COURSE TECHNOLOGY
CENGAGE Learning

EC-Council | Press

Penetration Testing
Network & Perimeter Testing

This title maps to

E|CSA™
EC-Council Certified Security Analyst

The Experts: EC-Council

EC-Council's mission is to address the need for well educated and certified information security and e-business practitioners. EC-Council is a global, member based organization comprised of hundreds of industry and subject matter experts all working together to set the standards and raise the bar in Information Security certification and education.

EC-Council certifications are viewed as the essential certifications needed where standard configuration and security policy courses fall short. Providing a true, hands-on, tactical approach to security, individuals armed with the knowledge disseminated by EC-Council programs are securing networks around the world and beating the hackers at their own game.

The Solution: EC-Council Press

The EC-Council | Press marks an innovation in academic text books and courses of study in information security, computer forensics, disaster recovery, and end-user security. By repurposing the essential content of EC-Council's world class professional certification programs to fit academic programs, the EC-Council | Press was formed.

With 8 Full Series, comprised of 25 different books, the EC-Council | Press is set to revolutionize global information security programs and ultimately create a new breed of practitioners capable of combating this growing epidemic of cybercrime and the rising threat of cyber war.

This Certification: E|CSA — EC-Council Certified Security Analyst

The objective of E|CSA is to add value to experienced security professionals by helping them analyze the outcomes of their tests. It is the only in-depth Advanced Hacking and Penetration Testing certification available that covers testing in all modern infrastructures, operating systems, and application environments.

Additional Certifications Covered By EC-Council Press:

Security|5
Security|5 is the entry level certification for anyone interested in learning computer networking and security basics. Security|5 means 5 components of IT security: firewalls, anti-virus, IDS, networking, and web security.

Network|5
Network|5 covers the 'Alphabet Soup of Networking' – the basic core knowledge to know how infrastructure enables a work environment, to help students and employees succeed in an integrated work environment.

C|EH – Certified Ethical Hacker
Information assets have evolved into critical components of survival. The goal of the Ethical Hacker is to help the organization take pre-emptive measures against malicious attacks by attacking the system himself or herself; all the while staying within legal limits.

Wireless|5
Wireless|5 introduces learners to the basics of wireless technologies and their practical adaptation. Learners are exposed to various wireless technologies such as Bluetooth, RFID, IEEE 802.11b/g standard, HomeRF, VoIP, and more; current and emerging standards; and a variety of devices. This certification covers how diverse technologies map to real world applications, requires no pre-requisite knowledge, and aims to educate the learner in simple applications of these technologies.

C|HFI – Computer Hacking Forensic Investigator
Computer Hacking Forensic Investigation is the process of detecting hacking attacks and properly extracting evidence to report the crime and conduct audits to prevent future attacks. The C|HFI materials will give participants the necessary skills to identify an intruder's footprints and to properly gather the necessary evidence to prosecute.

E|NSA – EC-Council Network Security Administrator
The E|NSA program is designed to provide fundamental skills needed to analyze the internal and external security threats against a network, and to develop security policies that will protect an organization's information.

E|DRP – EC-Council Disaster Recovery Professional
E|DRP covers disaster recovery topics, including identifying vulnerabilities, establishing policies and roles to prevent and mitigate risks, and developing disaster recovery plans.

Network and Perimeter Testing

EC-Council | Press

Volume 3 of 5 mapping to

Certification

COURSE TECHNOLOGY
CENGAGE Learning

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

COURSE TECHNOLOGY
CENGAGE Learning®

Network and Perimeter Testing: EC-Council | Press

Course Technology/Cengage Learning Staff:

Vice President, Career and Professional Editorial: Dave Garza

Director of Learning Solutions: Matthew Kane

Executive Editor: Stephen Helba

Managing Editor: Marah Bellegarde

Editorial Assistant: Meghan Orvis

Vice President, Career and Professional Marketing: Jennifer Ann Baker

Marketing Director: Deborah Yarnell

Marketing Manager: Erin Coffin

Marketing Coordinator: Shanna Gibbs

Production Director: Carolyn Miller

Production Manager: Andrew Crouth

Content Project Manager: Brooke Greenhouse

Senior Art Director: Jack Pendleton

EC-Council:

President | EC-Council: Sanjay Bavisi

Sr. Director US | EC-Council: Steven Graham

Library of Congress Control Number: 2010926142

ISBN-13: 978-1-4354-8368-2

ISBN-10: 1-4354-8368-5

Cengage Learning
5 Maxwell Drive
Clifton Park, NY 12065-2919
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at: **international.cengage.com/region**

Cengage Learning products are represented in Canada by
Nelson Education, Ltd.

For more learning solutions, please visit our corporate website at **www.cengage.com**

Printed in the United States of America
1 2 3 4 5 6 7 12 11 10

Brief Table of Contents

Table of Contents

Preface

Hacking and electronic crimes sophistication has grown at an exponential rate in recent years. In fact, recent reports have indicated that cyber crime already surpasses the illegal drug trade! Unethical hackers better known as *black hats* are preying on information systems of government, corporate, public, and private networks and are constantly testing the security mechanisms of these organizations to the limit with the sole aim of exploiting it and profiting from the exercise. High profile crimes have proven that the traditional approach to computer security is simply not sufficient, even with the strongest perimeter, properly configured defense mechanisms like firewalls, intrusion detection, and prevention systems, strong end-to-end encryption standards, and anti-virus software. Hackers have proven their dedication and ability to systematically penetrate networks all over the world. In some cases *black hats* may be able to execute attacks so flawlessly that they can compromise a system, steal everything of value, and completely erase their tracks in less than 20 minutes!

The EC-Council Press is dedicated to stopping hackers in their tracks.

About EC-Council

The International Council of Electronic Commerce Consultants, better known as EC-Council was founded in late 2001 to address the need for well-educated and certified information security and e-business practitioners. EC-Council is a global, member-based organization comprised of industry and subject matter experts all working together to set the standards and raise the bar in information security certification and education.

EC-Council first developed the *Certified Ethical Hacker,* C|EH program. The goal of this program is to teach the methodologies, tools, and techniques used by hackers. Leveraging the collective knowledge from hundreds of subject matter experts, the C|EH program has rapidly gained popularity around the globe and is now delivered in over 70 countries by over 450 authorized training centers. Over 80,000 information security practitioners have been trained.

C|EH is the benchmark for many government entities and major corporations around the world. Shortly after C|EH was launched, EC-Council developed the *Certified Security Analyst,* E|CSA. The goal of the E|CSA program is to teach groundbreaking analysis methods that must be applied while conducting advanced penetration testing. E|CSA leads to the *Licensed Penetration Tester,* L|PT status. The *Computer Hacking Forensic Investigator,* C|HFI was formed with the same design methodologies above and has become a global standard in certification for computer forensics. EC-Council through its impervious network of professionals, and huge industry following has developed various other programs in information security and e-business. EC-Council Certifications are viewed as the essential certifications needed where standard configuration and security policy courses fall short. Providing a true, hands-on, tactical approach to security, individuals armed with the knowledge disseminated by EC-Council programs are securing networks around the world and beating the hackers at their own game.

About the EC-Council | Press

The EC-Council | Press was formed in late 2008 as a result of a cutting edge partnership between global information security certification leader, EC-Council and leading global academic publisher, Cengage Learning. This partnership marks a revolution in academic textbooks and courses of study in Information Security, Computer Forensics, Disaster Recovery, and End-User Security. By identifying the essential topics and content of EC-Council professional certification programs, and repurposing this world class content to fit academic programs, the EC-Council | Press was formed. The academic community is now able to incorporate this powerful cutting edge content into new and existing Information Security programs. By closing the gap between academic study and professional certification, students and instructors are able to leverage the power of rigorous academic focus and high demand industry certification. The EC-Council | Press is set to revolutionize global information security programs and ultimately create a new breed of practitioners capable of combating the growing epidemic of cybercrime and the rising threat of cyber-war.

Penetration Testing Series

The EC-Council | Press *Penetration Testing* series, preparing learners for ElCSA/LPT certification, is intended for those studying to become Network Server Administrators, Firewall Administrators, Security Testers, System Administrators and Risk Assessment professionals. This series covers a broad base of topics in advanced penetration testing and security analysis. The content of this program is designed to expose the learner to groundbreaking methodologies in conducting thorough security analysis, as well as advanced penetration testing techniques. Armed with the knowledge from the Penetration Testing series, learners will be able to perform the intensive assessments required to effectively identify and mitigate risks to the security of the organization's infrastructure. The series when used in its entirety helps prepare learners to take and succeed on the ElCSA, Certified Security Analyst certification exam.

Books in Series:

- *Penetration Testing: Security Analysis*/1435483669
- *Penetration Testing: Procedures and Methodologies*/1435483677
- *Penetration Testing: Network and Perimeter Testing*/1435483685
- *Penetration Testing: Communication Media Testing*/1435483693
- *Penetration Testing: Network Threat Testing*/1435483707

Network and Perimeter Testing

Network and Perimeter Testing coverage includes techniques and tools to perform a thorough penetration test. Discussion includes legal requirements, rules of engagement, how to plan and schedule a test, how to perform vulnerability analysis, external and internal penetration testing, and techniques to conduct an advanced penetration test.

Chapter Contents

Chapter 1, *Advanced Googling* discusses the use of Google to reveal vulnerabilities that are potentially open to anyone with internet access. Chapter 2, *Routers and Swtiches Penetration Testing* includes coverage of how to identify a router host name and router protocols, as well as different tests that can be employed in testing routers and switches. Chapter 3, *Firewall Penetration Testing*, introduces concepts employed in testing firewalls. Chapter 4, *IDS Penetration Testing* familiarizes the reader with the different types of intrusion detection systems and how to test their effectiveness. Chapter 5, *Physical Security and Stolen Laptop, PDA and Cell Phone Penetration Testing*, covers the importance of physical security of and how to test the security of laptops, cell phones, and PDA's. Chapter 6, *E-Mail Security Penetration Testing*, discusses how e-mail is a prime target for attackers and ways to test the security of e-mail servers. Chapter 7, *Security Patches Penetration Testing*, describes the concept of patch management and enumerates tools that administrators can use to automate the process as well as introducing steps involved in performing penetration testing of security patches.

Chapter Features

Many features are included in each chapter and all are designed to enhance the learner's learning experience. Features include:

- *Objectives* begin each chapter and focus the learner on the most important concepts in the chapter.
- *Key Terms* are designed to familiarize the learner with terms that will be used within the chapter.
- *Chapter Summary*, at the end of each chapter, serves as a review of the key concepts covered in the chapter.
- *Hands-On Projects* encourage the learner to apply the knowledge they have gained after finishing the chapter. Chapters covering the Licensed Penetration Testing (LPT) materials do not have Hands-On Projects. The LPT content does not lend itself to these types of activities. Files for the *Hands-On Projects* can be found on the Student Resource Center. Note: you will need your access code provided in your book to enter the site. Visit *www.cengage.com/community/eccouncil* for a link to the Student Resource Center.

Student Resource Center

The Student Resource Center contains all the files you need to complete the Hands-On Projects found at the end of the chapters. Chapters covering the Licensed Penetration Testing (LPT) materials do not have Hands-On Projects. The LPT content does not lend itself to these types of activities. Access the Student Resource Center with the access code provided in your book. Instructions for logging onto the Student Resource Site are included with the access code. Visit *www.cengage.com/community/eccouncil* for a link to the Student Resource Center.

Additional Instructor Resources

Free to all instructors who adopt the *Network and Perimeter Testing* book for their courses is a complete package of instructor resources. These resources are available from the Course Technology web site, www.cengage.com/coursetechnology, by going to the product page for this book in the online catalog, click on the Companion Site on the Faculty side; click on any of the Instructor Resources in the left navigation and login to access the files. Once you accept the license agreement, the selected files will be displayed.

Resources include:

- *Instructor Manual*: This manual includes course objectives and additional information to help your instruction.
- *ExamView Testbank*: This Windows-based testing software helps instructors design and administer tests and pre-tests. In addition to generating tests that can be printed and administered, this full-featured program has an online testing component that allows students to take tests at the computer and have their exams automatically graded.
- *PowerPoint Presentations*: This book comes with a set of Microsoft PowerPoint slides for each chapter. These slides are meant to be used as a teaching aid for classroom presentations, to be made available to students for chapter review, or to be printed for classroom distribution. Instructors are also at liberty to add their own slides.
- *Labs*: Additional Hands-on Activities to provide additional practice for your students.
- *Assessment Activities*: Additional assessment opportunities including discussion questions, writing assignments, internet research activities, and homework assignments along with a final cumulative project.
- *Final Exam*: Provides a comprehensive assessment of *Network and Perimeter Testing* content.

Cengage Learning Information Security Community Site

This site was created for learners and instructors to find out about the latest in information security news and technology.

Visit *community.cengage.com/infosec* to:

- Learn what's new in information security through live news feeds, videos and podcasts.
- Connect with your peers and security experts through blogs and forums.
- Browse our online catalog.

How to Become E|CSA Certified

EC-Council Certified Security Analyst (E|CSA) complements the Certified Ethical Hacker (C|EH) certification by exploring the analytical phase of ethical hacking. While C|EH exposes the learner to hacking tools and technologies, E|CSA takes it a step further by exploring how to analyze the outcome from these tools and technologies.

E|CSA is a relevant milestone towards achieving EC-Council's Licensed Penetration Tester (LPT), which also ingrains the learner in the business aspect of penetration testing. The LPT standardizes the knowledge base for penetration testing professionals by incorporating the best practices followed by experienced experts in the field. The LPT designation is achieved via an application/approval process. LPT is obtained by holding both the CEH and ECSA, then completing the application process for LPT found here at *http://www.eccouncil.org/lpt.htm*.

EICSA Certification exams are available through Authorized Prometric Testing Centers. To finalize your certification after your training, you must:

1. Purchase an exam voucher from the EC-Council Community Site at Cengage: *www.cengage.com/community/eccouncil*.
2. Once you have your Exam Voucher, visit *www.prometric.com* and schedule your exam.
3. Take and pass the EICSA certification examination with a score of 70% or better.

About Our Other EC-Council | Press Products

Ethical Hacking and Countermeasures Series

The EC-Council | Press *Ethical Hacking and Countermeasures* Series is intended for those studying to become security officers, auditors, security professionals, site administrators, and anyone who is concerned about or responsible for the integrity of the network infrastructure. The series includes a broad base of topics in offensive network security, ethical hacking, as well as network defense and countermeasures. The content of this series is designed to immerse the learner into an interactive environment where they will be shown how to scan, test, hack and secure information systems. A wide variety of tools, viruses, and malware is presented in these books, providing a complete understanding of the tactics and tools used by hackers. By gaining a thorough understanding of how hackers operate, ethical hackers are able to set up strong countermeasures and defensive systems to protect their organization's critical infrastructure and information. The series when used in its entirety helps prepare readers to take and succeed on the CIEH certification exam from EC-Council.

Books in Series:

- *Ethical Hacking and Countermeasures: Attack Phases*/143548360X
- *Ethical Hacking and Countermeasures: Threats and Defense Mechanisms*/1435483618
- *Ethical Hacking and Countermeasures: Web Applications and Data Servers*/1435483626
- *Ethical Hacking and Countermeasures: Linux, Macintosh and Mobile Systems*/1435483642
- *Ethical Hacking and Countermeasures: Secure Network Infrastructures*/1435483650

Computer Forensics Series

The EC-Council | Press *Computer Forensics* Series, preparing learners for CIHFI certification, is intended for those studying to become police investigators and other law enforcement personnel, defense and military personnel, e-business security professionals, systems administrators, legal professionals, banking, insurance and other professionals, government agencies, and IT managers. The content of this program is designed to expose the learner to the process of detecting attacks and collecting evidence in a forensically sound manner with the intent to report crime and prevent future attacks. Advanced techniques in computer investigation and analysis with interest in generating potential legal evidence are included. In full, this series prepares the learner to identify evidence in computer related crime and abuse cases as well as track the intrusive hacker's path through client system.

Books in Series:

- *Computer Forensics: Investigation Procedures and Response*/1435483499
- *Computer Forensics: Investigating Hard Disks, File and Operating Systems*/1435483502
- *Computer Forensics: Investigating Data and Image Files*/1435483510
- *Computer Forensics: Investigating Network Intrusions and Cybercrime*/1435483529
- *Computer Forensics: Investigating Wireless Networks and Devices*/1435483537

Network Defense Series

The EC-Council | Press *Network Defense Series*, preparing learners for INSA certification, is intended for those studying to become system administrators, network administrators and anyone who is interested in network security technologies. This series is designed to educate learners, from a vendor neutral standpoint, how to defend the networks they manage. This series covers the fundamental skills in evaluating internal and external threats to network security, design, and how to enforce network level security policies, and ultimately protect an organization's information. Covering a broad range of topics from secure network fundamentals, protocols & analysis, standards and policy, hardening infrastructure, to configuring IPS, IDS and firewalls, bastion host and honeypots, among many other topics, learners completing this series will have a full understanding

of defensive measures taken to secure their organizations information. The series when used in its entirety helps prepare readers to take and succeed on the NISA, Network Security Administrator certification exam from EC-Council.

Books in Series
- *Network Defense: Fundamentals and Protocols*/1435483553
- *Network Defense: Security Policy and Threats*/1435483561
- *Network Defense: Perimeter Defense Mechanisms*/143548357X
- *Network Defense: Securing and Troubleshooting Network Operating Systems*/1435483588
- *Network Defense: Security and Vulnerability Assessment*/1435483596

Cyber Safety/1435483715

Cyber Safety is designed for anyone who is interested in learning computer networking and security basics. This product provides information cyber crime; security procedures; how to recognize security threats and attacks, incident response, and how to secure internet access. This book gives individuals the basic security literacy skills to begin high-end IT programs. The book also prepares readers to take and succeed on the Security|5 certification exam from EC-Council.

Wireless Safety/1435483766

Wireless Safety introduces the learner to the basics of wireless technologies and its practical adaptation. *Wireless|5* is tailored to cater to any individual's desire to learn more about wireless technology. It requires no pre-requisite knowledge and aims to educate the learner in simple applications of these technologies. Topics include wireless signal propagation, IEEE and ETSI Wireless Standards, WLANs and Operation, Wireless Protocols and Communication Languages, Wireless Devices, and Wireless Security Network The book also prepares readers to take and succeed on the Wireless|5 certification exam from EC-Council.

Network Safety/1435483774

Network Safety provides the basic core knowledge on how infrastructure enables a working environment. Intended for those in an office environment and for the home user who wants to optimize resource utilization, share infrastructure and make the best of technology and the convenience it offers. Topics include foundations of networks, networking components, wireless networks, basic hardware components, the networking environment and connectivity as well as troubleshooting. The book also prepares readers to take and succeed on the Network|5 certification exam from EC-Council.

Disaster Recovery Series

The *Disaster Recovery Series* is designed to fortify virtualization technology knowledge of system administrators, systems engineers, enterprise system architects, and any IT professional who is concerned about the integrity of the their network infrastructure. Virtualization technology gives the advantage of additional flexibility as well as cost savings while deploying a disaster recovery solution. The series when used in its entirety helps prepare readers to take and succeed on the EICDR and EICVT, Disaster Recovery and Virtualization Technology certification exam from EC-Council. The EC-Council Certified Disaster Recovery and Virtualization Technology professional will have a better understanding of how to setup Disaster Recovery Plans using traditional and virtual technologies to ensure business continuity in the event of a disaster.

Books in Series
- *Disaster Recovery* /1435488709
- *Virtualization Security*/1435488695

Acknowledgements

Michael H. Goldner is the Chair of the School of Information Technology for ITT Technical Institute in Norfolk Virginia, and also teaches bachelor level courses in computer network and information security systems. Michael has served on and chaired ITT Educational Services Inc. National Curriculum Committee on Information Security. He received his Juris Doctorate from Stetson University College of Law, his undergraduate degree from Miami University and has been working over fifteen years in the area of Information Technology. He is an active member of the American Bar Association, and has served on that organization's Cyber Law committee. He is a member of IEEE, ACM and ISSA, and is the holder of a number of industrially recognized certifications including, CISSP, CEH, CHFI, CEI, MCT, MCSE/Security, Security +, Network + and A+. Michael recently completed the design and creation of a computer forensic program for ITT Technical Institute, and has worked closely with both EC-Council and Delmar/Cengage Learning in the creation of this EC-Council Press series.

Chapter 1

Advanced Googling

Objectives

After completing this chapter, you should be able to:

- Use a site query
- Use an intitle:index.of query
- Use an error | warning query
- Use a login | logon query
- Use an admin | administrator query
- Locate source code with common strings
- Locate vulnerable targets
- Locate targets via demonstration pages
- Locate targets via source code
- Find vulnerable Web application examples
- Locate targets via CGI scanning
- Use a single CGI scan–style query
- Find directory listings
- Find Web server software error messages

Key Terms

Default pages pages that come with Web software that allow an administrator to connect to the Web server with a browser to validate that the Web software was installed correctly

Google Advanced Search Form a form that allows the user to conduct search techniques not offered by the regular search bar

Help desks places to go for common problems that users encounter when using a Web site

Intranet a generic term that describes an internal network or one confined to a small group

Login portal a Web page that serves as a "front door" to a Web site

Query a search term

Introduction to Advanced Googling

Many of the techniques involved in penetration testing involve software specifically designed to break passwords or control systems with malicious code. However, before any of these techniques can be used, extensive research must be done to find vulnerable targets. This is done most effectively through the use of Google. By using specific ***queries***, or search terms, the user can accomplish much of the work of hacking a system through publicly accessible means. This type of research is very important in penetration testing, because it reveals vulnerabilities that are potentially open to anyone with Internet access. This unit introduces a number of important queries along with their uses.

Common Queries

site

The site query is absolutely invaluable during the information-gathering phase of an assessment. Combined with a host or domain name, this query presents results that can be overwhelming, to say the least. However, the site query is meant to be used as a base search, not necessarily as a standalone search. It is possible to scan through every single page of results from this query, but in most cases it is just impractical.

Important information can be gained from a straight-up site search, however. First, remember that Google results are listed in page-ranked order. In other words, the most popular pages float to the top of the results. This means it is easy to get a quick idea about what the rest of the Internet thinks is most worthwhile about a site. The implications of this information are varied, but at a basic level, users can at least get an idea of the public image or consensus about an online presence by looking at what floats to the top. Outside the specific site search itself, it can be helpful to read into the context of links originating from other sites.

The site search can also be used to gather information about the servers and hosts that a target hosts. Using simple reduction techniques can give a quick idea about a target's online presence.

Consider the simple example of **site:washingtonpost.com–site:www.washingtonpost.com**, as shown in Figure 1-1. This query effectively locates pages on the *washingtonpost.com* domain other than *www.washingtonpost.com*. The figure shows three other domains: *yp.washingtonpost.com*, *eg.washingtonpost.com*, and *topics.washingtonpost.com*.

intitle:index.of

The intitle:index.of query is the universal search for directory listings, as shown in Figure 1-2. In most cases, this search applies only to Apache-based servers. However, due to the overwhelming number of Apache-derived Web servers on the Internet, there is a good chance that the server being profiled will be Apache based. Using an **intitle:index.of** query against a target is fast and easy, and could produce a large payoff.

error | warning

Error messages can reveal a great deal of information about a target. Often overlooked, error messages can provide insight into the application or operating system software a target is running, the architecture of the network the target is on, information about users on the system, and much more. Not only are error messages informative, they are prolific. A query of **intitle:error** results in over 55 million results, as shown in Figure 1-3.

Some error messages don't actually display the word *error*, as shown in Figure 1-4.

