he scans all the company's machines that he can find. That's his way of determining whether he feels he trusts their security enough to trust them with his money.

It's also important to note that scanning activity is nearly constant. On the Wild West of the modern Internet, all sorts of automated programs are scanning large ranges of addresses, all the time. Some of them might be yours. Network monitoring tools, worms and viruses, automated optimization applications, script kiddies, and more are constantly probing your machines and your network. If you don't make a deliberate effort to filter it out, seeing this traffic on the Internet is a fact of life.

OINK!

While it is important to know when your network is being scanned, you don't want to make the mistake of spending your valuable time tracking down every fool who appears to be scanning your network. One of the best things you can do with information about scans is to track the source IPs that are scanning you and then use them to correlate against alerts for higher priority events or look for repeat scanners. We talk about correlation methods and data analysis in depth in Chapter 8, "Dealing with the Data."

Viruses and Worms—SQL Slammer

Now that we've discussed scanning activity, let's get into a little more detail about some of the actual attempted compromises out there. Another very common type of traffic that you'll see triggering your IDSs is automated worms. Worms and viruses are often good candidates for IDSs, because they have repeatable and consistently identifiable behavior. Even polymorphic worms and viruses that attempt many attack vectors will have some network behavior in common, some traffic pattern that can be matched and detected by your IDS. As an example, let's look at the SQL Slammer worm.

On January 25, 2003, the SQL Slammer worm was released into the wild. Also known as Sapphire, the worm exploits a weakness in the Microsoft Structured Query Language (SQL) server. It sends a 376-byte User Datagram Protocol (UDP) packet to port 1434, overflows a buffer on the SQL server, and gains SYSTEM privileges, the highest possible level of compromise on a Windows operating system. Once it has successfully compromised a host, it starts scanning other IP addresses to further spread.

OINK!

Worms that use multiple attack paths are an excellent example of the value of correlation. The individual alerts from CodeRed or Nimda are common enough, but when they are seen together (as they would be from CodeRed or Nimda), they are a very distinct fingerprint for that worm. As mentioned before, we discuss correlation more in Chapter 8.

It is also worth noting that SQL Slammer is a perfect example of a situation where an "active response" IDS would not be able to prevent

infection, but an inline IDS would. The pluses and minuses of "active response" vs. inline IDS are discussed in Chapter 12, "Active Response."

From the moment of its release, it is estimated that the worm spread worldwide in approximately 10 minutes. Massive amounts of network bandwidth were chewed up by the worm's scanning and propagation attempts. Many systems were compromised. Five of the 13 root Domain Name servers that provide name service to the Internet were knocked down by the worm. You can read the Microsoft advisory about the worm at www.microsoft.com/technet/treeview/default.asp?url=/technet/security/alerts/slammer.asp, and the Computer Emergency Response Team Coordination Center's (CERT-CC) advisory about the worm at www.cert.org/advisories/CA-2003-04.html.

OINK!

The CERT/CC is a center of Internet security expertise located at the Software Engineering Institute, a federally funded research and development center operated by Carnegie-Mellon University.

So, what's a good candidate rule for catching this with an IDS? Obviously, this is just the type of activity that you want to detect on your network. One thing common among every Slammer-infected host is the exploit payload it sends out. And indeed, that's exactly what the Snort IDS signature for the rule matches against. Here's the Snort signature that matches this activity:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"MS-SQL Worm propagation
attempt"; content:"|04|"; depth:1; content:"|81 F1 03 01 04 9B 81 F1 01|";
content:"sock"; content:"send"; reference:bugtraq,5310; classtype:misc-
attack; reference:bugtraq,5311;
reference:url,vil.nai.com/vil/content/v_99992.htm; sid:2003; rev:2;)
```

We'll get into much greater detail about Snort rules and their construction in Chapter 5, "Playing by the Rules," but you can see that the alert is labeled as an attempt at worm propagation, and that it matches UDP traffic headed to our network $HOME_NET on port 1434 with a specific payload. Using this signature, we can detect and enumerate how many attack attempts we saw, and what hosts on our network they were attempting to reach. Massive automated attacks