This error page reveals usernames, filenames, path information, IP addresses, and line numbers, yet the word *error* does not occur anywhere on the page. Warning messages are almost as prolific as error messages, and they can be generated from application programs. In some cases, however, the word *warning* is specifically written into the text of a page to alert the Web user that something important has happened or is about to happen.

login | logon

A ***login portal*** is a "front door" to a Web site. Login portals can reveal the software and operating system of a target, and in many cases help documentation is linked from the main page of a login portal. These documents are designed to assist users who run into problems during the login process, such as forgetting a username or password. These documents can provide clues that might help an attacker gain access to the site.

Figure 1-1 Using a simple site query can reveal other domains that a target is associated with.

Many times, documentation linked from login portals lists e-mail addresses, phone numbers, or URLs of human assistants who can help a troubled user regain lost access. These assistants, or help desk operators, are perfect targets for a social engineering attack. Even the smallest security testing team should not be without a social engineering whiz. The vast majority of all security systems have one common weakest link: a human behind a keyboard. The words *login* and *logon* are widely used on the Internet, occurring on over 12 million pages.

Figure 1-5 shows the results of a **login | logon** query. Notice that the very first result for this query shows the words *login trouble* in the text of the page. This link provides help to users who have forgotten their login credentials. It is exactly these types of links that security testers might use to gain access to a system.

username | userid | employee.ID | "your username is"

There are many different ways to obtain a username from a target system. Even though a username is the less important half of most authentication mechanisms, it should at least be marginally protected from outsiders.

Figure 1-6 shows that even sites that reveal very little information in the face of a barrage of probing Google queries return many potentially interesting results to this query. To avoid implying anything negative about the target used in this example, some details of the figure have been edited. The mere existence of the word *username* in a result is not indicative of a vulnerability, but results from this query provide a starting point for an attacker.

password | passcode | "your password is"

The word *password* is so common on the Internet that there are over 73 million results for this one-word query. Launching a query for derivations of this word makes little sense unless the user actually combines that search with a site query.

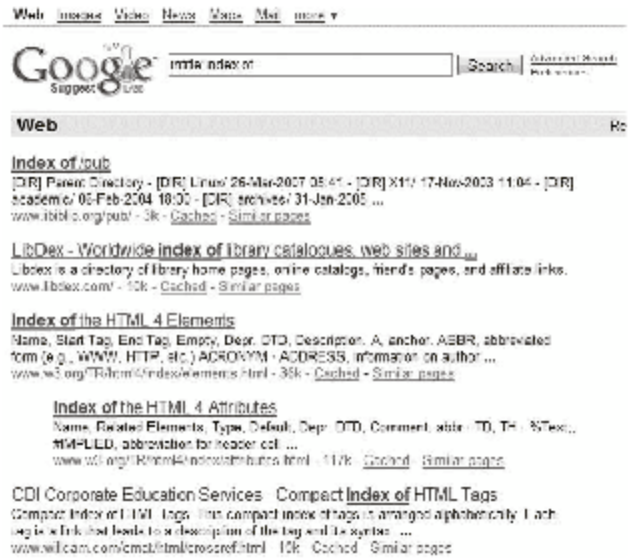


Figure 1-2 Using the **intitle:index.of** query reveals directory listings that will help the user map the target site.

Figure 1-3 Error messages are prolific and can reveal important information about a system.

During an assessment, it is very likely that results for this query, combined with a site query, will include pages that provide help to users who have forgotten their passwords. In some cases, this query will locate pages that provide policy information about the creation of a password. This type of information can be used in an intelligent guessing or even a brute-force campaign against a password field.

Despite how this query looks, it is quite uncommon for this type of query to return actual passwords. Passwords do exist on the Web, but this query isn't well suited for locating them.

Figure 1-4 Some error messages do not display the word *error*.

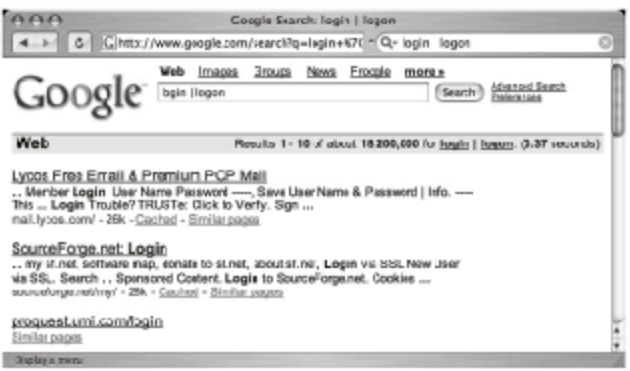


Figure 1-5 The **login | logon** query can reveal help sites for users who have forgotten their usernames or passwords.

Like the login portal and username queries, this query can provide an informational foothold into a system. Although this query is somewhat useless without the site query, Figure 1-7 shows that the first hit for this query is a "forgotten password" page—exactly the type of page that can be informative.

admin | administrator

The word *administrator* is often used to describe the person in control of a network or system. There are so many references to the word on the Web that a query for **admin | administrator** weighs in at over 15 million results. This suggests that these words will likely be referenced on a site that a user is charged with assessing.

However, the value of these and other words in a query does not lie in the number of results but in the contextual relevance of the words. In Figure 1-8, the word *administrator* is used in several common ways, each of which can provide relevance during an assessment.

Figure 1-6 A username query can reveal actual usernames or the method by which usernames are created.

Figure 1-7 A password query rarely reveals actual passwords, but it can turn up useful help pages.

admin login

The word *administrator* can also be used to locate administrative login pages or login portals. A query for "administrative login" returns 150,000 results, many of which are administrative login pages. A security tester can profile Web servers using seemingly insignificant clues found on these types of login pages. Most login portals provide clues about what software is in use on the server, drawing attackers who are armed with an exploit for that particular type of software. Remember that Google performs autostemming; a search for "admin login" returns approximately 1.3 million results, including results that were autostemmed to include the phrase "administrator login." As shown in Figure 1-9, many of the results are for administrative login pages.

Another interesting use of the administrator derivations is to search for them in the URL of a page using an inurl: search. If the word *admin* is found in the hostname, a directory name, or a filename within a URL, there is a chance that the URL has some administrative function, making it interesting from a security standpoint.

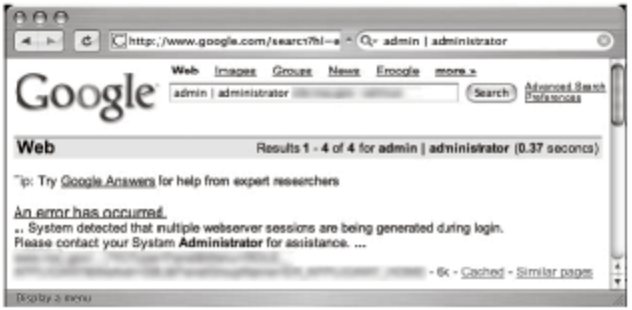


Figure 1-8 An **admin | administrator** query can reveal important information about access to a site.

Figure 1-9 The **"admin login"** query often brings up administrative login pages.

–ext:html –ext:htm –ext:shtml –ext:asp –ext:php

The **–ext:html –ext:htm –ext:shtml –ext:asp –ext:php** query uses ext, a synonym for the filetype operator, and is a negative query. It returns no results when used alone and should be combined with a site query to work properly. The idea behind this query is to exclude some of the most common Internet file types in an attempt to find files that might be of use.

There are certainly many HTML, PHP, and ASP pages that reveal interesting information. The file extensions used in this search were selected very carefully. The site *www.filext.com* contains a list of every known

File Extension	Approximate Number of Hits
HTML	17,800,000
PHP	16,500,000
HTM	16,100,000
ASP	15,400,000
PDF	11,600,000
CGI	11,100,000
CFM	9,870,000
SHTML	8,770,000
JSP	7,370,000
ASPX	7,110,000
PL	5,660,000
PHP3	3,870,000
DLL	3,340,000
SWF	2,260,000
PHTML	2,250,000
DOC	2,120,000
FCGI	1,850,000
TXT	1,700,000
MV	1,060,000
JHTML	930,000

Figure 1-10 These are the top 20 results of a query of every file extension.

file extension. Each entry in the list of over 8,000 file extensions was converted into a Google query using the filetype operator. For example, if a user wants to search for a PDF extension, a query such as **filetype:PDF** could be used to get the number of known results on the Internet. This Google query was performed for each and every known file extension from *filext.com*. Once the results were gathered, they were sorted in descending order by the number of hits. The top 20 results of this query are shown in Figure 1-10.

This figure reveals the most common file types on the Internet. Typically, a query like this, submitted with a site query, will reveal a list of results worth investigating. In some cases, this query will need to be refined, especially if the site uses a less common server-generated file extension. For example, consider this query combined with a site query, as shown in Figure 1-11.

As revealed in the search results, this site uses the ASPX extension for some Web content. By adding – ext:aspx to the query and resubmitting it, that type of content is removed from the search results. This modified search reveals some interesting information, as shown in Figure 1-12.

inurl:temp | inurl:tmp | inurl:backup | inurl:bak

The **inurl:temp | inurl:tmp | inurl:backup | inurl:bak** query, combined with a site query, searches for temporary or backup files or directories on a server. Although there are many possible naming conventions for temporary or backup files, this search focuses on the most common terms. Since this search uses the inurl operator, it will also locate files that contain these terms as file extensions, such as index.html.bak, for example. Modifying this search to focus on file extensions is tricky because this requires ORing the filetype operator (which is often flaky, since filetype also requires a search term that gets lost in the mess of ORs) and also limits the search, leaving out temporary or backup directories.

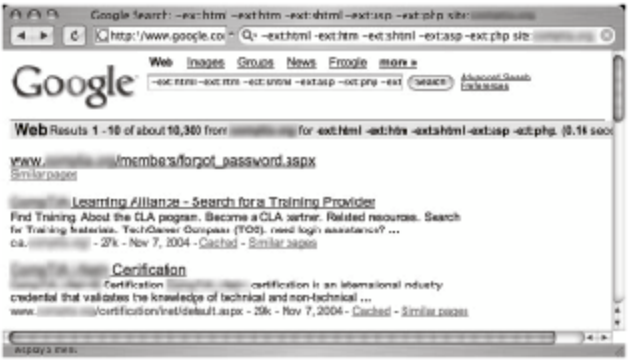


Figure 1-11 This query ignores common file extensions, such as html, so that more valuable sites come up in the search.

Figure 1-12 Removing the aspx extension from this search turned up valuable hits.

Google Advanced Search Form Queries

The ***Google Advanced Search Form*** is a form that allows the user to conduct search techniques not offered by the regular search bar. It provides the following options for a search:

- Allows for selecting or avoiding pages with more accuracy
- Focuses on options, which results in more targeted and accurate searches
- Allows categorizing the search by using word, exact phrase, or at least one word

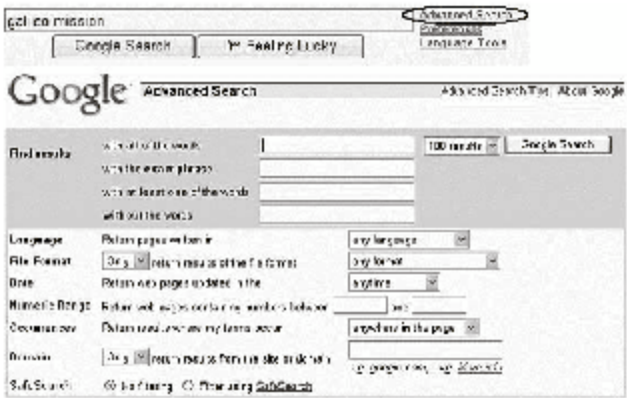


Figure 1-13 Google's advanced search gives the user a number of useful options.

A tester can use the following steps to initiate an advanced Google search, as shown in Figure 1-13:

1. Go to Google's standard search text box.
2. Click **Advanced Search** to the right side of the search box.

Table 1-1 shows a categorization of the Google search operators.

allinanchor:

A query with allinanchor: restricts the results to the pages containing all the query terms in their inbound links; that is, all query words appear in anchor text of links to the page. The allinanchor: operator cannot be used in combination with any other search operators.

See the following example:

```
allinanchor: Longest river
```

In this example, Google will return the results that contain "longer" and "river" in the URL of the pages. Anchor text is the text on a page that is linked to another Web page or a different place on the current page. In Figure 1-14, anchor text is clearly identified in the link, as it is highlighted in bold.

Search Service	Search Operators
Web Search	allinanchor:, allintext:, allintitle:, allinurl:, cache:, define:, filetype:, id:, inanchor:, info:, intext:, intitle:, inurl:, phonebook:, related:, rphonebook:, site:, stocks:
Image Search	allintitle:, allinurl:, filetype:, inurl:, intitle:, site:
Groups	allintext:, allintitle:, author:, group:, insubject:, intext:, intitle:
Directory	allintext:, allintitle:, allinurl:, ext:, filetype:, intext:, intitle:, inurl:
News	allintext:, allintitle:, allinurl:, intext:, intitle:, inurl:, location:, source:
Froogle	allintext:, allintitle:, store:

Table 1-1 This table shows a categorization of the Google search operators

Figure 1-14 The allinanchor: query returns results that include the query terms in a site's inbound links.

allintext:

A query with allintext: does not check the URL or the title, but restricts the results to the pages containing all the query terms in the text.

See the following example:

```
allintext: Best travel
```

As seen in Figure 1-15, Google will return the results that contain "best" and "travel" in the text of the page.

allintitle:

A query with allintitle: restricts results to pages containing all the query terms specified in the title. Users should avoid the use of any other search operators while using allintitle.

The following example illustrates this:

```
allintitle: Vulnerability attacks
```

As seen in Figure 1-16, Google will return the results that contain "vulnerability" and "attacks" in the title, whereas in image search, allintitle returns images that contain the terms specified.

author:

A query with author: includes newsgroup articles by the author specified in the query. The author name can be a full name, partial name, or e-mail address. The following example will return articles that contain the word "hacking" written by "Linda Lee":

```
hacking author: Linda Lee
```

The results of this query will be a group of messages written by the author whose name is specified.

Figure 1-15 The allintext: query restricts results to pages with all the query terms on them.

Figure 1-16 The allintitle: query restricts results to pages with the query terms in the title.

cache:URL

The cache:URL query displays Google's cached version of a Web page. If the page was already accessed through the Google search, a user can directly perform the search and click cache on the link. But if the user wants to access the cache directly, the **cache:URL** syntax can be used. When using this query, a user must ensure there is no space left between cache: and the URL.

Consider the following example:

```
cache:www.eccouncil.org
```

As Figure 1-17 shows, this query reveals the cached version of *www.eccouncil.org*. This is the case when the user supplies a correct URL in conjunction with the cache: operator. But if the URL is wrong, Google will show unexpected results or the results matching the URL provided.

define:

The define: query is one of Google's advanced search operators. This query displays the definition of the term that is specified. It is useful for finding definitions of words, phrases, and acronyms.

The results of the following example are shown in Figure 1-18:

```
define: hacking
```

filetype:

The filetype: search operator helps the user search Web pages for a specific file type. It restricts results to files of a specific type. The file type is the extension of the file, and by default is part of the URL. Table 1-2 shows some common file extensions.

Figure 1-17 The **cache:URL** query can be used to reveal the cached versions of Web sites.

Figure 1-18 The define: query is useful for finding definitions of words, phrases, and acronyms.

File Type	File Extension
Adobe Acrobat Document	PDF
Microsoft Excel	XLS
Microsoft PowerPoint	PPT
Microsoft Word	DOC
Rich Text Format	RTF
Shockwave Flash	SWF
Text	ANS, TXT

Table 1-2 Common file extensions can be used with the filetype: query

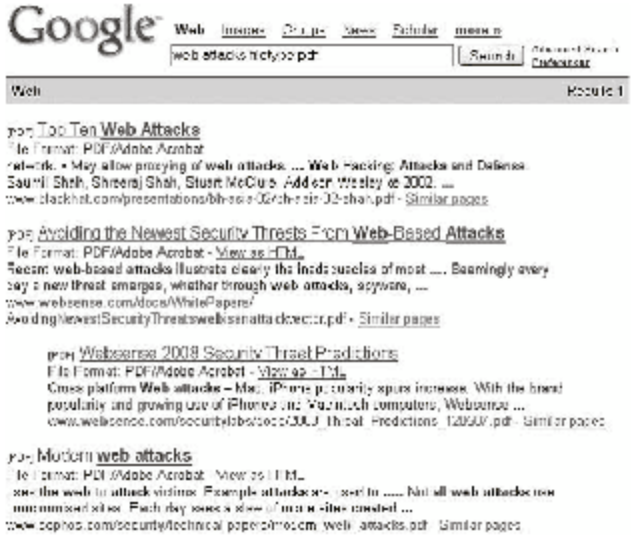


Figure 1-19 The filetype: query restricts a search to a specific file type.

The following example returns Adobe Acrobat PDF files that match the terms "Web" and "attacks," as shown in Figure 1-19:

```
Web attacks filetype: pdf
```

Table 1-2 shows common file extensions, but there are various other file extensions that are prevalent on the Web, as shown in Figure 1-10.

group:

group: is the Google search operator used for searching group titles. A query with this operator restricts results to newsgroup articles from certain groups or subareas. In a few cases, it not only searches for the term in the actual name of the group but also provides terms that describe the group.

The following example shows the use of this search operator, as shown in Figure 1-20:

```
sleep group: misc.kids
```

This search returns articles in the subarea "misc.kids" that contain the word "sleep."

Figure 1-20 The group: query restricts searches to specific newsgroup articles.

inanchor:

With the search operator inanchor:, the terms appear in anchor text of links to the page. It does not check the URL but checks the text in the link. It accepts the words or phrases as an argument, as shown in the following example:

```
restaurants inanchor:menu
```

As seen in Figure 1-21, this returns pages with anchor text on links to the pages containing the word "menu" and whose text contains the word "restaurants."

insubject:

The insubject: Google advanced operator is used for searching the subject lines of Google groups. A query with insubject: restricts articles in Google groups to pages containing the query terms specified. This operator is similar to the intitle: operator.

The following example shows the use of insubject:

```
insubject:"Security issue"
```

As seen in Figure 1-22, this search returns Google group articles that contain the phrase "Security issue" in the subject.

intext:

A query with intext: restricts results to documents containing the term in the text. Users must ensure that there is no space between intext: and the keyword.

Figure 1-21 The inanchor: query finds words that appear in the anchor text of links to a page.

Figure 1-22 The insubject: operator is used for searching the subject lines of Google groups.

Figure 1-23 The intext: query restricts results to documents containing the term after intext:.

The following example shows the use of this query, as shown in Figure 1-23:

```
intext:poem
```

link:

link: is the Google search operator used to search for links to a page. A query with **link:URL** shows pages that point to that URL. Users must ensure that there is no space between link: and the URL, or Google will come up with unexpected results. Here, either the URL or the server name can be used as an argument.

The following example shows the use of this query:

```
link:www.googleguide.com
```

As shown in Figure 1-24, this search shows links to the *googleguide.com* Web pages.

location:

The location: query shows Google news articles from the location specified. The following example illustrates this:

```
Hackers location: China
```

As seen in Figure 1-25, this search shows articles that match the term "hackers" from sites in China.

Figure 1-24 The link: query turns up any pages that are linked to the search term.

Figure 1-25 The location: query turns up Google news results from the specified location.

Other Useful Google Searches

Viewing Live Webcams

Live security cameras, traffic-monitoring cameras, and many other video devices can be found by using simple Google search operators such as inurl, intitle, and intext. These cameras generally use known protocols, which makes it easy for anyone to access them. The following Google search links can be used to find publicly accessible live streaming feeds:

- inurl:/view.shtml
- intitle:"Live View / - AXIS" | inurl:view/view.shtml
- inurl:ViewerFrame?Mode=
- inurl:ViewerFrame?Mode=Refresh
- inurl:axis-cgi/jpg
- allintitle: "Network Camera NetworkCamera"
- intitle:axis intitle:"video server"
- intitle:liveapplet inurl:LvAppl
- intitle:"EvoCam" inurl:"Webcam.html"
- intitle:"Live NetSnap Cam-server feed"
- intitle:"netcam live image"

intranet | help.desk

The term ***intranet***, despite more specific technical meanings, has become a generic term that describes a network confined to a small group. In most cases, the term *intranet* describes a closed or private network, unavailable to the general public. However, many sites have configured portals that allow access to an intranet from the Internet, bringing this typically closed network one step closer to potential attackers.

In rare cases, private intranets have been discovered on the public Internet due to a network device misconfiguration. In these cases, network administrators were completely unaware that their private networks were accessible to anyone via the Internet. Most often, an Internet-connected intranet is only partially accessible from the outside. In these cases, filters are employed that allow access to only certain pages from specific addresses, presumably inside a facility or campus. There are two major problems with this type of configuration. First, it is an administrative nightmare to keep track of the access rights of specific pages. Second, this is not true access control. This type of restriction can be bypassed very easily if an attacker gains access to a local proxy server, bounces a request off a local misconfigured Web server, or simply compromises a machine on the same network as trusted intranet users. Unfortunately, it is nearly impossible to provide a responsible example of this technique in action. Each example considered for this section was too easy for an attacker to reconstruct with a few simple Google queries.

Help desks are places to go for common problems that users encounter when using a Web site. Since the inception of help desks, hackers have been donning alternate personalities in an attempt to gain sensitive information from unsuspecting technicians. Recently, help desk procedures have started to address the hacker threat by insisting that technicians validate callers before attempting to assist them. Most help desk workers will (or should) ask for identifying information such as usernames, Social Security numbers, employee numbers, and even PINs to properly validate callers' identities. Some procedures are better than others, but for the most part, today's help desk technicians are at least aware of the potential threat that is posed by an imposter.

The **intranet | help.desk** query is designed not to bypass help desk procedures, but to locate pages describing help desk procedures. When this query is combined with a site search, the results could indicate the location of a help desk (Web page, telephone number, or the like), the information that might be requested by help desk technicians (which an attacker could gather before calling), and in many cases links that describe troubleshooting procedures.

Locating Public Exploit Sites

One way to locate exploit code is to focus on the file extension of the source code and then search for specific content within that code. Since source code is the text-based representation of difficult-to-read machine code, Google is well suited for this task. For example, a large number of exploits are written in C, which generally uses source code ending in a .c extension. A search for filetype:c returns nearly 500,000 results.

Site	Directory
packetstorm.linuxsecurity.com	packetstorm.linuxsecurity.com/0101-exploits/
synnergy.net	synnergy.net/downloads/exploits/
unsecure.altervista.org	unsecure.altervista.org/security/
www.blacksheepnetworks.com	www.blacksheepnetworks.com/security/hack/
www.circlemud.org	www.circlemud.org/pub/jelson/gethostbyname/
www.dsinet.org	www.dsinet.org/tools/Technotronic/
www.metasploit.com	www.metasploit.com/tools/
www.nostarch.com	www.nostarch.com/extras/hacking/chap2/
www.packetstormsecurity.org	www.packetstormsecurity.org/0409-exploits/
www.rosiello.org	www.rosiello.org/archivio/
www.safemode.org	www.safemode.org/files/zillion/exploits/
www.security-corporation.com	www.security-corporation.com/download/exploit/
www.thc.org	www.thc.org/exploits/

Figure 1-26 By using a filetype: query associated with source code, users can find exploits associated with specific code.

A query for **filetype:c exploit** returns around 5,000 results, most of which are exactly the types of programs hackers are looking for. These are the most popular sites hosting C source code containing the word *exploit*; the returned list is a good start for a list of bookmarks. Using page-scraping techniques, a user can isolate these sites by running a UNIX command such as:

```
grep Cached exp | awk -F" -" '{print $1}' | sort -u
```

against the dumped Google results page. Using the cut-and-paste function or a command such as **lynx–dump** works well for capturing the page this way. The slightly polished results of finding 20 results from Google in this way are shown in Figure 1-26.

Locating Exploits via Common Code Strings

Another way to locate exploit code is to focus on common strings within the source code itself. One way to do this is to focus on common inclusions or header file references. For example, many C programs include the standard input/output library functions, which are referenced by an include statement such as:

```
#include <stdio.h>
```

A query such as **"#include <stdio.h>" exploit** would locate the C source code that contains the word *exploit*, regardless of the file's extension. This would catch code (and code fragments) that is displayed in HTML documents. Extending the search to include programs that include a friendly usage statement with a query such as **"#include <stdio.h>" usage exploit** returns the results shown in Figure 1-27.

This search returns quite a few hits, nearly all of which contain exploit code. Using traversal techniques (or simply hitting up the main page of the site) can reveal other exploits or tools. Notice that most of these hits are HTML documents, which the previous filetype:c query would have excluded. There are many ways to locate source code using common code strings, but not all source code can fit into a nice, neat little box. Some code can be nailed down fairly neatly using this technique; other code might require a bit more query tweaking.

Locating Vulnerable Targets

Attackers are increasingly using Google to locate Web-based targets vulnerable to specific exploits. In fact, it is not uncommon for public vulnerability announcements to contain Google links to potentially vulnerable targets, as shown in Figure 1-28.

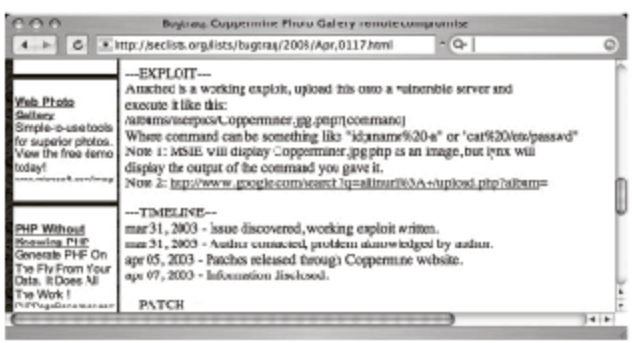


Figure 1-28 Some vulnerability announcements contain actual links to potentially vulnerable targets.

Locating Targets via Demonstration Pages

The user's goal is to develop a query string to locate vulnerable targets on the Web; the vendor's Web site is a good place to discover what exactly the product's Web pages look like. Like many software vendors' Web sites, the CubeCart site shows links for product demonstrations and live sites that are running the product, as shown in Figure 1-29.

At the time of this writing, this site's demonstration pages were offline, but the list of live sites was active. Live sites are often better for this purpose because they allow the user to account for potential variations in

Figure 1-29 Demonstration pages can be used to find information about a product's Web page.

Figure 1-30 Most live sites have a "powered by" message followed by a product version.

how a Web site is ultimately displayed. For example, some administrators might modify the format of a vendor-supplied Web page to fit the theme of the site. These types of modifications can impact the effectiveness of a Google search that targets a vendor-supplied page format.

Nearly every site has a "powered by" message at the bottom of the main page, as shown in Figure 1-30.

In this case, the live page displays "Powered by CubeCart 2.0.1" as a footer on the main page. Since CubeCart 2.0.1 is the version listed as vulnerable in the security advisory, it is necessary to do little else to create

Figure 1-31 Narrowing down the search to a specific version of a product makes it easier to find exploits.

a query that locates vulnerable targets on the Web. The final query, "Powered by CubeCart 2.0.1," returns results of over 27,000 potentially vulnerable targets, as shown in Figure 1-31.

By combining this list of sites with the exploit tool released in the Secunia security advisory, an attacker has access to a virtual smorgasbord of online retailers that could likely be compromised, potentially revealing sensitive customer information such as address, products purchased, and payment details.

Locating Targets via Source Code

Although this method is more drawn out (and could be short-circuited by creative thinking), it shows a typical process for detecting an exact working query for locating vulnerable targets. A hacker might use the source code of a program to discover ways to search for that software with Google. For example, an advisory was released for the CuteNews program, as shown in Figure 1-32.

To find the best search string to locate potentially vulnerable targets, users can visit the Web page of the software vendor to find the source code of the offending software. In cases where source code is not available, an attacker might opt to simply download the offending software and run it on a personal computer to get ideas for potential searches.

Figure 1-33 lists examples of some queries designed to locate targets running potentially vulnerable Web applications. These examples were all pulled from the Google Hacking Database (GHDB).

Locating Targets via CGI Scanning

One of the oldest and most familiar techniques for locating vulnerable Web servers is through the use of a CGI scanner. These programs parse a list of known "bad" or vulnerable Web files and attempt to locate those files on a Web server. Based on various response codes, the scanner could detect the presence of these potentially vulnerable files. A CGI scanner can list vulnerable files and directories in a data file, such as the ones shown in Figure 1-34.

A Single CGI Scan–Style Query

Instead of connecting directly to a target server, an attacker could use Google to locate servers that might be hosting vulnerable files and directories by converting each line into a Google query. For example, the first line searches for the filename userreg.cgi located in a directory called cgi-bin. Converting this to a Google query is fairly simple in this case, as a search for **inurl:/cgi-bin/userreg.cgi** shows in Figure 1-35.

Figure 1-32 A hacker might start a search for vulnerable targets by searching for security advisories related to vulnerable programs.

Query	Vulnerability
"Powered by A-CART"	A-CART 2.x vulnerable to cross-site scripting
inurl:"dispact.php/atarodetype" | inurl:"class.attachattribute.6.php	Achievo 0.8.x could allow remote code execution
intitle:guestbook "advanced guestbook 2.2 powered"	Advanced Guestbook v2.2 has an SQL injection problem that allows unauthorized access
"Powered by AJ-Fork v167"	AJ-fork, a fork based on the CuteNews 1.3.1 core, is susceptible to multiple vulnerabilities
"Blackboard 1.5.1-f | ¬© 2003-4 by Yves Goergen"	Blackboard 1.5.1 has a remote file inclusion vulnerability
"BusDates Calendar System " "powered by BusDates v3.2 by BusDev"	BusDates 3.2 is vulnerable to SQL injection
inurl:changepassword.cgi -cvs	changepassword.cgi allows for unlimited repeated failed login attempts
"Copyright ¬© 2005 Agustin Dondo Scripts"	ChelPHP 1.0 has multiple vulnerabilities
"Powered by CuteCart 2.0.1"	CuteCart 2.0.1 has an SQL injection vulnerability
"Powered by newsBlogonce" ("dasBlog 1.6"| "dasBlog 1.5"| "dasBlog 1.4"| "dasBlog 1.3")	DasBlog versions 1.3–1.6 are susceptible to an HTML injection vulnerability in their request log
"Powered by DCP-Portal v5.5"	DCP-Portal version 5.5 is vulnerable to SQL injection
"2005 DUNWAY All Rights Reserved"	DUForum 3.0 may allow a remote attacker to carry out SQL injection and HTML injection attacks
"inurl:/store/index.asp?checkout="	Dwc_Articles 1.6 has multiple input validation problems
inurl:cutmaa.asp	EarlyImpact ProductCart v1.5 contains multiple vulnerabilities

Figure 1-33 Queries can be designed to locate targets running vulnerable Web applications.

```
/cgi-bin/userreg.cgi
/cgi-bin/cgiemail/uargg.txt
/random_banner/index.cgi
/random_banner/index.cgi
/cgi-bin/mailview.cgi
/cgi-bin/maillist.cgi
/iissamples/iissamples/SQLQuit.asp
/iissamples/exair/search/query.htw
/siteserver/admin/findvserver.asp
/scripts/cphost.dll
/cgi-bin/finger.cgi
```

Figure 1-34 CGI scanning can bring up lists of vulnerable files and directories in a data file.

Figure 1-35 Google can be used to locate servers that might be hosting vulnerable files and directories by converting each line into a Google query.

This search locates more than 60 hosts that are running the supposedly vulnerable program. However, there is no guarantee that the program Google detected is the vulnerable program. This highlights one of the biggest problems with CGI scanner programs. The mere existence of a file or directory does not necessarily indicate that the vulnerability is present. Still, there is no shortage of these types of scanner programs on the Web, each of which provides the potential for many different Google queries.

There are other ways to go after CGI-type files. For example, the filetype operator can be used to find the actual CGI program, even outside the context of the parent cgi-bin directory, with a query such as **filetype:cgi inurl:userreg.cgi**. This locates approximately 15 more results. Unfortunately, this search is even less reliable, because the cgi-bin directory is an indicator that the program is in fact a CGI program. Depending on the configuration of the server, the userreg.cgi program might be a text file, not an executable, making exploitation of the program difficult, if not altogether impossible.

Another way of finding this file is via a directory listing with a query such as **intitle:index.of userreg.cgi**. This query returns no hits at the time of this writing, and for good reason. Directory listings are not nearly as common as URLs on the Web, and a directory listing containing a file this specific is rare.

Directory Listings

The server tag at the bottom of a directory listing can provide explicit detail about the type of Web server software that is running. If an attacker has an exploit for Apache 2.0.52 running on a UNIX server, a query such as **server.at "Apache/2.0.52"** will locate servers that host a directory listing with an Apache 2.0.52 server tag, as shown in Figure 1-36.

Not all Web servers place this tag at the bottom of directory listings, but most Apache derivatives turn on this feature by default.

Finding IIS 5.0 Servers

Other platforms, such as Microsoft's IIS, display server tags as well, as a query for **"Microsoft-IIS/5.0 server at"** shows in Figure 1-37.

When searching for these directory tags, syntax is very important. There are many irrelevant results from a query for **"Microsoft-IIS/6.0" "server at"**, whereas a query like **"Microsoft-IIS/6.0 server at"** provides very relevant results.

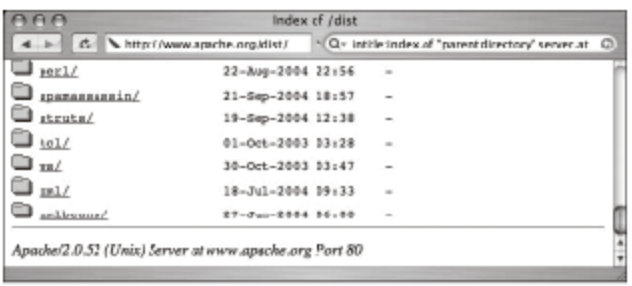


Figure 1-36 Directory listings often include server tags that reveal the Web server software that is running the site.

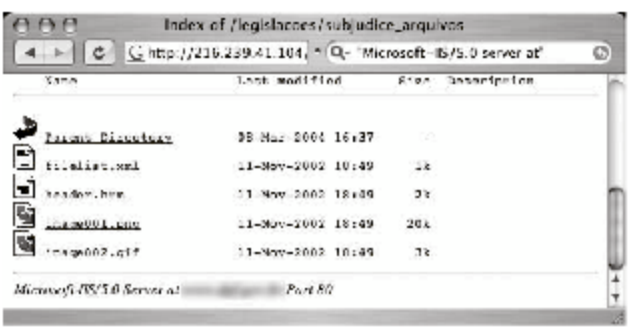


Figure 1-37 Microsoft server software is sometimes revealed by a tag in a directory listing.

Web Server Software Error Messages

Error messages contain a lot of useful information, but in the context of locating specific servers, users can use portions of various error messages to locate servers running specific software versions.

The absolute best way to find error messages is to figure out what messages the server is capable of generating. Users could gather these messages by examining the server source code or configuration files, or by actually generating the errors on the server. The best way to get this information from IIS is by examining the source code of the error pages themselves.

IIS 5 and 6, by default, display static HTTP/1.1 error messages when the server encounters some sort of problem. These error pages are stored by default in the %SYSTEMROOT%\help\iisHelp\common directory. These files are essentially HTML files named by the type of error they produce, such as 400.htm, 401-1.htm, 501.htm, and so on. For example, the file that produces 400 error pages, 400.htm, contains a line similar to the following one:

```
<title>The page cannot be found</title>
```

This is a dead giveaway for an effective intitle query such as intitle: "The page cannot be found." Unfortunately, this search yields far too many results. The searcher must dig deeper into the 400.htm file to get more clues about what to look for. Lines 65–88 of 400.htm are shown here:

```
65. <p>Please try the following:</p>
66. <ul>
67. <li>If you typed the page address in the Address bar, make sure that it is
spelled correctly.</li>
68.
69. <li>Open the
70.
71. <script language="JavaScript">
72. <!--
73. if (!((window.navigator.userAgent.indexOf("MSIE") > 0) && (window.navigator.ap-
pVersion.charAt(0) == "2")))
74. {
75. Homepage();
76. }
77. -->
78. </script>
79.
80. home page, and then look for links to the information you want.</li>
81.
82. <li>Click the
83. <a href="javascript:history.back(1)">
84. Back</a> button to try another link.</li>
85. </ul>
86.
87. <h2 style="COLOR:000000; FONT: 8pt/11pt verdana">HTTP 400 - Bad
Request<br>
88. Internet Information Services</h2>
```

The phrase "Please try the following" in line 65 exists in every single error file in this directory, making it a perfect candidate for part of a good base search. This line could effectively be reduced to "Please * * following." Line 88 shows another phrase that appears in every error document: "Internet Information Services." These are "golden terms" to use to search for IIS HTTP/1.1 error pages that Google has crawled. A query such as **intitle: "The page cannot be found" "Please * * following" "Internet * Services"** can be used to search for IIS servers that present a 400 error page, as shown in Figure 1-38.

Figure 1-38 Common error messages can be used as effective queries.

400	The page cannot be found
401.1, 401.2, 401.3, 401.4, 401.5	You are not authorized to view this page
403.1, 403.2	The page cannot be displayed
403.3	The page cannot be saved
403.4	The page must be viewed over a secure channel

Figure 1-39 IIS default installations have unique error codes.

On this cached page, the actual error code is printed on the page, about halfway down. This error line is also printed on each of IIS's error pages, making it another good limiter for a search. The line on the page begins with "HTTP Error 404," which might seem out of place, considering the user was searching for a 400 error code, not a 404 error code. This occurs because several IIS error pages produce similar pages. Although commonalities are often good for Google searching, they could lead to some confusion and produce ineffective results if the user is searching for a specific, less-benign error page.

It is necessary to sort out exactly what's what in these error page files. Figure 1-39 lists all the unique HTML error page titles and error codes from a default IIS 5 installation.

These page titles, used in an intitle search, combined with the other golden IIS error searches make for very effective searches, locating all sorts of IIS servers that generate all sorts of error pages. To troll for IIS servers with the esoteric 404.1 error page, a user can try a query such as **intitle: "The Web site cannot be found" "Please ? ? following."** A more common error can be found with a query such as **intitle: "The page cannot be displayed" "Internet Information Services" "Please ? ? following."** This query is very effective because this error page is shown for many different error codes.

"Object Not Found" Error Message Used to Find IIS 5.0

In addition to displaying the default static HTTP/1.1 error pages, IIS can be configured to display custom error messages, configured via the Management Console. An example of this type of custom error page is shown in Figure 1-40. This type of functionality makes the job of the Google hacker a bit more difficult, since there is no apparent way to hone in on a customized error page. However, some error messages, including 400, 403.9, 411, 414, 500, 500.11, 500.14, 500.15, 501, 503, and 505 pages (which are explained in Table 1-3), cannot be customized. In terms of Google hacking, this means that there is no easy way an IIS server can prevent displaying

the static HTTP/1.1 error pages. This opens the door for locating these servers through Google, even if the server has been configured to display custom error pages.

Besides trolling through the IIS error pages looking for exact phrases, a user can also perform more generic queries, such as **intitle: "the page cannot be found" inetmgr**, which focuses on the fairly unique term used to describe the IIS Management Console, inetmgr. Other ways to perform this same search might be **intitle: "the page cannot be found" "internet information services"** or **intitle: "Under construction" "Internet Information Services"**.

Other, more specific searches can reveal the exact version of the IIS server, such as a query for **intext:"404 Object Not Found" Microsoft-IIS/5.0**, as shown in Figure 1-40.

Apache Web Server Error Messages

Apache Web servers can also be located by focusing on server-generated error messages. Some generic searches, such as **"Apache/1.3.27 Server at" –intitle:index.of intitle:inf"** or **"Apache/1.3.27 Server at" –intitle:index.of intitle:error"**, can be used to locate servers that might be advertising their server version via an information or error message, as shown in Figure 1-41.

A query such as **"Apache/2.0.40" intitle: "Object not found!"** will locate Apache 2.0.40 Web servers that present this error message.

Code	Description
400 Bad Request	The request contains bad syntax or cannot be fulfilled.
403 Forbidden	The request was a legal request, but the server is refusing to respond to it.
411 Length Required	The request did not specify the length of its content, which is required by the requested resource.
414 Request-URI Too Long	The URI provided was too long for the server to process.
500 Internal Server Error	A generic error message, given when no more specific message is suitable
500.11	Application is shutting down on the Web server.
500.14	Invalid application configuration on the server.
500.15	Direct requests for Global.asa are not allowed.
501 Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfill the request.
503 Service Unavailable	The server is currently unavailable (because it is overloaded or down for maintenance).
505 HTTP Version Not Supported	The server does not support the version of HTTP presented.

Table 1-3 These are some IIS error messages that cannot be customized

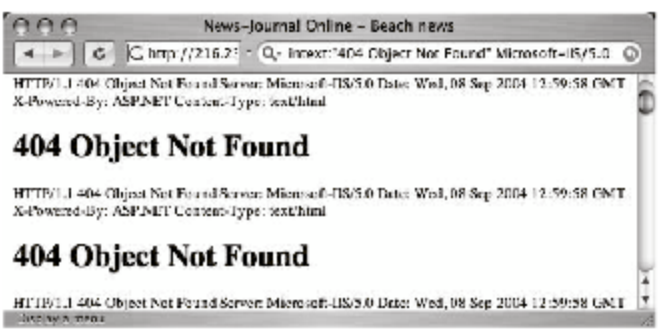


Figure 1-40 Specific searches can reveal the exact version of the IIS server.

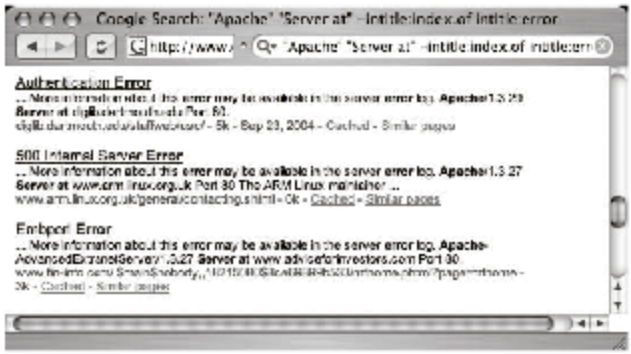


Figure 1-41 An Apache Web server can be located by focusing on server-generated error messages.

Most Apache installations rely on a configuration file called httpd.conf. Searching through Apache 2.0.40's httpd.conf file reveals the location of the HTML templates for error messages. The referenced files, seen in Figure 1-42, are located in the Web root directory—such as /error/http_BAD_REQUEST.html.var, which refers to the /var/www/error directory on the file system.

By looking at one of these template files, a user can see recognizable HTML code and variable listings that show the construction of an error page.

Apache 2.0 Error Pages

Using some basic shell commands, a user can isolate both the title of an error page and the text that might appear on the error page:

```
grep -h -r "Content-language: en" -A 10 | grep -A5 "TITLE" | grep –v virtual
```

This Linux bash shell command, when run against the Apache 2.0 source code tree, will produce output similar to that shown in Figure 1-43. This figure lists the title of each English Apache (2.0 and newer) error page as well as a portion of the text that will be located on the page. Instead of searching for English messages only, a user could search for errors in other Apache-supported languages by simply replacing the content-language string "en" in the previous grep command to, for example, de, es, fr, or sv, for German, Spanish, French, or Swedish, respectively.

To use this table, simply supply the text in the Error Page Title column as an intitle search and a portion of the text column as an additional phrase in the search query. Since some of the text is lengthy, users might need to select a unique portion of the text or replace common words with an asterisk, which will reduce the search query to the 10-word limit imposed on Google queries.

Application Software Error Messages

In many cases, applications running on the Web server can generate errors that reveal information about the server as well. There are untold thousands of Web applications on the Internet, each of which can generate any number of error messages. Dedicated Web assessment tools such as WebInspect excel at performing detailed Web application assessments, making it seem a bit pointless to troll Google for application error messages.

One query, **"Fatal error: Call to undefined function" –reply –the –next**, will locate Active Server Page (ASP) error messages. These messages often reveal information about the database software in use on the server as well as information about the application that caused the error, as shown in Figure 1-44.

Although this ASP message is fairly benign, some ASP error messages are much more revealing. Consider the query **"ASP.NET_SessionId" "data source="**, which locates unique strings found in ASP.NET application-state dumps, as shown in Figure 1-45. These dumps reveal all sorts of information about the running application

```
ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
ErrorDocument 410 /error/HTTP_GONE.html.var
ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var
```

Figure 1-42 An httpd.conf file reveals the location of the HTML templates for error messages.

Error Page Title	Error Page Partial Text
Bad gateway!	The proxy server received an invalid response from an upstream server.
Bad request!	Your browser (or proxy) sent a request that this server could not understand.
Access forbidden!	You do not have permission to access the requested directory. Either there is no index document or the directory is read-protected.
Resource is no longer available!	The requested URL is no longer available on this server and there is no forwarding address.
Server error!	The server encountered an internal error and was unable to complete your request.
Method not allowed!	A request with the method is not allowed for the requested URL.
No acceptable object found	An appropriate representation of the requested resource could not be found on this server.
Object not found!	The requested URL was not found on this server.
Client request aspect!	The server does not support the action requested by the browser.
Precondition failed	The precondition on the request for the URL failed positive evaluation.
Request entity too large!	The method does not allow the data transmitted, or the data volume exceeds the capacity limit.
Request timeout!	The server closed the network connection because the browser didn't finish the request within the specified time.
Submitted URI too large!	The length of the requested URL exceeds the capacity limit for this server. The request cannot be processed.
Service unavailable	The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.
Authentication required!	This server could not verify that you are authorized to access the URL. You either supplied the wrong credentials (e.g., a bad password), or your browser doesn't understand how to supply the credentials required.
Unsupported media type	The server does not support the media type transmitted in the request.
Variant also varies!	A variant for the requested entity is itself a negotiable resource. Access not possible.

Figure 1-43 A Linux bash shell command, when run against the Apache 2.0 source code tree, will reveal a list of error messages with their meanings.

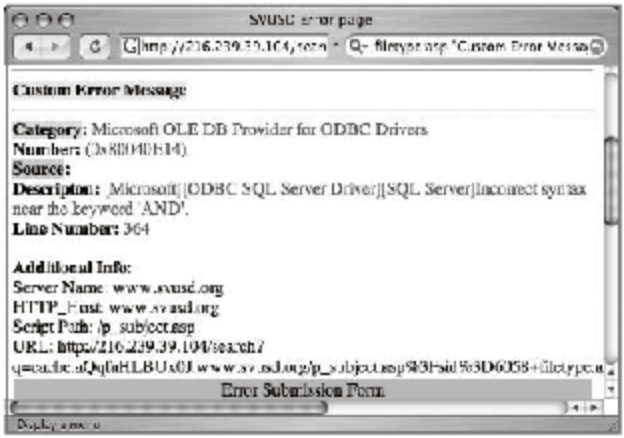


Figure 1-44 Application software error messages often reveal information about the database software in use on the server as well as information about the application that caused the error.

and the Web server that hosts that application. An advanced attacker could use encrypted password data and variable information in these stack traces to subvert the security of the application and perhaps the Web server itself.

PHP application errors are fairly commonplace. They can reveal all sorts of information that an attacker can use to profile a server. One very common error can be found with a query such as **intext: "Warning: Failed opening" include_path**, as shown in Figure 1-46.

CGI programs often reveal information about the Web server and its applications in the form of environment variable dumps. A typical environmental variable output page is shown in Figure 1-47.

This screen shows information about the Web server and the client that connected to the page when the data were produced. Since Google's bot crawls, one way to find these CGI environment pages is to focus on the trail left by the bot, reflected in these pages as the "HTTP_FROM=googlebot" line. A user can search for pages like this with a query such as **"HTTP_FROM=googlebot" googlebot.com "Server_Software"**. These pages are dynamically generated, which means that the user must look at Google's cache to see the document as it was crawled. To locate good base searches for a particular application, it is best to look at the source code of that application.

Default Pages

Another way to locate specific types of servers or Web software is to search for default Web pages, as shown in Figure 1-48. Most Web software, including the Web server software itself, ships with one or more ***default pages***. These pages can make it easy for a site administrator to test the installation of a Web server or application. By providing a simple page to test, the administrator can simply connect to the Web server with a browser to validate that the Web software was installed correctly.

Some operating systems even come with Web server software already installed. In this case, the owner of the machine might not even realize that a Web server is running on the machine. This type of casual behavior on the part of the owner will lead an attacker to rightly assume that the Web software is not well maintained and is, by extension, insecure. By further extension, the attacker can also assume that the entire operating system of the server might be vulnerable by virtue of poor maintenance. In some cases, Google crawls a Web server while it is in its earliest stages of installation, still displaying a set of default pages. In these cases, there is generally a

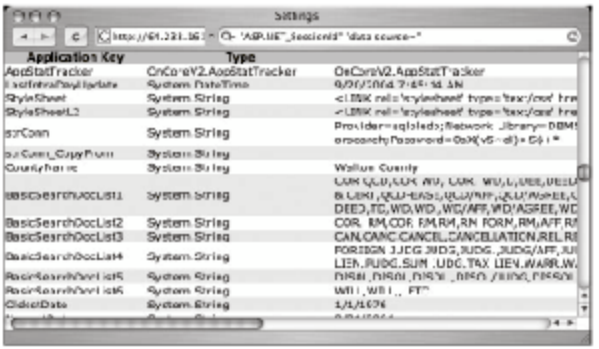


Figure 1-45 ASP dumps provide valuable details.

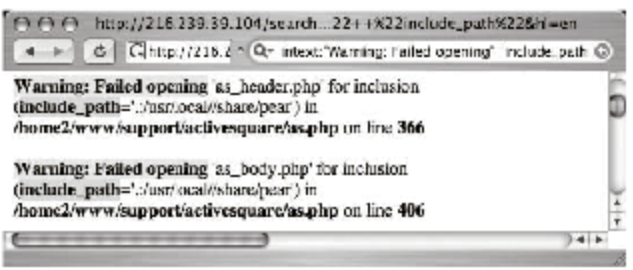


Figure 1-46 PHP errors reveal pathnames and filenames.

Figure 1-47 CGI environment listings can reveal valuable information.

Figure 1-48 Default Web pages for servers can indicate a poorly maintained system.

Apache Server Version	Query
Apache 1.2.6	*intitle:"Test Page for Apache Installation" "You are free"*
Apache 1.3.0–1.3.9	*intitle:"Test Page for Apache" "It worked!" "this Web site!"*
Apache 1.3.11–1.3.31	*intitle:Test.Page.for.Apache seeing.this.instead*
Apache 2.0	*intitle:Simple.page.for.Apache Apache.Hook.functions*
Apache SSL/TLS	*intitle:test.page "Hey, it worked !" "SSL/TLS-aware"*
Apache on Red Hat	*"Test Page for the Apache Web Server on Red Hat Linux"*
Apache on Fedora	*intitle:"test page for the apache http server on fedora core"*
Apache on Debian	*intitle:"Welcome to Your New Home Page!" debian*
Apache on other Linux	*intitle:"Test Page Apache Web Server on " -redhat -fedora*

Figure 1-49 These queries can be used to locate specific families of Apache running default pages.

short window of time between the moment when Google crawls the site and when the intended content is actually placed on the server. This means that there could be a disparity between what the live page is displaying and what Google's cache displays. This makes little difference from a Google hacker's perspective, since even the past existence of a default page is enough for profiling purposes.

Notice that the administrator's e-mail address is generic as well, indicating that attention was not paid to detail during the installation of this server. These default pages do not list the version number of the server, which is an important piece of information for a successful attack. It is possible, however, that an attacker could search for specific variations in these default pages to find specific ranges of server versions. Figure 1-49 shows queries that can be used to locate specific families of Apache running default pages.

IIS also displays a default Web page when first installed. A query such as **intitle:"Welcome to IIS 4.0"** can locate very specific versions of IIS, as shown in Figure 1-50.

The queries in Figure 1-51 locate specific IIS server versions.

Other types of Web servers can be located by querying for default pages as well. Figure 1-52 lists a sample of lesser-known Web servers that can be profiled with this technique.

Default Login Portals

Login portals are designed to allow access to specific features or functions after a user logs in. Google hackers search for login portals as a way to profile the software that is in use on a target and to locate links and documentation that might provide useful information for an attack. In addition, if an attacker has an exploit for a particular piece of software, and that software provides a login portal, the attacker can use Google queries to locate potential targets.

Some login portals, like the one shown in Figure 1-53, captured with **allinurl: "exchange/logon.asp"**, are obviously default pages provided by the software manufacturer—in this case, Microsoft. Just as an attacker can get an idea of the potential security of a target by simply looking for default pages, a default login portal can

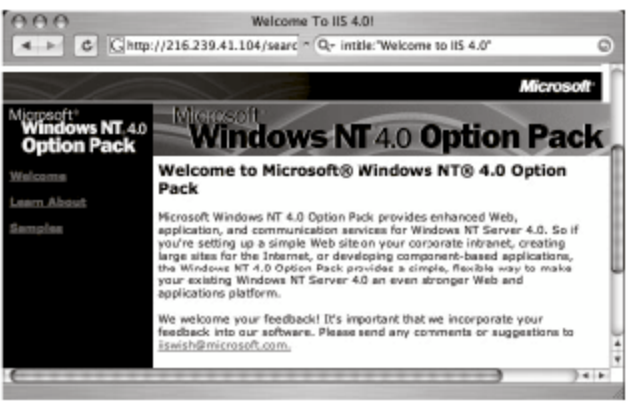


Figure 1-50 An IIS default Web page can be located with a query such as **intitle: "Welcome to IIS 4.0".**

IIS Server Version	Query
Many	*intitle:"welcome to" intitle:internet IIS*
Unknown	*intitle:"Under construction:" "does not currently have"*
IIS 4.0	*intitle:"welcome to IIS 4.0"*
IIS 4.0	*allintitle:Welcome to Windows NT 4.0 Option Pack*
IIS 4.0	*allintitle:Welcome to Internet Information Server*
IIS 5.0	*allintitle:Welcome to Windows 2000 Internet Services*
IIS 6.0	*allintitle:Welcome to Windows XP Server Internet Services*

Figure 1-51 Specific IIS server versions can be located with queries matching their default pages.

Server/Version	Query
Cisco Micro Webserver 200	*"micro webserver home page"*
Generic Appliance	*"default web page" congratulations "hosting appliance"*
HP appliance sa1	*intitle:"default domain page" "congratulations" "hp web"*
iPlanet/Many	*intitle:"web server, enterprise edition"*
Intel Netstructure	*"congratulations on choosing" Intel netstructure*
JWS/1.0.3–2.0	*allintitle:default home page java web server*
J2EE/Many	*intitle:"default J2ee home page"*
Jigsaw/2.2.3	*intitle:"jigsaw overview" "this is your"*
Jigsaw/Many	*intitle:"jigsaw overview"*
KFSensor honeypot	*"KF Web Server Home Page"*
Kwiki	*"Congratulations! You've created a new Kwiki website."*
Matrix Appliance	*"Welcome to your domain web page" matrix*
NetWare 6	*intitle:"welcome to netware 6"*
Resin/Many	*allintitle:Resin Default Home Page*
Resin/Enterprise	*allintitle:Resin-Enterprise Default Home Page*
Samba Server	*intitle:"samba server" "1997...2004 Samba"*
Sun AnswerBook Server	*inurl:"/Answerbook2/options"*
TivoConnect Server	*inurl:/TiVoConnect*

Figure 1-52 Lesser-known servers also use default Web pages.

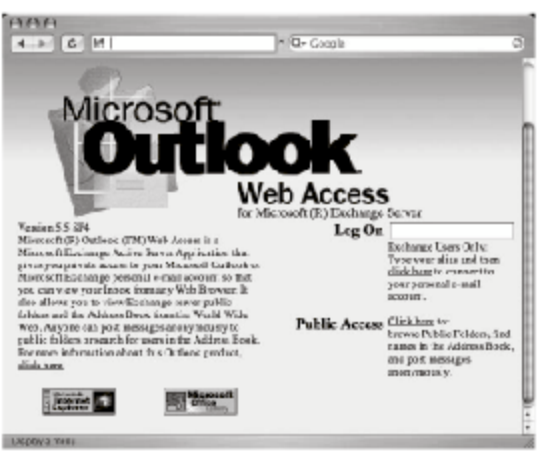


Figure 1-53 Default login portals can reveal information such as the version of the server being used.

indicate that the technical skill of the server's administrators is generally low, revealing that the security of the site will most likely be poor as well. Default login portals can even indicate the software revision of the program. An attacker can use this information to search for known vulnerabilities in that software version.

By following links from the login portal, an attacker can often gain access to other information about the target. The Outlook Web Access portal is particularly renowned for this type of information leak because it provides an anonymous public access area that can be viewed without logging in to the mail system. This public access area sometimes provides access to a public directory or to broadcast e-mails that can be used to gather usernames or information.

Searching for Passwords

Password data, one of the Holy Grails during a penetration test, should be protected. Unfortunately, many examples of Google queries can be used to locate passwords on the Web, as shown in Figure 1-54.

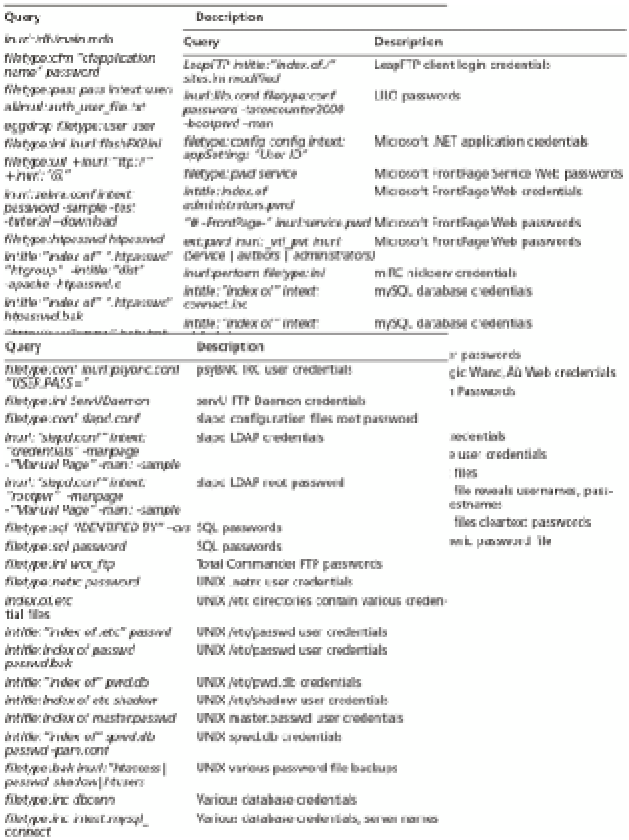


Figure 1-54 Specific queries can be used to search for passwords.

In most cases, passwords discovered on the Web are either encrypted or encoded in some way. In most cases, these passwords can be fed into a password cracker such as John the Ripper to produce plaintext passwords that can be used in an attack. Figure 1-55 shows the results of the search **ext:pwd inurl:_vti_pvt inurl:(Service | authors | administrators)**, which combines a search for some common Microsoft FrontPage support files.

Windows Registry Entries Can Reveal Passwords

Exported Windows registry files often contain encrypted or encoded passwords as well. If a user exports the Windows registry to a file and Google subsequently crawls that file, a query like **filetype: reg intext: "internet account manager"** could reveal interesting keys containing password data, as shown in Figure 1-56.

Figure 1-55 The search **ext:pwd inurl:_vti_pvt inurl:(Service | authors | administrators)** combines a search for some common Microsoft FrontPage support files.

Figure 1-56 A Windows registry file can be exported and searched to find encrypted passwords.

Note that live, exported Windows registry files are not very common, but it is not uncommon for an attacker to target a site simply because of one exceptionally insecure file. It is also possible for a Google query to uncover **plaintext** passwords. These passwords can be used as is, without having to employ a password cracking utility. In these extreme cases, the only challenge is determining the username as well as the host on which the password can be used.

Usernames, Plaintext Passwords, and Hostnames

Another generic search for password information, **intext: (password | passcode | pass) intext: (username | userid | user)**, combines common words for passwords and user IDs into one query. This query returns a lot of results, but the vast majority of the top hits refer to pages that list forgotten password information, including either links or contact information.

As shown in Figure 1-57, certain queries will locate all the following information: usernames, plaintext passwords, and the host that uses that authentication.

GoolagScan

GoolagScan (Figure 1-58) is software published by a famous hacker group called Cult of the Dead Cow (CDC). This software is used to convert the Google search engine into a vulnerability scanner. GoolagScan scans Web sites or Internet domains for vulnerabilities. It works on the dork pattern, which is used by the Google search

name: = "momo"; password: = "momo"; URL: = "password.htm" ...
name: = "momo"; password: = "momo"; URL: = "password.htm"; END_FILE
net/password.log - 1k - Supplemental Result - Cached - Similar pages

name: = "jbhunt"; password: = "jbhunt"; URL: = "http://home.nc.rr. ...
name: = "jbhunt"; password: = "jbhunt"; URL: = "http: /clay123/ref23.
html"; Beth Haas name: = "BHaas"; password: = "Beth Haas"; URL: = "http ...
.com/clay123/password.log - 2k - Supplemental Result - Cached - Similar pages

name: = "dv21"; password: = "dv21_2004"; URL: = "intern.htm"; name ... -
Translate this page
name: = "dv21"; password: = "dv21_2004"; URL: = "intern.htm"; name: = "dv22"; password: =
"dv22_2004"; URL: = "intern.htm"; name: = "dv23"; password ...
.de/grossmann/password.log - 1k - Cached - Similar pages

Figure 1-57 Certain queries will locate all the following information: usernames, plaintext passwords, and the host that uses that authentication.

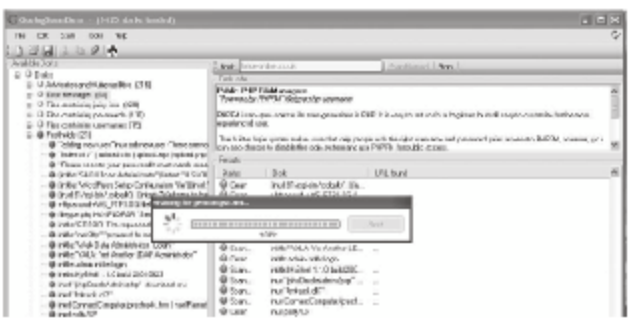


Figure 1-58 GoolagScan uses Google to find vulnerabilities.

engine. The search results obtained from the scan are used to exploit possible security vulnerabilities. Goolag is a standalone Windows GUI–based application and uses one XML-based configuration file for its settings.

Goolag includes the following features:

- Simplifies the use of myriad numbers of dorks to a few mouse clicks
- No need for cryptic command-line options knowledge
- Not mandatory to have knowledge about Google hacking to test a host
- Uses simple and readable XML documents

Chapter Summary

- The site query is invaluable during the information-gathering phase of an assessment.
- Error messages can reveal a great deal of information about a target.
- Even though a username is the less important half of most authentication mechanisms, it should at least be marginally protected from outsiders.
- The word *password* is so common on the Internet that there are over 73 million results for this one-word query. Launching a query for derivations of this word makes little sense unless the user actually combines that search with a site query.
- A query with author: includes newsgroup articles by the author specified in the query.
- The filetype: search operator helps the user search Web pages for a specific file type.
- Live security cameras, traffic-monitoring cameras, and many other video devices can be found by using simple Google search operators such as inurl, intitle, and intext.
- In rare cases, private intranets have been discovered on the public Internet due to a network device misconfiguration.
- One way to locate exploit code is to focus on the file extension of the source code and then search for specific content within that code.
- It is not uncommon for public vulnerability announcements to contain Google links to potentially vulnerable targets.
- One of the oldest and most familiar techniques for locating vulnerable Web servers is through the use of a CGI scanner.
- The server tag at the bottom of a directory listing can provide explicit detail about the type of Web server software that is running.

Hands-On Projects

1. Perform the following steps:
   - Navigate to Chapter 1 of the Student Resource Center.
   - Open Advanced Google and read the content.
2. Perform the following steps:
   - Navigate to Chapter 1 of the Student Resource Center.
   - Open Internet_for_Research.pdf and read the content.
3. Perform the following steps:
   - Navigate to Chapter 1 of the Student Resource Center.
   - Open Advanced Googling for Senior Executives and read the content.

Chapter 2

Routers and Switches Penetration Testing

Objectives

After completing this chapter, you should be able to:

- Identify a router hostname
- Port scan a router
- Identify router protocols
- Test for router misconfigurations
- Test for IP spoofing
- Access a router with a Web browser
- Test the address cache size of a switch
- Test a switch for frame loss
- Test a switch for latency

Key Terms

ARP (Address Resolution Protocol) protocol that maps IP addresses to hardware addresses used by the data-link layer

Cisco Discovery Protocol (CDP) a self-governing protocol supported by every Cisco networking device

Denial-of-service attack an attack on a network or computer system that creates a disturbance in the services provided by bandwidth consumption or overloading the system resources

Finger command used to detect the users present on a particular system

Hop the traversing of a packet from one router to another as it travels from its source to its destination

IP spoofing a method in which an attacker accesses a computer by assuming an authenticated identity

Network Time Protocol (NTP) an Internet protocol used for synchronizing time across computer networks

Open Shortest Path First (OSPF) protocol a link-state-based routing protocol that measures several metrics to determine the cost, or efficiency, of the route

Port scanning scanning that determines the ports that are open and the services that are active

Router a device that forwards data packets by determining the destination network point on the Internet

Routing Information Protocol (RIP) a distance-vector routing protocol that determines the route solely based on the distance, or number of hops, involved

Source routing a method in which the sender specifies a path that a packet should take in order to travel in a network

Introduction to Routers and Switches Penetration Testing

A *router* is a device that forwards data packets by determining the destination network point on the Internet. These devices are used to direct network traffic across the network. Router testing is performed to accomplish the following tasks:

- Assessing end-to-end router security with or without target knowledge
- Validating functionality
- Analyzing, measuring, and recording the bandwidth and speed of an Internet connection
- Manipulating, analyzing, and recording the bandwidth of an Internet connection and the speed of data transfer
- Determining the reliability and performance of the router; a router's performance determines how efficiently it forwards packets.
- Guaranteeing whether or not the router is able to handle various network environments
- Providing a single point of reference for router security assessment and a countermeasure for identified flaws

General Requirements

The following general requirements are necessary for testing a router:

- *Understanding the organization's network environment*: The organization may deploy any network architecture such as LAN or PAN. It is necessary to check the configuration of all network devices.
- *Understanding router placement in the network architecture*: The placement of the router should be correct in the network architecture, because it routes all the network traffic in and out of the organization.
- *Understanding the traffic managed by the router*: Because the router is a layer-3 device and routes the packets, it is necessary to know about the packets that the router manages.

While performing a test on the router for security, a penetration tester should test for the following issues:

- *Misconfigurations of routers*: A misconfigured router presents itself as the best route to the network, resulting in disorder. Routers that use Border Gateway Protocol (BGP) do not request digital identification from other neighboring networks on the Internet. The attacker can take advantage of this situation to redirect traffic or capture data.
- *Product-specific vulnerabilities of the router*: IOS vulnerabilities and the HTTP configuration administrative access vulnerability in Cisco routers, for example, can be found using vulnerability scanners. A penetration tester can also find these vulnerabilities using Web browsers. A compromise of routing devices compromises the entire network.

Technical Requirements

To properly test a router, a tester must meet the following technical requirements:

- *Knowledge of the basics of routing*: These include the following elements:
  - The definition of networking devices

- Functions of the router in layer 3
- Routing protocols used

- *Knowledge of routing protocols to understand the routing protocol attacks*: Routing protocols determine the best path for packets through the use of algorithms. The following major routing protocols are often used:
  - Border Gateway Protocol (BGP)
  - Interior Gateway Routing Protocol (IGRP)
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)

Protocols directed through a network are called routed protocols. The following major routed protocols are used:

- Internet Protocol (IP)
- Novell Internetwork Packet Exchange (IPX)
- AppleTalk

Routers can be compromised using the following methods:

- Cracking the password of the router.
- Accessing the router using HTTP and attempting brute forcing. If the router can be accessed using HTTP, then it is possible to do the following:
  - Execute system commands directly on the router
  - Get the router's configuration
  - Grab the type-7 password hashes
- *Checking for SNMP vulnerabilities*: A tester should try to scan the device for the SNMP port. The router configuration settings can be changed by enabling the RW SNMP string.
- Checking for VTY/TTY access vulnerabilities.
- *Testing for TFTP vulnerabilities*: A tester should try to scan the device for TFTP. If TFTP is enabled on the router, it is very easy to upload a configuration file to the device with no authentication.
- *Testing for router console port vulnerabilities*: Dial-up modems that are attached to console ports on the routers can introduce vulnerabilities.

Steps for Router Penetration Testing

The following steps should be taken when testing a router:

- *Step 1*: Identify the router hostname.
- *Step 2*: Port scan the router.
- *Step 3*: Identify the router operating system and its version.
- *Step 4*: Identify protocols running on the router.
- *Step 5*: Test for packet leakage at the router.
- *Step 6*: Test for router misconfigurations.
- *Step 7*: Test for VTY/TTY connections.
- *Step 8*: Test for router running modes.
- *Step 9*: Test the router's SNMP capabilities.
- *Step 10*: Test for TFTP connections.
- *Step 11*: Test if finger is running on the router.
- *Step 12*: Test for CDP running on the router.
- *Step 13*: Test for NTP.

- *Step 14*: Test for access to router console port.
- *Step 15*: Test for loose and strict source routing.
- *Step 16*: Test for IP spoofing.
- *Step 17*: Test for IP handling bugs.
- *Step 18*: Test for ARP attacks.
- *Step 19*: Test for routing protocol assessment.
- *Step 20*: Test for RIP.
- *Step 21*: Test for OSPF protocol.
- *Step 22*: Test for BGP.
- *Step 23*: Test for EIGRP.
- *Step 24*: Test router denial-of-service attacks.
- *Step 25*: Test the router's HTTP capabilities.
- *Step 26*: Test the an HSRP attack.

Step 1: Identify the Router Hostname

If the router is registered with DNS, a reverse query on the router's IP address will give the DNS name of the router. This DNS name might be the same as the hostname.

Tools such as Nslookup can be used for querying DNS information for hostname resolution. It is bundled with both UNIX and Windows operating systems and can be accessed from the command prompt. When Nslookup is run, it shows the hostname and IP address of the DNS server that is configured for the local system and then displays a command prompt for further queries.

Step 2: Port Scan the Router

Port scanning determines the ports that are open and the services that are active on a router. Each port is enabled with a particular service. Some of the common ports and services used are SMTP on port 23 and FTP on port 21. Open ports can pose a risk, so it is always recommended to disable services and ports that are not in use.

Common ports and services that run on routers can be seen in Table 2-1.

Port	Service	Protocol
21	FTP	TCP
23	Telnet	TCP
25	SMTP	TCP
80	HTTP	TCP
161	SNMP	UDP

Table 2-1 Specific ports are generally assigned to specific services

Step 3: Identify the Router Operating System and Its Version

Knowledge of the operating system of the network device enables a penetration tester to exploit the common vulnerabilities of that operating system. A port-scanning tool that can be used to identify the operating system of a router is Nmap. It identifies the operating system by using a TCP SYN scan.

IDS mechanisms, firewalls, and filters generally restrict connection requests to avoid footprinting of the operating systems or services that are running on the target host. Most hosts on the Internet permit communication using TCP, however. This scan is performed on networks with high data-transfer rates.

The following options can be used for OS detection:

- -O: Enable OS detection
- *--osscan_limit*: Limit OS detection to promising targets
- *--osscan_guess*: Guess OS more aggressively

Step 4: Identify Protocols Running on the Router

The following protocols are commonly used in routers:

- *CDP*: ***Cisco Discovery Protocol (CDP)*** is a self-governing protocol supported by every Cisco networking device. It facilitates obtaining information about the adjacent routers directly linked to a router. These details include the uniqueness of the device, the hardware platform, the IP address of the adjacent router, and the hold time for the router. CDP is activated by default on all Cisco routers, regardless of the IOS version used on them. CDP must be deactivated on all routers inside a network.
- *RIP*: ***Routing Information Protocol (RIP)*** is a distance-vector routing protocol. To choose the route, RIP uses the hop count, which is a router metric. A ***hop*** is the traversing of a packet from one router to another as it travels from its source to its destination. In a RIP network, the maximum hop count is 15. The sixteenth hop is infinity. RIP information can be viewed by using the following commands on the router:
  - *show ip route*: This command displays and checks IP routes in the routing table of a router. It displays all IP routes that are connected statically or directly to the local router.
  - *show ip protocol*: This command displays and checks information, such as the protocol used, routing timers, and other information linked with the local router.
- *RIPv1*: RIPv1 is version 1 of the distance-vector protocol. The protocol is enhanced by the following features:
  - External route tags
  - Subnet masks
  - Next-hop router addresses
  - Authentication
  - Multicast support
- *OSPF*: The ***OSPF (Open Shortest Path First) protocol*** was developed for huge networks with no hop-count constraint. It is a link-state-based routing protocol that measures several metrics to determine the cost, or efficiency, of the route. It provides scalability by allowing division of the domain for ease of management. The protocol allows subnetting together with VLSM and noncontiguous subnets. OSPF checks the link operation by sending "hello" packets that do not consume excess bandwidth. The protocol has the option to tag the routes through which the packets traverse.
- *IGMP*: IGMP (Internet Group Management Protocol) allows dynamic participation of Internet hosts using multitasking. A host group is a group of hosts that use a specific multicast address. The protocol permits the router to determine the host groups existing on the network.

Step 5: Test for Packet Leakage at the Router

A Cisco router can be identified by the RST packets that it sends in response to the SYN packet on port 1999. The response is independent of whether the port is open or closed.

Step 6: Test for Router Misconfigurations

An attacker can easily gain access to the system if the router is misconfigured. RATS (Rough Auditing Tool for Security) can be used to monitor Cisco IOS routers to assess the security measures configured on them. This tool is supported by Windows and UNIX platforms.

Step 7: Test for VTY/TTY Connections

VTY/TTY connections connect terminal devices directly to the router. The console port is not set to secured mode during the router's configuration. This means the VTY/TTY access is insecure. Asynchronous access to the router is possible and should be checked. To enable connections simultaneously, the five ports that are available on the router can be used. To test the VTY/TTY connections, the following information is required:

- IP address of the router
- Telephone number through which the router and modem are connected
- Console access to the router
- Open ports

Step 8: Test for Router Running Modes

The common modes for which routers are configured are user and privileged modes.

- *User mode*: In user mode, the router displays the hostname appended by the > symbol, for example, TargetRouter>. In user mode, the commands are limited.
- *Privileged mode*: Enable mode is also called privileged mode. To access enable mode, a user types **enable** at the user-mode command prompt. If a password is not configured and the "TargetRouter#" prompt appears, then the router is vulnerable. If the router prompts for the password, an attacker can perform brute-force password attacks.

Step 9: Test the Router's SNMP Capabilities

Simple Network Management Protocol (SNMP) is a division of the Internet protocol suite as defined by the Internet Engineering Task Force (IETF). The protocol allows administrators to examine and manage network-connected devices.

The SNMP protocol is a solution for all types of TCP/IP networks. It is useful for the transfer of data from the client location to the server location, where the data are stored in the form of logs for centralized analysis and viewing. Some examples of network management software are IBM's Tivoli, Microsoft's MOM, and HP's OpenView.

Tools such as SNMP Sniff can be used to capture a password from the network when someone connects to a device using SNMP.

Step 10: Test for TFTP Connections

Trivial File Transfer Protocol (TFTP) is a basic file transfer protocol with the operations of a fundamental form of FTP. TFTP is used when booting devices that do not have any mass storage devices. TFTP uses UDP for transfer and lacks security features. Servers use TFTP to boot diskless workstations, terminals, and routers. Routers and switches connect to a TFTP server during firmware upgrades. For a majority of routers, TFTP is used to fetch and deliver configuration files to these routers.

The following details are true of TFTP:

- It employs UDP (port 69) as a transport protocol.
- It cannot enumerate directory contents.
- It has no verification or encoding mechanisms.
- It is employed for reading files from, or writing files to, a remote server.

The router configuration file can be retrieved using TFTP commands such as **tftp <tftp server> get <device-name>.cfg**.

Step 11: Test if Finger Is Running on the Router

Finger services are used to detect the users present on a particular system by employing a finger program. This service is risky and can result in a system being compromised, as it provides complete user information. The finger service should be disabled in /etc/inetd.conf.

Finger services expose the system's users on port 79 TCP/UDP by default. A tester can check if the finger service is running on the router by entering the following commands:

```
finger -l @<router-ip-address>
finger -l root@<router-ip-address>
```

Step 12: Test for CDP Running on the Router

Cisco Discovery Protocol (CDP) is a layer-2 protocol that identifies routers present on the same link. The following information pertaining to the broadcasting Cisco router is present in the CDP messages:

- Device ID (hostname)
- Port ID (port information about the sender)
- Operating system platform

- IOS software version being used
- Capabilities of the router
- Network IP address

These details are sent along with the MAC address, which is visible to all the routers in the segment. By default, CDP is enabled on Cisco routers. However, the CDP protocol is not routed, so the tool is helpful in the local segment. If the default configuration options are not updated, routers broadcast the messages once every 30 seconds. Cisco IOS employs the device ID as a tool to identify whether or not the received message is an update and the adjacent routers are already identified.

Step 13: Test for NTP

The ***Network Time Protocol (NTP)*** is an Internet protocol used for synchronizing time across computer networks and is enabled by default. The Network Time Protocol is often used on border routers, as many companies use these border routers to synchronize their internal servers. NTP is one of the oldest Internet protocols and is designed to be exceedingly fault tolerant.

A clock can be set on a server using NTP. This time is passed to the other systems connected to it. An intruder can change the system clock and corrupt the time.

Step 14: Test for Access to Router Console Port

Router console ports allow access to the router via a terminal. If physical access to the router is possible, then the attacker can override any security measures. Physical security can be achieved by providing user authentication, which can help prevent malicious attacks and protect against spoofing, port scanning, etc. Making the router accessible to only a limited number of users can prevent breaches of security.

The following techniques can help provide physical security to the router:

- Placing the router in a closed room and making it inaccessible to unauthorized or malicious users
- Using a password to protect the router console
- Giving privileges to change router configurations to authorized administrators only

After accessing a router, an administrator should close the console terminal session, as an open session could provide an access point for an attacker.

Step 15: Test for Loose and Strict Source Routing

Source routing is a method in which the sender specifies a path that a packet should take in order to travel in a network. Source routing is used for purposes such as mapping the network, troubleshooting, improving performance, and hacking. Source routing is performed in the following two ways:

1. *Loose source routing*: In this type of routing, the sender defines one or more hops that the packet must go through.
2. *Strict source routing*: In this type of source routing, the exact path a packet must take is defined.

The ping utility can be used with the source routing options on Windows to perform both types of routing.
Loose routing:

```
ping -j <hosts>
```

Strict routing:

```
ping -k <hosts>
```

Step 16: Test for IP Spoofing

IP spoofing is a method in which an attacker accesses a computer by assuming an authenticated identity. The attacker does this by sending an IP address to the computer and making it appear that the message is coming from a trusted host.

The following different techniques can be used to conduct IP spoofing:

- Domain Name Service (DNS)
- TCP sequence number prediction

- Packet forging using UDP
- Source routing

Step 17: Test for IP Handling Bugs

ICMP redirect helps notify the hosts on a data link that a better route is available for a particular destination. It can also specify new gateways for specific networks.

Step 18: Test for ARP Attacks

ARP (Address Resolution Protocol) is a TCP/IP protocol that maps IP addresses to hardware addresses used by the data-link layer. The following types of attacks can be conducted through the manipulation of ARP:

- *ARP spoofing*: ARP spoofing is possible only on a local network. If it is possible, a man-in-the-middle attack can be conducted. In this type of attack, an attacker gains access to information exchanged between two parties. The attacker can use ARP spoofing to access information that is exchanged between the originating host and the receiving host, and can alter the data without the knowledge of either party.
- *MAC flooding*: MAC flooding is an attack on the ARP cache of network switches. When the switches in the network are flooded with requests, they change to hub mode. In hub mode, the switch becomes too busy to enforce its port security features and, therefore, broadcasts all network traffic to every computer in the network. The attacker can then sniff the packets that the compromised switch broadcasts.

The Ettercap tool can be used to conduct ARP attacks. It supports many features related to network and host analysis, such as sniffing live connections, content filtering, and so on. It also supports active and passive dissection of protocols. Ettercap also supports active and passive dissection of protocols.

Step 19: Test for Routing Protocol Assessment

Routing protocols help exchange routing information among networks; this helps in building routing tables. Many of these routing protocols have variations in authentication, and thus they might have weak or no authentication. Attackers can easily manipulate routing tables, and spoofed packets can also be sent to these tables. If authentication is not enabled for these protocols, attempts can be made to inject RIP packets into the network.

Step 20: Test for RIP

Routing Information Protocol (RIP) is a standard for exchanging information between hosts and gateways. It is designed to carry out its functions on moderate-capacity networks using reliable homogenous technology.

There are two versions of RIP:

1. *RIPv1:* RIP version 1 does not support authentication of routing updates; therefore, the routing updates can be easily sniffed.
2. *RIPv2:* RIP version 2 supports both plaintext and MD5 authentication.

The following tools can be used for brute-force attacks and cracking RIPv2 authentication:

- *L0phtCrack*: This is a password auditing and recovery application that uses multiple assessment processes to aid the administrator in minimizing security risks. L0phtCrack helps in identifying and solving the security weaknesses that evolve from weak or easily guessed passwords.
- *John the Ripper*: This tool is able to detect vulnerabilities in passwords.

Step 21: Test for OSPF Protocol

OSPF (Open Shortest Path First) is a routing protocol used for Internet protocol networks. This protocol was developed to handle large heterogeneous networks that became difficult for RIP to handle. OSPF is based on the SPF (Shortest Path First) algorithm, and its specifications are in the public domain.

Open Shortest Path First supports the following two forms of authentication:

1. Plaintext
2. MD5

Plaintext authentication is preferred only if the adjacent devices do not support the more secure MD5 authentication. In MD5 authentication, both routers use the same secret key, which is used to generate the hash and append it to the message.

A dictionary attack, along with a brute-force attack, can be used to crack the password so that the message can be read and routing updates can be modified.

Step 22: Test for BGP

BGP (Border Gateway Protocol) is an external routing protocol. It enables communication between multiple transit autonomous systems. It is efficient and flexible and requires little bandwidth. It provides greater administrative efficiency by making use of the path attributes that include the AS (autonomous system) numbers along with the information on the various routes. It supports internal sessions that run between the same autonomous system and external sessions that run between the routers in different autonomous systems.

BGP sessions can be hijacked, and incorrect information about the routing tables can be injected within the hijacked session. If the TCP number sequence that uses BGP is identified, a session hijacking attack can be attempted. Tools such as Hunt and T-sight aid in BGP session hijacking. Also, services such as Inter-domain Routing Validation (IRV) provide greater security for BGP by working independently from the protocol itself.

Step 23: Test for EIGRP

The Enhanced Interior Gateway Routing Protocol (EIGRP) is a proprietary routing protocol used by Cisco systems. EIGRP is an enhanced version of IGRP that uses the distance-vector method. The efficient and convergent behavior of the architecture of EIGRP has led to its enhancement as an independent network-layer protocol. EIGRP has the capability to minimize the use of bandwidth and processing power along with the routing instability that occurs after changes in the topology. The maximum hop count of the packets routed by EIGRP is 224. Its authentication works similar to that of RIPv2 and supports MD5 encryption.

Step 24: Test Router Denial-Of-Service Attacks

A ***denial-of-service attack*** is an attack on a network or computer system that creates a disturbance in the services provided. This interruption could be a jam in the network traffic caused by bandwidth consumption or an overload of system resources.

Network denial-of-service (DoS) attacks are conducted in several ways, including the following:

- *Malformed-packet attack*: In this method, an attacker sends packets that are formatted incorrectly for the target protocol.
- *Packet-flood*: This type of attack occurs when the attacker sends too many unprocessable packets to the destination.

Step 25: Test the Router's HTTP Capabilities

Web browsers can often be used to manage routers. HTTP is used to remotely manage routers. HTTP is presented in plaintext, and passwords can be sniffed. The router must have a Web management port listener to access it in this manner. By using a browser such as Internet Explorer and a router remote management tool (e.g., Cisco Secure Policy Manager), an attacker can access and manipulate routers remotely.

Step 26: Test the HSRP Attack

The Hot Standby Router Protocol (HSRP) attack requires two routers. To test the HSRP attack, a tester can perform the following steps:

1. Send packets with high priority so that the active network slows down.
2. Forward all incoming packets to the correct destination.
3. Test if the traffic sent via HSRP group is forwarded to the specified IP address.
4. Check if the man-in-the-middle attack is established as all traffic is forwarded to the specified IP address.

Router Testing Report

Testers should document all their router testing findings in the penetration testing report. Router testing can be a tedious task, and testers must be patient, because there is a large amount of traffic recorded.

Testing Switches

The following steps can be taken to test switches:

- *Step 1*: Test address cache size.
- *Step 2*: Test data integrity and error checking.
- *Step 3*: Test for back-to-back frame capacity.
- *Step 4*: Test for frame loss.
- *Step 5*: Test for latency.
- *Step 6*: Test for throughput.
- *Step 7*: Test for frame error filtering.
- *Step 8*: Perform a fully meshed test.
- *Step 9*: Perform a stateless QoS functional test.
- *Step 10*: Test the spanning tree network convergence performance.
- *Step 11*: Test OSPF performance.
- *Step 12*: Test for VLAN hopping.
- *Step 13*: Test for MAC table flooding.
- *Step 14*: Test for ARP attacks.
- *Step 15*: Check for VTP attacks.

Step 1: Test Address Cache Size

Testers can use the following steps to test address cache size:

1. Send frames of half the size of the initial user-specified table size.
2. Send generic frames at a specified frame rate.
3. If the switch is able to handle all of the addresses, increase the frame rate.
4. Repeat the above steps until frame loss or flooding is detected.

Testers can use tools such as Ixia's IxScriptMate to automate this process.

Step 2: Test Data Integrity and Error Checking

The following steps can be taken to test for data integrity:

1. Check the switch's ability to forward frames under certain traffic rates without corrupting the payload. Frames are transmitted with a predefined data pattern.
2. Verify whether the switch forwards the frames properly.
3. Calculate the number of sequence errors and the number of data errors.

Step 3: Test for Back-To-Back Frame Capacity

The back-to-back value is the number of frames in the longest burst that the switch will handle without the loss of any frames. For this test, the tester sends a burst of frames with minimum interframe gaps to the switch and counts the number of frames the switch forwards.

If the count of transmitted frames is equal to the number of frames forwarded, the length of the burst is increased and the test is rerun. If the number of forwarded frames is less than the number transmitted, the length of the burst is reduced and the test is rerun.

The trial length must be 2 seconds and should be repeated 50 times, with the average of the recorded values being reported.

Step 4: Test for Frame Loss

For this test, the tester sends a specific number of frames at a specific rate through the switch to be tested and counts the frames that the switch transmits. The frame loss rate at each point is calculated using the equation ((input_count – output_count) * 100)/input_count.

Step 5: Test for Latency

The following steps can be used to test for latency:

1. Send a stream of frames through the switch at the determined rate to a specific destination for a duration of 120 seconds.
2. Provide an identifying tag in one frame after 60 seconds.
3. Record the time at which this frame is fully transmitted (time stamp A).
4. Record the time at which the receiver received the tagged frame (time stamp B).
5. The latency is time stamp B minus time stamp A.
6. Repeat the test 20 times, with the reported value being the average of the recorded values.

Step 6: Test for Throughput

To test for throughput, the tester sends a specific number of frames at a specific rate through the switch and then counts the frames that the switch transmits. If fewer frames are received than were transmitted, the rate of the offered stream is reduced and the test is rerun.

Step 7: Test for Frame Error Filtering

In this step, the tester checks if the switch correctly filters illegal frames, such as the following:

- Undersized frames
- Oversized frames
- Frames with CRC errors
- Fragmented frames
- Frames with alignment errors
- Frames with dribble errors

Step 8: Perform a Fully Meshed Test

The tester checks the total number of IP frames that the switch can handle when it receives frames on all its ports. Each port in the test sends frames to all other ports in an evenly distributed, round-robin fashion at a specific user-defined rate.

Step 9: Perform a Stateless QoS Functional Test

The tester uses the following steps to conduct this test:

1. Measure the baseline performance of the switch with and without QoS.
2. Inject stateless traffic into the network.
3. Check the latency and the packet loss on the egress traffic port.
4. Measure and record when QoS is disabled on the switch, and when QoS with IP precedence classifying and marking is enabled on the switch.

Step 10: Test the Spanning Tree Network Convergence Performance

The following steps can be used to conduct this test:

1. Measure the following fields:
   - Network convergence based on the handling of topology changes notifications
   - Configuration BPDUs (Bridge Protocol Data Units), as well as traffic switchover
2. Check the switch spanning tree convergence performance.
3. Check for any changes in path cost to root changes.
4. Check if the bridge link slows down.

Step 11: Test OSPF Performance

The following steps can be used to conduct this test:

1. Set the defined routes and a topology.
2. Test the no-drop throughput and latency.
3. Execute the test either with OSPFv2 or OSPFv3 protocols.
4. Measure the OSPF performance and scalability of the switch.

Step 12: Test for VLAN Hopping

The tester can use the following steps to conduct this test:

1. Spoof a computer to appear as another switch.
2. Send a fake DTP negotiate message requesting to be a trunk.
3. Check whether the real switch turns on 802.1Q trunk.
4. Check all traffic for all VLANs sent to the computer.

Step 13: Test for MAC Table Flooding

The tester can use the following steps to conduct this test:

1. Use the macof tool to flood the content addressable memory (CAM) with random MAC addresses.
2. Check whether all ports are flooded.
3. Check whether sniffing can be conducted in a switched environment.

Step 14: Test for ARP Attacks

The tester can use the following steps to conduct this test:

1. Send a spoofed ARP reply to another host.
2. Check the MAC address of the other host.
3. Associate the MAC address of the original machine with the host MAC address in the MAC address table of the switch.
4. Check all the frames that are being sent to the host address.

Step 15: Check for VTP Attacks

The following steps can be used to conduct this test:

1. Eliminate all the VLANs by using VTP (VLAN Trunk Protocol).
2. Check whether other users share the same VLAN.
3. Change the IP to be on the same network on which the other users are present.
4. Check whether an attack can be conducted on the host.

Chapter Summary

- A router is a device that forwards data packets by determining the destination network point on the Internet.
- Router testing is necessary to assess end-to-end router security, bandwidth, functionality, performance, and reliability.
- Nslookup is a valuable tool for querying DNS information for hostname resolution.

- Knowledge of the operating system of the network device will enable the user to exploit the common vulnerabilities in that operating system. A port-scanning tool that can be used to identify the operating system of a router is Nmap.
- Cisco Discovery Protocol (CDP) is a self-governing protocol supported by every Cisco networking device.
- Routing Information Protocol (RIP) is a distance-vector routing protocol that uses routing metrics (hop count) to select a route.
- An attacker can easily gain access to the system if the router is misconfigured.

Chapter 3

Firewall Penetration Testing

Objectives

After completing this chapter, you should be able to:

- Read a firewall rule set
- Identify firewalls by type
- List firewall limitations
- Locate a firewall
- Port scan a firewall
- Test to identify the firewall architecture
- Use Firewalk
- Test covert firewall channels
- Document a firewall penetration test

Key Terms

Application-level gateways firewalls that act as a protection layer for internal network applications by developing proxy services

Circuit-level gateways firewalls that help in determining whether or not a requested session is legitimate by checking TCP handshaking between packets from trusted clients or servers and untrusted hosts

Covert channel a hidden communications mechanism

Firewall a hardware device or piece of software that monitors and filters data packets that traverse a network perimeter

Firewall policy the implementation of an information security policy through a firewall

Packet-filtering firewalls firewalls handle access to a network by evaluating the incoming and outgoing packets

Stateful multilayer inspection firewalls firewalls that are a combination of packet-filtering, application-level, and circuit-level gateways

Introduction to Firewall Penetration Testing

A ***firewall*** is a hardware device or piece of software that monitors and filters data packets that traverse a network perimeter. Firewalls prevent unauthorized users from accessing a network and scrutinize the data transfers to and from the network. Firewalls are extensively used to grant users secure access to the Internet while separating a company's public Web server from its interior network. Advanced personal firewalls identify outbound traffic to protect against spyware that sends surfing habits to a Web site. They alert the user when an application initiates an outbound request for the first time.

A firewall checks all the traffic that travels between two networks. Firewalls record all attempts to enter a private network and generate alarms if the attempt is unauthorized. They restrict public access to private networked resources such as host applications. Firewalls monitor and filter both inbound and outbound traffic.

Firewall Policy

Firewall policy is essentially the implementation of an information security policy through a firewall. A firewall policy specifies how to build and handle firewall application traffic like telnet or e-mail. Without a security policy guide, it is difficult to manage firewalls. A good firewall policy will provide recommendations for testing and periodically updating the policy.

Before creating a firewall policy, an organization should conduct risk analysis on applications that are necessary for the organization's business. The output of this risk analysis should include both the applications list and how to secure those applications.

The following steps are required for creating a firewall policy:

1. Identify the network applications that are found to be critical.
2. Identify the vulnerabilities that are related to network applications.
3. Prepare a cost-benefit analysis to secure the network applications.
4. Build a network applications traffic channel to identify the protection method.
5. Build a firewall rule set that depends on the application's traffic channel.

Firewall Rule Sets

Incoming packets are checked against a rule set that the firewall's administrator develops. These rules are customized according to the individual needs of the network that the firewall is protecting, as shown in Figure 3-1.

The information fields in a rule set include the following:

- Source and destination address of the packet
- Type of packet

The following are some of the types of packets that a rule set can be written to deny:

- Inbound packet that contains Internet Control Message Protocol (ICMP) traffic
- Inbound packet that contains IP source routing information
- Inbound packet that contains nonauthenticated source system with a firewall's destination address
- Inbound packet from nonauthenticated source system that contains Simple Network Management Protocol (SNMP) traffic
- Packets that contain both source and destination address as 0.0.0.0
- Packets that contain direct broadcast addresses

Figure 3-1 Firewalls use complex rules to decide which packets to allow.

Firewall Logging Functionality

Most firewalls provide advanced logging functionality. The commonly accepted logging application for firewalls is the UNIX syslog utility. Syslog provides centralized logging options as well as various options for checking and parsing the logs. This firewall logging option is available for major operating systems like Windows NT/2000/XP, UNIX, and Linux variants.

If a firewall does not support syslog, it should provide some sort of internal logging functionality. There are a number of tools to manage and parse these firewall logs.

Periodic Review of Information Security Policies

Administrators should plan periodic reviews for information security policies. They should review and update information security policies every six months. If a firewall application is upgraded, then the firewall rule set must be modified accordingly. It is mandatory to audit firewall installations, firewall systems, and other resources on a regular basis. Periodic reviews of information security policies should include the following elements:

- *Computer systems*: Administrators should perform periodic reviews for computer systems that contain information security policy.
- *Actual audits*: Administrators should prepare audits for security policies using periodic reviews.
- *Vulnerability assessments*: Administrators should assess the vulnerability of the system by performing periodic reviews.
- *Components used for backup communications*: Administrators should list the components that require periodic reviews for backup communications.

Firewall Implementation

Firewalls can be implemented as application-based or commercial-based firewalls.

Application-Based Firewall

Application-based firewalls are generally more secure than commercial-based firewalls, because they do not face any security vulnerabilities related to the basic operating system. This type of firewall uses application-specific integrated circuit (ASIC) technology. Application-based firewalls are also faster than commercial-based systems.

Commercial-Based Firewall

Firewalls that are implemented on commercial operating systems must be highly flexible and scalable so that they can be customized to improve the performance of the organization. These firewalls tend to be more vulnerable to security threats.

Maintenance and Management of Firewalls

The three mechanisms used by commercial firewall platforms for configuration and maintenance are:

1. *Command-line interface (CLI) configuration*: CLI configuration mode enables the administrator to configure the firewall by typing commands at the command line.
2. *Graphical user interface (GUI) configuration*: GUI configuration mode enables the administrator to configure the firewall through a graphical user interface.
3. *Web-based configuration*: For Web-based interfaces, security is provided through Secure Sockets Layer (SSL) encryption, along with a user ID and password.

Administrators should do the following to manage and maintain a firewall:

- Monitor the firewall to identify all suspicious activities like port scans or half scans.
- Configure the firewall to log all activities that may identify suspicious activities.
- Store all the logs on a secure server to prevent unauthorized access by hackers. Intruders often alter logs in an attempt to avoid detection.
- Configure the firewall to provide alerts. Before sending messages, alerts should be properly configured and tested.

Types of Firewalls

Firewalls can be either hardware or software, as shown in Figures 3-2 and 3-3. Firewalls are part of a security scheme that prevents unauthorized users from accessing the network or scrutinizing data transferred to and from the network.

Firewalls can function in the following ways:

- *As packet-filtering firewalls*: Packet-filtering firewalls monitor the data packets entering the network. They inspect the IP/port addresses or protocols of the incoming and outgoing packets and then decide to allow or reject them. These firewalls operate on the network layer of the OSI model. They are also known as screening routers and are considered first-generation firewalls.
- *As application-level gateways*: Application-level gateways operate in the application layer of the OSI network model. They are also called *proxy servers* and are considered third-generation firewalls.
- *As circuit-level gateways*: These firewalls work at the session layer of the OSI model. They first validate the connection that is used to hide the information of the protected networks, and then allow the data to be transmitted. These are also considered third-generation firewalls.
- *As stateful multilayer inspection firewalls*: Packet-filtering firewalls, application-level gateways, and circuit-level gateways are combined to create stateful multilayer inspection firewalls. These too are considered third-generation firewalls.

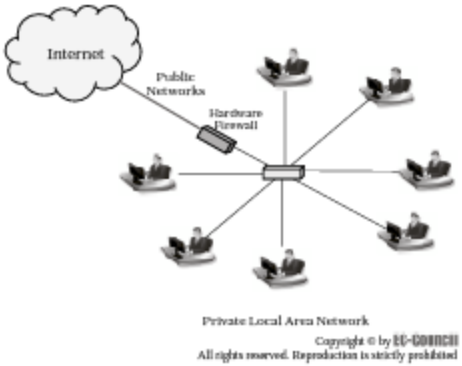


Figure 3-2 Hardware firewalls are often part of a TCP/IP router.

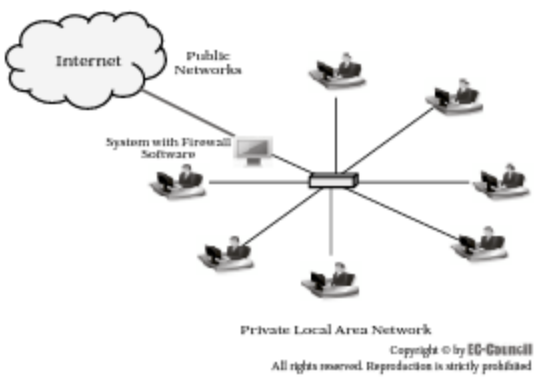


Figure 3-3 Software firewalls are installed in computers.

- *As dynamic packet filters*: The firewall facilitates monitoring the state of all active connections and uses this information to determine access. By monitoring session information such as IP addresses and port numbers, a much tighter security can be implemented than in earlier firewall versions. These are considered fourth-generation firewalls.

Packet-Filtering Firewall

Packet-filtering firewalls monitor data packets entering the network. They act as tools to filter network traffic, as shown in Figure 3-4. ***Packet-filtering firewalls*** handle access to a network by evaluating the incoming and outgoing packets.

Routers are the most common packet-filtering devices. They accept only authorized data packets and reject unauthorized data packets. Packet filters check the data header, conceal the header with a new header, and then send the packet to the intended location on the network. The header consists of common information such as the capacity of the packet, the protocol used for transmission, and the IP address of both the source and the destination.

Factors such as source IP address, destination IP address, protocols used, source port number, and destination port number are included in packet-filtering rules. Packet filtering operates at the network layer of the OSI network model. NAT offers packet filtering by transmitting only the requested traffic to private network hosts. The primary advantages of packet filters are that they do not utilize bandwidth and that they provide security for a relatively low cost.

Packet filtering can be approached in the following ways:

- *Stateless packet filtering*: This type of packet filtering examines the content of packet headers and decides whether or not to transmit the packets. It mainly monitors the data traffic.
- *Stateful packet filtering*: This type of filtering maintains a record of transactions that have already occurred. It decides whether to accept or deny a packet based on both the current packet's information and this record.

Circuit-Level Gateway

Circuit-level gateways operate at the session layer of the OSI model or the TCP layer of TCP/IP. These gateways help in determining whether or not a requested session is legitimate by checking TCP handshaking between packets from trusted clients or servers and untrusted hosts, as shown in Figure 3-5. When information is passed using a circuit-level gateway to a remote host, it looks as if the information originated from that particular gateway. This hides information about protected networks.

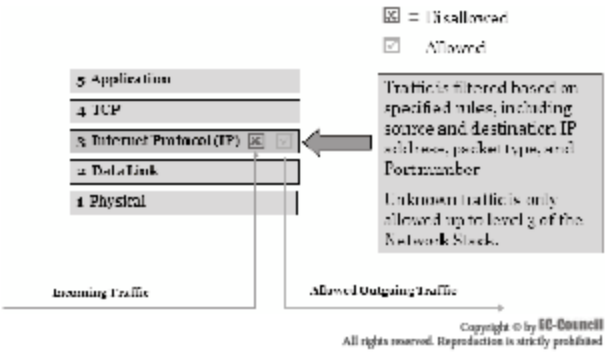


Figure 3-4 IP packet-filtering firewalls monitor the packets coming into a network.

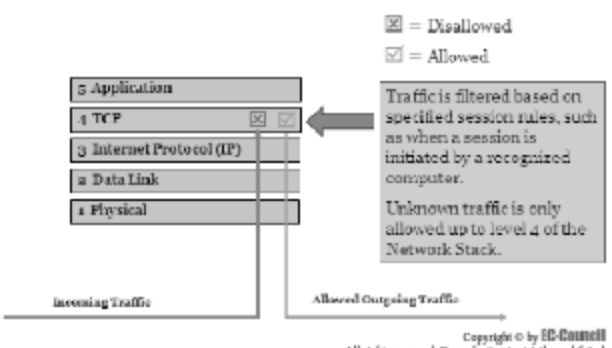


Figure 3-5 Circuit-level gateways monitor packets that travel between clients and servers.

Application-Level Gateways

Application-level gateways are also called proxy servers. They operate at the application layer of the OSI network model, i.e., packets can be filtered at the application layer itself, as shown in Figure 3-6. ***Application-level gateways*** act as a protection layer for internal network applications by providing proxy services. Such services protect users from directly connecting to the Internet if any requests are made to access Web pages. Thus, viruses, worms, etc., are prevented from infecting the system. An application-level gateway proxy does not allow traffic like FTP, Gopher, or telnet to pass through.

Application-level gateway services are limited, as HTTP commands like POST and GET are filtered. The primary advantage of application-level firewalls is that they provide a high level of security. However, they can have a significant impact on the capacity of network cables to transmit information.

Stateful Multilayer Inspection Firewall

Stateful multilayer inspection firewalls are a combination of packet-filtering, application-level, and circuit-level gateways, as shown in Figure 3-7. They offer higher security levels and operate at the network layer of the OSI model.

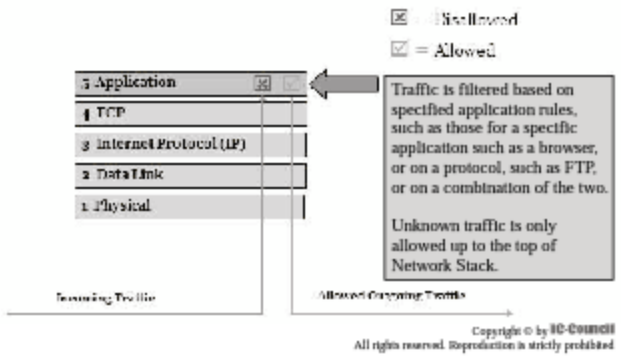


Figure 3-6 Application-level gateways monitor packets based on specified application rules.

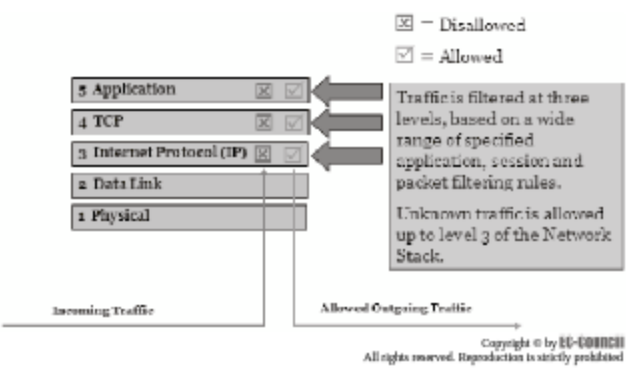


Figure 3-7 Multilayer inspection firewalls use a combination of application, session, and packet-filtering rules.

Stateful multilayer inspection firewalls inspect the incoming and outgoing packets based on the IP header. They check whether the SYN, ACK, and sequence number are logical, in order to identify the legitimacy of the session packets. The packet contents are evaluated at the application level.

Stateful multilayer inspection firewalls provide better security levels than packet filters, and they provide transparency to end users. These firewalls are relatively expensive, and the device needs to be administered in order to maintain its security levels.

Firewall Limitations

Firewalls have the following limitations:

- A user cannot be restricted from dialing in or out of the network using modems; thus, bypassing the firewall cannot be prevented.
- Firewalls cannot control employee misbehavior and negligence. A policy concerning the proper usage of passwords and user accounts has to be strictly enforced.
- A firewall cannot protect users from malicious insiders. For example, a firewall cannot prevent a user from copying confidential data to a disk, tape, or paper or from stealing physical storage devices.
- A firewall cannot prevent a user from being attacked when the data have to pass through connections that are not firewalled. For example, a firewall cannot prevent an intruder from breaking into a modem.
- A firewall cannot ensure complete protection from viruses such as PC viruses and Macintosh viruses. Even sophisticated firewalls cannot filter viruses to a great extent. Firewalls cannot completely protect systems against new threats.
- A firewall that has been set up once cannot be expected to protect the system forever.

Steps for Conducting Firewall Penetration Testing

- *Step 1*: Locate the firewall.
- *Step 2*: Conduct a traceroute to identify the network range.
- *Step 3*: Port scan the firewall.
- *Step 4*: Grab the banner.
- *Step 5*: Create custom packets and look for firewall responses.
- *Step 6*: Test access control enumeration.
- *Step 7*: Test to identify firewall architecture.
- *Step 8*: Test firewall policy.
- *Step 9*: Test firewall using the Firewalk tool.
- *Step 10*: Test for port redirection.
- *Step 11*: Test the firewall from both sides.
- *Step 12*: Perform an overt firewall test from the outside.
- *Step 13*: Test covert channels.
- *Step 14*: Perform a covert firewall test from the outside.
- *Step 15*: Test HTTP tunneling.
- *Step 16*: Test firewall-specific vulnerabilities.
- *Step 17*: Document everything.

Step 1: Locate the Firewall

In this step, the tester uses Hping or any other packet crafter to create data packets and send them to the firewall. Hping is a TCP utility used to create IP packets containing UDP, TCP, or ICMP payloads. By crafting packets in order to locate the access points of a targeted system, the tester can search the firewall rule set.

Step 2: Conduct a Traceroute to Identify the Network Range

Running traceroute against the router accomplishes the following tasks:

- Helps determine the path taken by packets to travel through a network between two systems
- Helps determine all the intermediate routers and devices involved in establishing a connection
- Obtains information pertaining to filtering devices and protocols

Step 3: Port Scan the Firewall

Port scanning identifies the IP address of the targeted system; it helps determine open ports and scans for services that are active. Some firewalls can be identified through port scanning. For remote management purposes, most firewall implementations have default ports in use—for example, user authentication, management, VPN connections, etc. A tool such as Nmap can be used to perform port scanning.

Step 4: Grab the Banner

In this step, the tester tries to grab the firewall's banner. The banner reveals the version of the firewall used, which can be used to explore vulnerabilities.

Consider the following example:

```
nc -nvv 10.0.0.1 80
HEAD / HTTP/1.0
HTTP/1.1 503 Service Unavailable
MIME-Version: 1.0
Server: Simple, Secure Web Server 1.1
Date: Tue, 12 Dec 2005 19:08:35 GMT Connection: close
Content-Type: text/html
<HTML>
<HEAD><TITLE>Firewall Error: Service Unavailable</TITLE></HEAD>
```

Step 5: Create Custom Packets and Look for Firewall Responses

Custom packets sent to firewalls can provoke unique responses from the firewall. This technique is useful for analyzing the type of firewall.

Consider the following example:

```
hping 10.0.0.5 -c 2 -S -p 23 -n
HPING 10.0.0.5 (eth0 10.0.0.5): S set, 40 data bytes
60 bytes from 10.0.0.5: flags=RA seq=0 ttl=59 id=0 win=0 time=0.4 ms
```

Step 6: Test Access Control Enumeration

Nmap can be employed to enumerate the firewall access control list. Nmap enumerates the following three states of ports:

1. *Open*: Port is listening
2. *Filtered*: Port is blocked by an access control device (router/firewall)
3. *Unfiltered*: Traffic is passing from access control devices (router/firewall), but the port is not open

Consider the following example:

```
nmap -sA 192.168.0.1
Interesting ports on 192.168.0.1:
```

```
(The 65530 ports scanned but not shown below are in state: filtered)
PORT STATE SERVICE
110/tcp UNfiltered pop-3
13701/tcp UNfiltered VeritasNetbackup
13711/tcp UNfiltered VeritasNetbackup
13721/tcp UNfiltered VeritasNetbackup
13782/tcp UNfiltered VeritasNetbackup
Nmap run completed -- 1 IP address (1 host up) scanned in 12206.371
seconds
```

Step 7: Test to Identify Firewall Architecture

Hping2 is a packet-crafting tool that permits custom-crafted packets to be sent. Hping2 can be used to identify the following port status:

- *Open*: A port is considered open if a SYN/ACK packet is received.
- *Rejected*: If an RST/ACK packet is received, then the target host might have rejected the packet.
- *Dropped*: If a packet is not received, it is assumed that the security devices have dropped it.
- *Blocked*: A connection to the host is considered blocked if an ICMP type 3 code 13 message is received.

Consider the following examples of Hping2 packets that generate responses:

- TCP port 80 is open:

```
hping2 -c 3 -s 53 -p 80 -S google.com
HPING google.com (eth0 216.239.39.99): S set, 40 headers + 0 data
ip=216.239.39.99 ttl=128 id=289 sport=80 flags=SAP seq=0 win=64240
ip=216.239.39.99 ttl=128 id=290 sport=80 flags=SAP seq=1 win=64240
ip=216.239.39.99 ttl=128 id=291 sport=80 flags=SAP seq=2 win=64240
```

- Access to TCP port 139 is rejected by the firewall:

```
hping2 -c 3 -s 53 -p 139 -S 192.168.0.1
HPING 192.168.0.1 (eth0 192.168.0.1): S set, 40 headers + 0 data
ip=192.168.0.1 ttl=128 id=283 sport=139 flags=R seq=0 win=64240
ip=192.168.0.1 ttl=128 id=284 sport=139 flags=R seq=1 win=64240
ip=192.168.0.1 ttl=128 id=285 sport=139 flags=R seq=2 win=64240
```

- TCP port 23 is blocked by a router ACL:

```
hping2 -c 3 -s 53 -p 23 -S gw.example.org
HPING gw (eth0 192.168.0.254): S set, 40 headers + 0 data
ICMP unreachable type 13 from 192.168.0.254
ICMP unreachable type 13 from 192.168.0.254
ICMP unreachable type 13 from 192.168.0.254
```

- TCP probe packets are dropped in transit:

```
hping2 -c 3 -s 53 -p 80 -S 192.168.10.10
HPING 192.168.10.10 (eth0 192.168.10.10): S set, 40 headers + 0 data
```

Step 8: Test Firewall Policy

The two different methods to test firewall policy are as follows:

1. In the first method, the tester obtains hard copies of the firewall configuration and compares them with hard copies of the expected configuration.
2. The second method involves actual in-place testing that determines the configuration of the firewall by attempting to perform operations that should be prohibited.

Step 9: Test Firewall Using the Firewalk Tool

Firewalk is a network-auditing tool that analyzes IP packet responses with the use of traceroute-like techniques in order to determine gateway ACL filters and map networks. System administrators use this tool to increase the security level of their systems.

Firewalk discovers ports that are open on a filtering device by checking the system behind a firewall. It also helps in determining whether or not traffic sent to a given port can pass through a firewall.

Firewalk performs advanced network mapping by transferring packets to all the hosts behind the firewall. This generates a map of the network topology behind the firewall.

Firewalk works by creating IP packets that expire exactly one hop after the firewall of the system (target host). If the packet is allowed to pass through the filtering device, it expires once it passes through the firewall, giving an ICMP TTL-exceeded message. If the filtering device blocks the packet, then no response is generated. A firewall port is assumed to be open when the TTL-exceeded message comes back.

Step 10: Test for Port Redirection

When a port cannot be accessed directly, port redirection helps gain access to those ports. A malicious user might compromise a target system that can help bypass the firewall and then exploit it with the use of port redirection. Once the port redirector is installed, it listens to certain port numbers and redirects all packets received on the listening port to a remote host.

The following tools can perform port redirection:

- *Fpipe*: Fpipe is a TCP redirector that can create a custom TCP or UDP stream. It helps traffic with source port 23 pass through a filtering device in order to connect to internal servers. Fpipe is the only port redirector on the Windows platform that binds to a static port in order to bypass filters. By default, clients use a high source port for connection, which passes through the filter. The firewall acknowledges the stream and lets it through.
- *Datapipe*: Datapipe is a popular port redirector and works on UNIX-based platforms. It is quite time-consuming, because it must run on both the attacker's system and the compromised host.

Step 11: Test the Firewall from Both Sides

In this step, testers check the firewall by concurrently testing it from both sides. From the outside, testers send packets; from the inside, testers analyze the packets that arrive, and vice versa.

Step 12: Perform an Overt Firewall Test from the Outside

In overt firewall testing, the tester creates a network connection from the outside in order to secure the network segment.

The following steps should be used to perform overt firewall testing from the outside:

1. Execute a vulnerability scanner tool from behind the firewall on the firewall system hosts (internal router, firewall host, and external router).
2. Specify the firewall rules with the help of firewall tools such as Firewalk.
3. Examine the response of the firewall for fragmented and spoofed packets that are generated using a packet generator.
4. Make an attempt to reach systems that are behind the firewall.

Step 13: Test Covert Channels

A ***covert channel*** can be defined as a hidden communications mechanism. Hackers use covert channels as a means to conceal their activities. By means of covert channels, a hacker can stealthily access the system's resources, and a backdoor can be installed on the victim's machine.

The Reverse Shell tool can be used to create a secure remote shell tunnel. Commands and interactive shells are launched from the tunnel destination machine to the tunnel originating machine once the tunnel is created.

Step 14: Perform a Covert Firewall Test from the Outside

In covert firewall testing, the tester creates a network connection to protect the network segment from the outside.

The following steps should be performed to perform covert firewall testing from the outside:

1. Specify the firewall rules with the help of firewall tools like Firewalk.
2. Make an attempt to reach systems that are behind the firewall.
3. Examine the response of the firewall for fragmented and spoofed packets that are generated using a packet generator.

Step 15: Test HTTP Tunneling

HTTP tunneling techniques work by sending POST requests to an HTTP server and receiving replies. The POST request can be sent using a URL connection that specifies the hostname, port number, path, and CGI reference.

The HTTPort tool avoids HTTP proxies that block the Internet. HTTPort performs tunneling through either the SSL/CONNECT mode or the Remote Host mode. The SSL/CONNECT mode requires certain features of HTTP, such as the CONNECT HTTP method, to be supported by the proxy, which allows HTTPort to make a tunnel through the proxy. This technique is much faster, but in most proxies, the CONNECT HTTP method is disabled. Encryption is also not possible.

Remote Host mode requires HTTHost, which is a Web server. This technique is slower; however, it works more often, and strong data encryption is also possible.

Step 16: Test Firewall-Specific Vulnerabilities

Hackers scan networks to check for vulnerable services running on the hosts and ports that are open. Open, unused ports are vulnerabilities that firewalls must be aware of. A firewall should block services, such as print and file sharing, from unauthorized access through an open port.

A misconfiguration of a firewall is the main reason hackers are able to find vulnerabilities. An attacker might compromise a target system, which can help bypass the firewall or can make use of other services like dial-up to pass through these systems. Some technically sophisticated users might be able to circumvent security measures by dialing in through remote access to connect to the network and thus open a security hole.

Step 17: Document Everything

Testers should document all findings obtained as a result of the penetration test of the firewall. The following items should be documented:

- *Firewall logs*: Firewall logs play an important role in identifying security breaches, if any. The printout of the firewall logs should be attached to the report.
- *Tools output*: Output generated by the tools that were used to test the firewall should be documented in a format that the client organization can easily understand.
- *Analysis*: Analysis of the findings should be documented in an easy-to-read format.
- *Recommendations (if any)*: Recommendations with respect to the firewall configurations or security measures that need to be taken should also be documented.

Chapter Summary

- Firewalls prevent unauthorized users from accessing a network and scrutinize the data transfers to and from the network.
- Before creating a firewall policy, an organization should conduct risk analysis on applications that are necessary for the organization's business. The output of this risk analysis should include both the applications list and also how to secure those applications.
- Incoming packets are checked against a rule set that the firewall's administrator develops. These rules are customized according to the individual needs of the network that the firewall is protecting.
- Organizations should review and update information security policies every six months.
- Administrators should store all firewall logs on a secure server to prevent unauthorized access by hackers.
- A firewall that has been set up once cannot be expected to protect the system forever.
- Nmap is a tool that can be used to search the network to find the state of ports and services running on those ports.
- Firewalk is a network-auditing tool that analyzes IP packet responses with the use of traceroute-like techniques in order to determine gateway ACL filters and map networks.
- When a port cannot be accessed directly, port redirection helps gain access to those ports.
- Testers should document all findings obtained as a result of the penetration test of the firewall.

Chapter 4

IDS Penetration Testing

Objectives

After completing this chapter, you should be able to:

- Understand intrusion detection systems
- Recognize types of intrusion detection systems
- Understand techniques used to evade intrusion detection systems
- Test intrusion detection systems

Key Terms

Indication an alert generated by an intrusion detection system

Introduction to IDS Penetration Testing

An intrusion detection system (IDS) is software or hardware that identifies, logs, and reports any suspected malicious activity. Some IDS are designed to simply monitor for intrusions and generate alerts, called ***indications***, when they are found, while others immediately take action against the unwelcome activity. This chapter familiarizes you with the different types of IDS and teaches you how to test their effectiveness.

Types of Intrusion Detection Systems

Network IDS

A network IDS (NIDS) checks every packet entering the network for any unexpected or incorrect data. Unlike firewalls that filter data packets containing obviously malicious content, an NIDS checks every packet thoroughly, regardless of whether it is explicitly permitted. If it detects an anomaly, it assigns a threat level to the incident and generates an alert. An NIDS attached to a network is shown in Figure 4-1.

Host-Based IDS

A host-based IDS (HIDS) analyzes an individual system's behavior. This type of IDS can be installed on any system, ranging from a desktop PC to a server, making it more versatile than an NIDS. One example of a host-based system is a program that receives application or operating system audit logs.

These programs are especially effective in detecting abuses by authentic users. If one of these users attempts unauthorized activity, host-based systems quickly detect it and collect relevant information. HIDS are also effective at detecting unauthorized file modification because they are focused on watching for changes in the local system. These systems are platform-centric, with most working on Windows, but there are several for UNIX platforms.

Figure 4-2 shows an HIDS.

Application-Based IDS

An application-based IDS is similar to a host-based IDS, except that it is specifically designed for one application. Because it is so specialized, it is extremely effective in protecting that one application.

Multilayer Intrusion Detection Systems (mIDS)

Multilayer intrusion detection systems (mIDS) integrate many different IDS technologies into a single engine. These systems analyze data from integrity monitoring software, system logs, IDS logs, and firewall logs, and provide one clear and concise report. Figure 4-3 shows how an mIDS combines multiple technologies into one.

Using a multilayer intrusion detection system is faster than using many IDS, which makes them very valuable when recognizing and recovering from an incident.

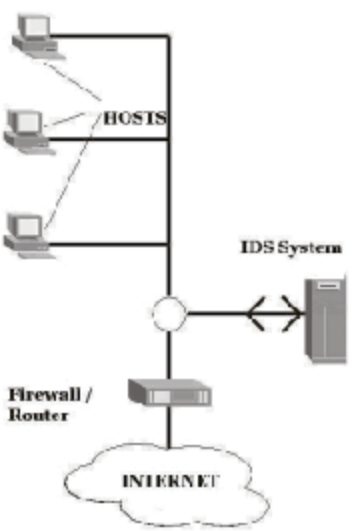


Figure 4-1 An NIDS sits between the firewall or router and the host systems.

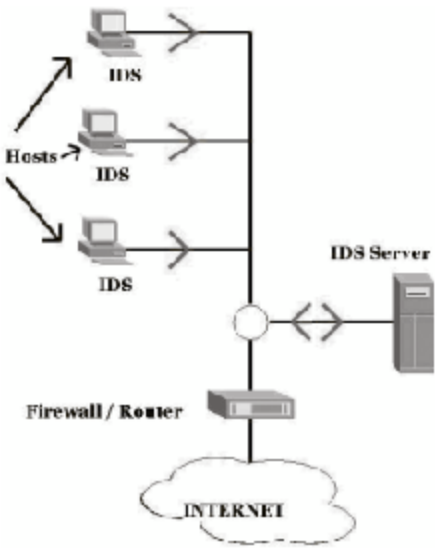


Figure 4-2 An HIDS watches traffic inside individual hosts.

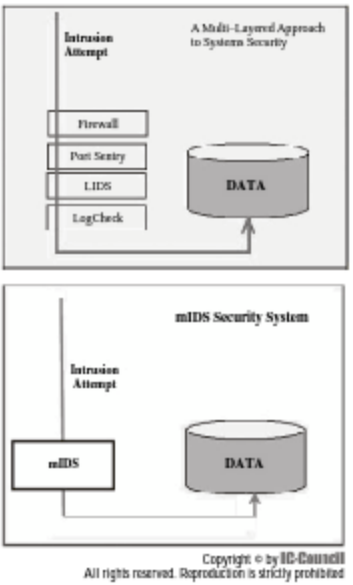


Figure 4-3 An mIDS analyzes multiple types of data.

Wireless Intrusion Detection Systems (WIDS)

A wireless IDS (WIDS) is specifically made to monitor wireless networks. WIDS analyze user and system activities, detect abnormal network activity, and detect policy violations for WLANs. They watch all local wireless transmissions for known signatures of malicious content.

A wireless IDS can be either centralized or decentralized. A centralized WIDS contains a grouping of individual sensors that gather and forward all information to a central management system, where WIDS data are stored and processed. All sensors collaborate with one another, increasing the system's accuracy. In a decentralized WIDS, one or more devices separately collect and analyze data. This system is less expensive and is useful for smaller WLANs.

Using more sensors increases the probability that an attack will be detected. It can also make it possible to determine the physical location of an attacker.

A WIDS detects the following attacks and events:

- Rogue WAPs
- Nonencrypted 802.11 traffic
- Monkey/hacker jacks, null probes, null associations, floods, and various other attacks and probes
- Bad SSIDs controlled by an ESSID blacklist
- MAC address spoofing
- Ad hoc networks
- Other standard and nonstandard wireless threats

However, there are some drawbacks to using a WIDS. The technology is relatively new, so there may be undiscovered vulnerabilities. Also, the cost of implementing a WIDS can be high.

IDS Testing Tools

IDS Informer

Blade Software's IDS Informer tests the efficiency of an IDS in a lab or production environment. It subjects itself to various attacks that an IDS should detect, without interrupting the system's normal operations.

IDS Informer is shown in Figure 4-4, and its features include the following:

- Replays network traffic to check the compatibility of security policies without posing any real threat to production servers
- Customizable, with options such as rate of transmission (per attack and per packet), packet timeout, and packet expiration
- Spoofs any source or destination IP address and port combination
- Spoofs source or destination MAC addresses
- Guarantees packet delivery
- Controls packet expiration, timeout, and retries
- Creates a realistic, virtual attack that does not affect network traffic

Evasion Gateway

Evasion Gateway attempts known evasion techniques to circumvent firewalls, routers, and intrusion detection systems. It uses these techniques to probe for a wide range of host-based vulnerabilities and validate network requirements, and then generates clear and concise results from these tests.

Evasion Gateway is shown in Figure 4-5, and its features include the following:

- Bidirectional network-based evasion
- Fragmentation
- HTTP evasion

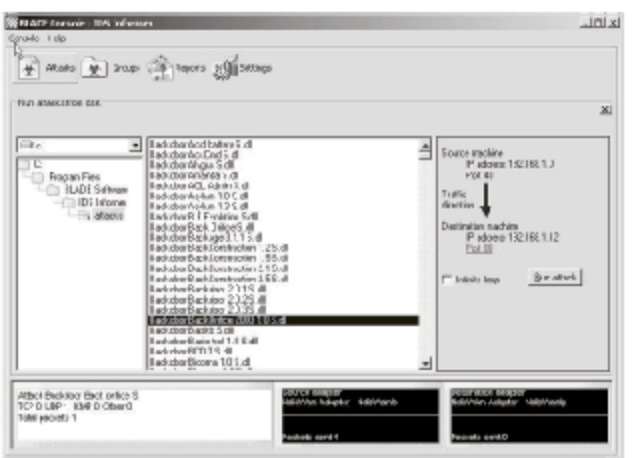


Figure 4-4 IDS Informer tests an IDS without interrupting normal system operations.

Figure 4-5 Evasion Gateway uses known evasion techniques to test IDS vulnerabilities.

- URL encoding
- Random URL encoding (non-UTF-8, random hex encoding)

Firewall Informer

Firewall Informer, also from Blade Software, tests the configuration and performance of any packet-filtering device, including firewalls, routers, switches, and gateways. It uses a technology called SAFE (Simulated Attack For Evaluation) to actively, yet safely, test a security infrastructure with real-world exploits.

Firewall Informer is shown in Figure 4-6, and its features include the following:

- Sends and receives packets without the need for protocols to be bound to the NIC
- Allows customization of rate of transmission (per attack or per packet), packet timeout, and packet expiration values
- Retransmits stateful attacks between two unique hosts from one PC
- Spoofs any source or destination IP address and port combination
- Spoofs any source or destination MAC address
- Guarantees packet delivery
- Controls packet expiration, timeout, and retries

Traffic IQ Professional

Traffic IQ Professional generates both standard application traffic and attack traffic between two virtual machines in order to test security devices. It is shown in Figure 4-7 and can test any nonproxy packet-filtering device, including the following:

- Application-layer firewalls
- Intrusion detection systems
- Intrusion prevention systems
- Routers and switches

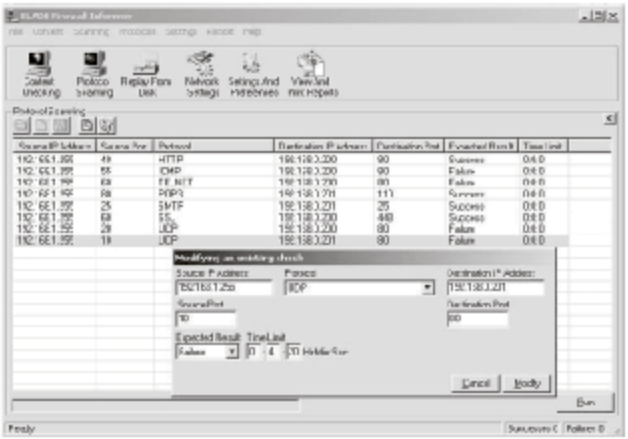


Figure 4-6 Firewall Informer tests any packet-filtering device.

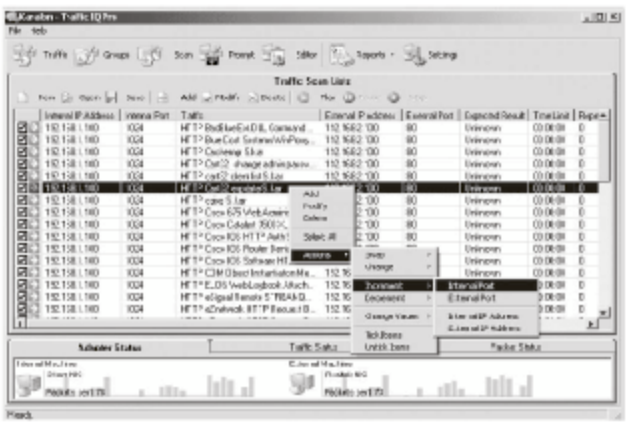


Figure 4-7 Traffic IQ Professional generates traffic between two virtual machines to test security systems.

OSSEC HIDS

OSSEC is a scalable, multiplatform, and open-source host-based IDS. It performs log analysis, integrity checking, Windows registry monitoring, rootkit detection, real-time alerting, and active response, monitoring one system or multiple systems simultaneously. OSSEC HIDS is shown in Figure 4-8.

Figure 4-8 OSSEC HIDS can monitor multiple systems at once.

Techniques Used to Evade Intrusion Detection Systems

Attackers can use several techniques to get around IDS, including the following:

- Some IDS use a pattern-matching approach in which a database of known exploits is matched against live traffic. If an attacker even slightly modifies a known attack, the IDS may not catch the attack.
- In Unicode, every character has several different representations. Using alternate representations of the same character can fool signature-matching IDS techniques.
- Many organizations use a central logging server to accumulate IDS alert logs. If an attacker can discover that server's IP address, he or she could initiate a DoS (denial of service) attack and render it useless.
- A target system can be flooded with packets specifically designed to trigger IDS alerts, which will either cause a DoS or make it difficult to discover the actual intrusion among all of the false positives.
- Flooding the network with any traffic can overload the IDS.

IDS Penetration Testing Steps

An IDS should be able to catch several well-known exploits. A tester can use these steps to attempt these exploits and ensure that the IDS recognizes them.

Step 1: Resource Exhaustion

IDS are prone to resource exhaustion attacks because, like any other system, they have memory, CPU, and bandwidth limitations. The IDS's performance will degrade or the IDS will fail if these resources get exhausted. The tester can test the effects of resource exhaustion by sending a large amount of traffic to the IDS.

Step 2: ARP Flood

A switch routes traffic in a passive manner. It maintains a cache of ARP responses that it sees passing through the network, and then routes traffic to the port that originated the ARP response matching the traffic's destination address. The switch temporarily enters hub mode and broadcasts the data packet to all ports when it

encounters a packet with an unknown destination address. The host that responds is recorded in the ARP cache. When this happens, attackers can see it, so the tester should try sending a flood of ARP packets to see how the switch reacts.

Step 3: MAC Spoofing

If two pieces of hardware on a network have the same MAC address, traffic on the network will be disrupted. Network switches have only a limited ability to internally map MAC addresses to physical ports. By flooding a switch with packets containing different source MAC addresses, the switch will enter failopen mode, in which all incoming packets are broadcast on all ports. This allows the attacker to view all network traffic routed through the affected switch.

Step 4: IP Spoofing

A tester can spoof the source IP address to flood the IDS, and analyze the responses. IP spoofing takes place when the tester uses an IP address that is not assigned to him or her to try to masquerade as an authorized user.

Step 5: Send a Packet to the Broadcast Address

The effect of spoofed packets can be amplified when sent to a broadcast address. The broadcast address present in the subnet is the highest address, perhaps 192.168.1.255 or 192.168.0.255. If a packet is sent to this address, it is delivered to all hosts on the network, and all of these hosts will respond to it.

Step 6: Inconsistent Packets

An IP header's packet length field is 16 bits, so a packet can only be 65,535 bytes. The IHL (IP Header Length) field gives the size of the header in 32-bit words. The minimum value is 5, as the minimum size of an IP header that contains all the correct fields is 160 bits, or 20 bytes (5×32 bits $= 160$). This indicates exactly where the payload data begin. So, it can be expected that the data portion of the packet is the difference between the values in these two fields.

Step 7: IP Packet Fragmentation

If a packet is too large, it may be split into multiple fragments. These fragments are stored and then reassembled when they are all present. The TTL value of a fragment must be greater than one for it to pass through a router. Hosts must accept packets with a minimum length of 68 bytes. The minimum size of the IP header is 20 bytes, and the last fragment could be as small as one byte if the previous unfragmented packet size was one greater than a multiple of eight.

Step 8: Duplicate Fragments

If duplicate fragments are received with different content, the penetration tester should check which fragment is saved.

Step 9: Overlapping Fragments

If a fragment is received whose contents partially overlap an already received fragment, the tester should check if the new packet's contents overwrite the original packet's contents. This type of denial of service (DoS) is known as a Teardrop attack.

Step 10: Ping of Death

The tester should test what happens if a fragmented ping packet is sent with a total packet length greater than the maximum of 65,535 bytes, to see if the user's operating system is still vulnerable to this attack.

Step 11: Odd-Sized Packets

Tests must be conducted to detect odd-sized packets. Packets are fragmented in multiples of eight, so an odd-sized packet has a length that is not a multiple of eight.

Step 12: TTL Evasion

Malicious hosts can use a combination of retransmission and TTL manipulation to fool an IDS into believing that it has seen the traffic that a host has seen, even if it has not.

Step 13: Send a Packet to Port 0

Port 0 is an officially reserved port in both TCP and UDP, and cannot be used for any type of network communications. Any traffic related to port 0 is probably not legitimate.

Step 14: UDP Checksum

If the 16-bit UDP checksum field is equal to 0, it means that the UDP checksum was not computed on transmission, so it should not be checked upon reception. This option only exists because much smaller systems may have saved CPU time by skipping the checksum process, but today, the processing power required to create a UDP checksum is trivial. The checksum is the only way to detect whether or not the packet is corrupted in transit, so any packets that have the UDP checksum turned off are questionable and may be subtle evasion attempts.

Step 15: TCP Retransmissions

Because IP is unreliable by nature, TCP retransmits packets in order to ensure that they reach their destination. If the IDS sees a retransmitted packet with correct checksums and different contents than the original packet, it is either a buggy TCP/IP implementation or a malicious attack.

Step 16: TCP Flag Manipulation

Different TCP stacks respond to illegal TCP flags in various ways. Sometimes, these illegal inputs will lead to a crash. Attackers can monitor the way the TCP stack responds to illegal TCP flags and determine the operating system with the help of programs like Nmap or Queso.

Step 17: TCP Flags

Testers should ensure that the following combinations of TCP flags are handled appropriately:

- *None*: If a packet has no flags set, it is invalid, because neither session initiation (SYN), session termination (FIN), nor termination (FIN/RST) is set.
- *SYN/FIN*: This indicates both session initiation (SYN) and session termination (FIN), which is an impossible condition.
- *SYN/RST*: This indicates both session initiation (SYN) and session termination (RST), which is an impossible condition.
- *SYN/FIN/ACK*: This indicates session initiation (SYN), session termination (FIN), and midstream (ACK), which is an impossible condition.
- *SYN/RST/ACK*: This indicates session initiation (SYN), session termination (RST), and midstream (ACK), which is an impossible condition.
- *All flags*: This is sometimes called the Xmas Tree flag combination. It combines the session initiation, midstream, and session termination flags along with the PSH (deliver data to application) flag and the URG (urgent data) flag, which is invalid.

Step 18: SYN Floods

Many TCP implementations are vulnerable to the SYN flooding resource-exhaustion attack, in which an attacker makes excessive session creation requests. If these SYN packets are spoofed from addresses that do not exist, no response packet containing SYN/ACK will be received, and the pending connection queue will expand until the memory is full.

Step 19: Initial Sequence Number Prediction

When large amounts of data are sent, they are fragmented at the network layer into IP packets, each of which has a sequence number to facilitate reassembly at the destination. Many TCP servers have easily predictable sequence numbers. The success of spoofing TCP connections for a man-in-the-middle (MITM) or session hijacking attack, in these cases, would be predicated on the ease of predicting the initial sequence number used by the target host.

Step 20: Backscatter

When an attacker floods a target with SYN packets, the target will then send an equal number of SYN/ACK packets to the source IP address. If this source IP address is spoofed, then the actual user of the falsified IP address will be flooded with unsolicited SYN/ACK packets. These unsolicited packets are known as backscatter.

Backscatter can cause a DoS, and if the attacker spoofs the same source IP to multiple targets, it will cause a more effective DDoS. If a system is flooded with multiple SYN/ACK packets without sending a SYN packet, it can be assumed that a backscatter attack is taking place.

Step 21: ICMP Packets

ICMP packets represent ping messages and error messages. ICMP packet flooding is the most common type of DoS attack. In this attack, the source address of an ICMP packet is spoofed, so the target system is flooded with unsolicited response packets, preventing legitimate access.

Step 22: Covert Channels

A covert channel is a hidden communications mechanism, used by attackers to hide their activities and maintain communications with the system. This can include techniques such as keyloggers or backdoors.

Step 23: Tcpreplay

Tcpreplay is a collection of BSD-licensed tools that can exploit previously captured traffic in order to perform tests on network devices. These tools replay traffic saved in files created by Tcpdump. Excessively replaying traffic can increase network traffic to the point where network performance degrades, which makes it easier for an attacker to compromise the network. Tcpreplay impersonates real traffic. However, it cannot replay all types of traffic, such as retransmission and congestion control traffic, and it does not support all networks and operational environments.

Step 24: TCPopera

The TCPopera tool works like Tcpreplay, except it allows the user to define specific, theoretical network conditions and simulate traffic in a realistic environment in which packets may be delayed or lost. This provides IDS testing environments with traffic that accurately displays TCP performance, helps determine whether the IDS tracks the TCP connection state, and shows how the IDS handles retransmitted packets. The primary disadvantage of TCPopera is that it is complex. It can take a good amount of time and skill to use.

Step 25: Method Matching

Many IDS fail because they expect attackers to use the GET method during attacks. For example, the IDS may be looking for the following:

```
GET /cgi-bin/some.cgi
```

However, by simply replacing it with *HEAD*, an attacker can fool the IDS:

```
HEAD /cgi-bin/some.cgi
```

Step 26: URL Encoding

URL encoding involves replacing the characters in a URL with their escaped equivalents. HTTP allows characters to be represented in %*xx* notation, where *xx* indicates the character's hex value. For instance, %*20* represents a space, and %*63* represents a *c*. An attacker could replace the phrase *cgi-bin* with %63%67%69%2d%62%69%6e, and an IDS specifically looking for *cgi-bin* may miss it.

Step 27: Double Slashes

HTTP allows any number of slashes to function as a single slash. For example, */cgi-bin/some.cgi* is functionally equivalent to *//cgi-bin//some.cgi*. Still, adding extra slashes can fool an outdated IDS that is simply watching specifically for */cgi-bin/some.cgi*. Newer IDS will automatically combine multiple slashes into one before evaluating the content.

Step 28: Reverse Traversal

Similar to the double-slash technique, reverse traversal adds extra information to the request to fool the IDS. An HTTP request containing two dots will instruct the recipient to go up one level in the request. For instance, in the following example, the two dots are instructing the recipient to go up one level from the *abcd* directory:

```
GET /cgi-bin/abcd/../some.cgi HTTP/1.0
```

This effectively tells it to ignore the *abcd* portion of the request, making the request equivalent to the following example:

```
GET /cgi-bin/some.cgi HTTP/1.0
```

Step 29: Self-Referencing Directories

While two dots indicate that the request should go up one directory level, one dot indicates that it should remain at the current level. These two examples are functionally equivalent to one another:

```
GET /cgi-bin/./././some.cgi HTTP/1.0
```

```
GET /cgi-bin/some.cgi HTTP/1.0
```

Step 30: Premature Request Ending

Many IDS stop scanning a request after *HTTP/1.0\r\n* in order to save time and processing power. This means that an IDS looking for *cgi-bin/some.cgi* may miss its target in the following string:

```
GET / HTTP/1.0\r\nHeader: /../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n
```

However, if the IDS scans the input before decoding any encoded characters, it will catch this:

```
GET /%20HTTP/1.0%0d%0aHeader:%20/../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n
```

Step 31: IDS Parameter Hiding

Some IDS do not scan the parameters submitted with dynamic content. These parameters follow a question mark in the request string, so an affected IDS will stop scanning once the question mark is reached. This means that an IDS looking for *cgi-bin/some.cgi* may miss its target in the following string:

```
GET /index.htm?param=/../cgi-bin/some.cgi HTTP/1.0
```

However, if the IDS scans the input before decoding any encoded characters, it will catch this:

```
GET /index.htm%3fparam=/.../cgi-bin/some.cgi HTTP/1.0
```

Step 32: HTTP Misformatting

HTTP calls for spaces separating the method, the URL, and the HTTP version. This makes it easy to extract a URL; the program needs only to look for the spaces to separate the parameters.

This can make it easier to reduce false positives. When configuring an IDS to search for a URL ending in "/phf," an administrator can define the search string as "/phf ," with a trailing space. This way, it will look for the space after the string, indicating that it is at the end of a URL.

Step 33: Long URLs

Some IDS look only within the first several bytes of the request. This is usually effective, because the first line of the request includes the URL, but it can be exploited, as in the following example:

```
GET /rfprfp<lots of characters>rfprfp/../cgi-bin/some.cgi HTTP/1.0
```

If the specified directory, which is later ignored thanks to the */../*, includes an extremely large amount of random characters, the IDS may not be able to scan the entire line, and the offending request would pass through. Long URL attempts can quickly clutter Web server logs.

Step 34: DOS/Windows Directory Syntax

Microsoft uses backslashes (\) to separate directories, while UNIX and HTTP use forward slashes (/). Local Windows users can use either in their requests, so the following requests are equivalent:

```
GET /cgi-bin/some.cgi HTTP/1.0
GET /cgi-bin\some.cgi HTTP/1.0
```

However, an IDS looking specifically for */cgi-bin/some.cgi* may miss it, so this must be tested.

Step 35: Null Method Processing

The null character is used to indicate the end of a string in most C string libraries. Many IDS use these libraries, so they will stop scanning at a null character and miss */cgi-bin/some.cgi* in the following example:

```
GET%00 /cgi-bin/some.cgi HTTP/1.0
```

Step 36: Case Sensitivity

In UNIX systems, filenames are case sensitive; *index.html* and *INDEX.HTML* are different files. However, in DOS/Windows systems, files are case insensitive, so *index.html* and *INDEX.HTML* point to the same file. This means that, for local Windows users, the following examples are equivalent:

```
GET /cgi-bin/some.cgi HTTP/1.0
GET /CGI-BIN/SOME.CGI HTTP/1.0
```

Step 37: Session Slicing

Some IDS check single packets at a time, so they will miss a target string if it is split into multiple packets. This technique is called session slicing. For example, the request *GET /cgi-bin/some.cgi HTTP/1.0* can be split into the following packets:

```
GE
T /
cg
i-bi
n/som
e.c
gi H
TTP/1.0
```

However, if the IDS checks the request after the packets are reassembled, then it will catch any target strings.

Chapter Summary

- The main feature of an intrusion detection system (IDS) is to monitor network activity on a network or workstations and generate alerts when there may be an intrusion.
- A host-based IDS, or HIDS, is focused on analyzing the behavior of individual systems.
- A network IDS (NIDS) checks every packet entering the network for any unexpected or incorrect data.
- An application-based IDS can identify several of an attacker's suspected activities, but only for one specific application.
- Several tools, such as IDS Informer, can be used to test the effectiveness of an IDS.
- The session-slicing technique divides a payload over multiple packets to avoid simple pattern matching.

Chapter 5

Physical Security and Stolen Laptop, PDA, and Cell Phone Penetration Testing

Objectives

After completing this chapter, you should be able to:

- Map a company's physical perimeter
- Obtain information from outside a building
- Pick locks
- Test for electromagnetic interception
- Test a company's physical security policy
- Test for RFID
- Check for active network jacks
- Conduct social engineering
- Use fake IDs to gain entrance to a secure area
- Obtain information through dumpster diving
- Dress as a courier to obtain access to a facility
- Identify sensitive data in laptops, PDAs, and cell phones
- Find server information in portable devices
- Use browsers to find sensitive information

Key Terms

Lock picking the practice of unlocking a lock without using a valid key

Physical security the measures taken to protect personnel, critical assets, and systems against deliberate attacks and accidents

Piggybacking the act of following an authorized user through a secure entrance, as when a polite user opens and then holds the door for those following

Radio-frequency ID (RFID) tags tags that can be tracked through the use of radio waves

Shoulder surfing looking over someone's shoulder to see what that person is typing, writing, or doing

Introduction to Physical Security and Stolen Laptop, PDA, and Cell Phone Penetration Testing

Physical security describes the measures taken to protect personnel, critical assets, and systems against deliberate attacks and accidents. Physical security ensures that attackers cannot access a resource or information stored on media of the organization.

Cyber-security development focuses on mitigating attacks to computer networks as well as preventing physical attacks. Many companies spend a lot of time and resources protecting their network resources from attack over the Internet, while neglecting simple physical security measures such as locking doors. Firewalls cannot prevent the compromise of sensitive information if proper physical security is not present in an organization.

An important aspect of physical security is the security of laptops, cell phones, and PDAs. These devices carry sensitive data. Executives and mobile workers depend on these devices every day, so they represent an easily accessible source of information. The security of laptops and PDAs is left to the individual who possesses them, rather than the organization he or she works for. However, these devices contain information that could compromise the organization, so they represent a potential weak link in any company's security policy.

Steps in Conducting Physical Security Penetration Testing

The following steps can be used to find loopholes in the physical security of an organization:

- *Step 1*: Map the possible entrances.
- *Step 2*: Map the physical perimeter.
- *Step 3*: Penetrate locks used on the gates, doors, and closets.
- *Step 4*: View sensitive information from outside the building.
- *Step 5*: Penetrate server rooms, cabling, and wires.
- *Step 6*: Attempt lock-picking techniques.
- *Step 7*: Test fire detection systems.
- *Step 8*: Test air conditioning systems.
- *Step 9*: Attempt electromagnetic interception.
- *Step 10*: Test if the company has a physical security policy.
- *Step 11*: Enumerate physical assets.
- *Step 12*: Perform a risk test.
- *Step 13*: Check if any valuable paper documents are kept at the facilities.
- *Step 14*: Check how these documents are protected.
- *Step 15*: Check employee access policies.
- *Step 16*: Test for radio-frequency ID (RFID).
- *Step 17*: Check physical access to facilities.
- *Step 18*: Document processes for contractors.
- *Step 19*: Test people in the facility.
- *Step 20*: Determine who is authorized.
- *Step 21*: Test fire doors.
- *Step 22*: Check for active network jacks in meeting rooms.
- *Step 23*: Check for active network jacks in company lobby.
- *Step 24*: Check for sensitive information left in meeting rooms.
- *Step 25*: Check for a receptionist or guard leaving lobby.
- *Step 26*: Check for accessible printers in the lobby and print a test page.
- *Step 27*: Obtain phone/personnel listing from the lobby receptionist.

- *Step 28*: Listen to employee conversations in communal areas/cafeteria.
- *Step 29*: Check ceiling space access.
- *Step 30*: Check windows/doors for visible alarm sensors.
- *Step 31*: Check visible areas for sensitive information.
- *Step 32*: Try to shoulder-surf users logging on.
- *Step 33*: Try to videotape users logging on.
- *Step 34*: Check if exterior doors are guarded and monitored.
- *Step 35*: Check guard patrol routines for holes in the coverage.
- *Step 36*: Intercept and analyze guard communication.
- *Step 37*: Attempt piggybacking on guarded doors.
- *Step 38*: Attempt to use a fake ID to gain access.
- *Step 39*: Test "after office hours" entry methods.
- *Step 40*: Identify all unguarded entry points.
- *Step 41*: Check for unsecured doors.
- *Step 42*: Check for unsecured windows.
- *Step 43*: Attempt to bypass sensors configured on doors and windows.
- *Step 44*: Attempt dumpster diving.
- *Step 45*: Use binoculars from outside the building to view activities inside.
- *Step 46*: Use active high-frequency voice sensors to hear private conversations among company staff.
- *Step 47*: Dress up as a FedEx/UPS employee and try to gain access to the building.
- *Step 48*: Document everything.

Step 1: Map the Possible Entrances

An invader often tries to locate an unusual path to enter the target premises. The provided security may vary according to the structure of the building. The protection of a building always depends on the size and the premises. An invader may choose a way to intrude into a building where a security hole is present by the following means:

- *Through doors*: These are common ways to enter or exit any building. A big building may have many doors installed for convenience. The attacker will typically choose to enter through the least-used door.
- *Through windows*: Easily breakable windows can be used as an entry for an attacker to get into a building. Some server rooms have windows that can be accessed from outside the building.
- *Through fire exits*: These exits are also a security loophole that can come in handy for an invader to gain entrance to the building.

Step 2: Map the Physical Perimeter

Penetration testers should do a survey of the physical perimeter of the target. An in-depth study about the surroundings and the building will make the work easier and ensure a thorough search. The strength of the penetration testing plan depends on the thoroughness of this search, so the penetration testers must survey the strength of the premises security as well as the surveillance equipment and personnel deployed in the building.

The penetration tester should learn about the following security measures:

- *Doors*: What types of doors are used and what is their strength? What locks are used?
- *Windows*: Are the windows easily movable or breakable?
- *Roof strength*: What type of roof does the building have? What is its strength?
- *Basement*: How strong is the basement of the building?
- *Access policies*: What are the general access policies in the organization? What are the common visiting policies and what are their weaknesses?
- *Types of locks used*: Are the locks secured?

Step 3: Penetrate Locks Used on the Gates, Doors, and Closets

Locks are mainly used as security devices in places where access is controlled, such as storage containers, doors, gates, and windows. The safety of any facility and the items it contains relies greatly upon locking devices. Locks merely dissuade or delay access and should be supplemented with other protection devices when a proper balance of physical security is needed. Locks vary in appearance as well as function and application.

Two types of locks can be used:

1. *Mechanical locks*: These depend on the key shape that retracts secured bolts (combination and safe locks). They appear to be simple, but some of the technologies related to the development of these physical security devices are complex.
2. *Electromagnetic locks*: These devices accept different types of inputs including keys that are magnetic strips on ID cards, radio signals from name badges, personal identification numbers (PINs) input through a keypad, or a possible combination of these.

Step 4: View Sensitive Information from Outside the Building

Telephotography can be attempted from outside to photograph sensitive work done in an office through a window. Any documents that are at an angle greater than 15 degrees above the horizontal can be photographed through a window. Equipment used and conditions at the time determine the effective range. With the help of artificial light, net or opaque glass becomes transparent and a photograph can be easily taken from outside.

Step 5: Penetrate Server Rooms, Cabling, and Wires

The server is the most important part of any network, so usually it is given a higher level of security. System administrators have a habit of labeling the server and other systems in the server room. These labels have the operating system name and the hardware specification written on them. Gaining physical access to the server room allows an attacker to steal secret data, such Social Security numbers, credit card numbers, and business plans. Additionally, worms or Trojans could be planted that send passwords or other sensitive information to the attacker's e-mail address.

To test the security of the servers, a tester should try to reboot the system remotely. If DOS is disabled or removed from the servers, an intruder cannot boot the server remotely. The following vulnerabilities should be looked for:

- Can the server be rebooted from a floppy disk or CD-ROM?
- Is there a hardware- or software-based RAID system?
- Are there any surveillance cameras to monitor the movements in server rooms?

Step 6: Attempt Lock-Picking Techniques

Lock-picking is the practice of unlocking a lock without using a valid key. An attacker can pick a lock to get direct, unauthorized access to the system without raising any suspicion about his or her entry. Tools used for lock picking are called lock picks, and include simple tools such as hairpins, safety pins, and blank keys. The lock picks differ for each type of lock.

Lock picks are used to manipulate or operate the internal pins of a lock. They are inserted in the keyhole and rotated systematically to set the internal pins to the right position; thus, a lock can be released without using valid keys. Using an appropriate set of lock-picking tools, many locks can be picked. A tester can slip into the lock, as shown in Figure 5-1, by using a tension wrench and a safety pin. The front part of the safety pin is bent 90 degrees to allow the tester to hook the tumblers.

A tester can use the following steps to pick a lock:

1. Insert the safety pin into the lock. When it has reached the correct depth, slowly raise the safety pin toward the last tumbler and hold it as shown in the figure. Insert the tension or torque wrench into the lock and rotate it. A gentle sound will be heard when the tumbler breaks.
2. Keeping the same pressure with the tension wrench, perform the steps explained in step 1 again on the next tumbler.

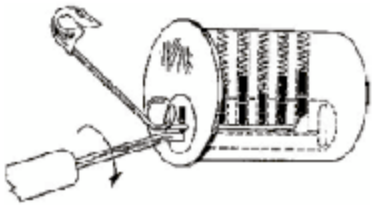


Figure 5-1 Locks can be picked to breach physical security.

Step 7: Test Fire Detection Systems

Fire detection systems are installed to mitigate vulnerabilities arising due to fire. Fire accidents can cause excessive damage to organizational infrastructure and human lives. Different types of fire detection systems can be installed according to requirements.

Installing fire detection systems ensures early detection of fire outbreaks. Security personnel should be prepared in case any fire alarm goes off. Security planners should visualize the situations that could occur if the fire alarm is triggered and be prepared for such situations. In the case of a fire alarm going off, there could be chaos, as people will try to evacuate in a panic. These situations may cause more damage to human lives. Security planners should have a proper plan to overcome such situations.

An attacker can take advantage of the panic created after a fire alert to steal computers and laptops containing valuable information. If the security of the organization is not prepared for such scenarios, valuable equipment and information can be stolen or destroyed.

Penetration testers should check the fire alarm system policies and procedures within the company. Fire alarm system policies and procedures provide important information about the organization. For example, they provide an outline of the fire vulnerabilities and security systems of the organization.

Step 8: Test Air Conditioning Systems

Testers should check the air conditioning systems for possible penetration attempts. Air conditioning systems of large organizations use wide ducts, which attackers can use to physically sneak into an organization and steal sensitive information.

Testers can investigate the air conditioning ducts and check for ways of hiding information devices. Air conditioning ducts should not lead to rooms that hold important systems and information. These ducts should be regularly checked and protected by grills wherever possible. Separating AC systems from computer rooms and restricted areas within the rest of an organization is a better practice. This allows the AC systems of the computer rooms to be monitored more efficiently. The openings of the AC ducts can also be designed to make them difficult to access.

Step 9: Attempt Electromagnetic Interception

An organization's data transmissions are often vulnerable to interception because they occur through the use of electromagnetic waves. An attacker can covertly observe and gain sensitive data using an antenna and a receiver. In these scenarios, users are not aware of the interception so they cannot take measures to stop it. This could inflict serious damage on the organization.

In this step, testers can bug a telephone line inside the building and see if the signals can be picked up from outside the building using frequency receivers. Wiretapping bugs can be fixed anywhere on the wire or telephone handle. These bugs capture the communications and transmit them through a radio transmitter. The attacker receives these transmissions with a radio receiver, decodes the signals, and listens to or monitors the conversations or communications.

Data encryption can be very effective in protecting the information from being intercepted. Secure communication protocols such as WPA are used for the protection of information while in transmission.

Step 10: Test If the Company Has a Physical Security Policy

Penetration testers should determine if the company maintains a physical security policy. The following elements should be clearly defined in the policy:

- Physical security programs to ensure protection of personnel and assets
- Responsibilities of a physical security officer
- External and internal access controls
- Restricted area access
- Identification card policy
- Visitor policy
- Fire policy
- Disposal policy

Step 11: Enumerate Physical Assets

The testing team should prepare a list of physical assets. The list should include all the physical assets that the organization owns, including the following:

- Company buildings
- Building perimeter and surroundings
- Access control devices
- CCTVs
- Air conditioners and ducts
- Computer equipment
- Network devices
- Information storage devices
- Wireless devices
- Communication wires
- Fax and photostat machines
- Fire extinguishers

Step 12: Perform a Risk Test

The risk associated with physical security largely depends on the value of items inside the facility. In some organizations, especially in federal offices, it can be easy to walk into the office, posing as an employee.

Testers should identify the areas that are prone to attack by attempting the following intrusions:

- Try to intrude into an office to exploit entrance and exit policy.
- Try to steal a file or computer network equipment.
- Check the potential for damaging any equipment.
- Check whether the equipment inside the facility is locked down when it is unused.
- Try to pick locks with weak security mechanisms.

Testers should estimate the risk of intrusion through the following procedures:

- Identify and evaluate threats.
- Evaluate the existing security mechanisms.
- Determine the remaining possible risk.
- Identify appropriate additional measures to check the risk.

Step 13: Check If Any Valuable Paper Documents Are Kept at the Facility

Testers should attempt to penetrate an organization's facilities and search for documents that relate to the following topics:

- Finances
- Marketing strategy
- Future projects
- Stakeholders and personnel
- Meeting notes

They should try to collect documents (if the contract rules permit) that are vulnerable in the following ways:

- Easily accessible
- Left unsecured
- Not properly destroyed
- Not stored in a secured vault

Step 14: Check How These Documents Are Protected

Testers can analyze the procedures for protecting documents by checking for the following factors:

- Guidelines followed to protect the documents
- Physical access measures implemented to prevent unauthorized access to paper documents
- Disposal of documents with shredding machines
- Impact on the company if unauthorized individuals access the documents
- Security mechanisms implemented to protect the areas that store sensitive documents
- Responsible authorities deployed to protect sensitive records, files, and documents
- Levels of clearance required for accessing the documents
- Alarm systems alerting personnel of theft and destruction
- Transfer of documents to authorized persons

Step 15: Check Employee Access Policies

The testing team can check employee access policies by doing the following:

- Observe the personnel of various departments visiting facilities within the organization
- Analyze their visitation requirements
- Identify sensitive facilities
- Identify the employees who must use those facilities frequently
- Check the restricted-area access policy
- Verify that measures implemented to control access through the use of biometric devices work
- Check the physical security measures that are related to personnel security
- Install CCTVs at critical areas to monitor actions of individuals
- Maintain log books of entries into critical areas

Step 16: Test for Radio-Frequency ID (RFID) Tags

Radio-frequency ID (RFID) tags are tags that can be tracked through the use of radio waves. These tags are often used to secure portable resources so that they can be retrieved in case of theft. The RFDump tool can be used to identify RFID tags. If the RFID is encrypted, testers can try to break the encryption. If access to the RFID tag is gained, the tag's data can be modified.

Step 17: Check Physical Access to Facilities

Penetration testers need to check how physical access to facilities is controlled for the following personnel:

- Employees
- Contractors (stakeholders)
- Trainees
- Visitors

The testers need to identify the physical security measures implemented for internal and external access.

Step 18: Document Processes for Contractors

Testers should check whether there is a documented process implemented for contractors. If such a process exists, the testers should analyze the entire process and determine if the following information is recorded before permitting a contractor to visit:

- Contractor's details
- Purpose of visit
- Devices carried inside the facility
- Whether the list of rules the contractor must follow is mentioned on the back of the permission document
- Previous records of documented processes
- The level of authority of the permission-issuing officer
- Whether all developments in the process are properly documented for future reference

Step 19: Test People in the Facility

The testing team should check for the following procedures regarding personnel:

- Determine whether people in the facility possess identification badges that identify them while on duty. If there are identification badges, try to forge one.
- Test whether the identification policy covers the following personnel:
  - Employees (both temporary and permanent)
  - Contractors
  - Visitors
  - Security personnel
- Test whether people are carrying any suspicious parcels with them in and out of the facility.
- Try to intrude into a restricted area with the following media:
  - USB devices
  - Camera
  - iPod
  - Tapping devices
  - Phones

Step 20: Determine Who Is Authorized

Does a current list of individuals who are authorized to physically access the facilities exist? Is this list periodically reviewed and purged so that any inactive or terminated personnel's access is removed? Testers can utilize the following procedures to assess authorization policy:

- List the personnel who have authority over facility visitation. Access to the different areas of the organization should be categorized according to the authorization of visitors. Employees of the organization could be provided with identity badges and uniforms. Security personnel should accompany outside visitors during their visit to the organization.

- Check whether all people on the list are still associated with the organization.
- Find out the typical time gap between each review of the authorization list. The list of authorized people should be regularly updated so that any change in authorization can be reflected as early as possible.

Step 21: Test Fire Doors

Penetration testing teams should test fire doors periodically to ensure that the alarms work properly. Regular monitoring of the fire alarm system and a planned mock test of emergency planning ensures a better response in case of any real emergency.

Unauthorized individuals can gain access to facilities without anyone noticing and can cause damage, steal, or disrupt operations. Fire doors are generally overlooked for maintenance, as they are rarely used and are generally in secluded areas of the organization. This makes them a convenient entry point for an attacker.

To avoid such incidents, fire doors should be taken care of properly and checked at regular intervals. Proper lighting should be provided to reveal any intrusion attempts.

Step 22: Check for Active Network Jacks in Meeting Rooms

The testing team should do the following to check for active network jacks:

- Try to attach a wireless access point to access the network from outside.
- Make sure to use an AP device that is not easy for anyone in the organization to detect.
- Make sure to mark the jack with a specific code. This will help prove that the physical security was breached.
- Identify the active network jacks that are not in use and secure them.

Step 23: Check for Active Network Jacks in Company Lobby

The testing team should do the following to check for active network jacks:

- Check for active network jacks in the company lobby.
- Check if the active network jacks are monitored as to who is connecting to them.
- Check for active network jacks in common areas, where staff and visitors can connect to their laptops.

Step 24: Check for Sensitive Information Left in Meeting Rooms

Penetration testers should check the following:

- Check for papers/electronic media in conference rooms that are left after a meeting has ended. These papers or media may contain information about what was discussed in the meeting and the conclusions of the meeting.
- Check for notes or details in the meeting rooms. They can provide an attacker with an overview of important decisions in the office.

Step 25: Check for a Receptionist or Guard Leaving Lobby

Testers should check the following:

- Check for a receptionist or guard leaving the lobby.
- Look for people loitering in the reception area. They might be looking for a chance to acquire details in the absence of a guard.
- Note the timing of absences of the receptionist/guard.

Step 26: Check for Accessible Printers in the Lobby and Print a Test Page

The testing team should do the following to check for printer access in the lobby:

- Check for printers in the lobby.
- Print a test page.
- Secure the test page as evidence.

Step 27: Obtain Phone/Personnel Listings from the Lobby Receptionist

Testers should do the following to try to get information:

- Obtain phone/personnel listings from the lobby receptionist. The staff can be socially engineered for sensitive information, checking to see if they reveal sensitive credentials.
- Obtain the phone extension numbers of the employees from the receptionist. These will enable an intruder to talk directly to an employee for social engineering purposes.

Step 28: Listen to Employee Conversations in Communal Areas/Cafeteria

Testers should do the following to try to get information:

- Listen to employee conversations in communal areas/cafeteria.
- Try to note the latest projects going on within the company and the names of key personnel involved in the projects.

Step 29: Check for Ceiling Space Access

Penetration testers should check that the ceiling is secure and is not vulnerable to break-in attempts by attackers/thieves. Secure rooms can often be accessed through the ceiling.

Step 30: Check Windows/Doors for Visible Alarm Sensors

Testers should check for the following factors to ensure that the security of windows/doors is intact:

- Check windows/doors for visible alarm sensors.
- See if the alarms are working.
- See if the windows/doors allow a place for a thief to hide.

Step 31: Check Visible Areas for Sensitive Information

Testers should check visible areas for sensitive information such as accounts and passwords written on whiteboards or pasted on monitors. People tend to forget passwords and write them on papers in visible areas. Such data can be used to crack their passwords.

Step 32: Try to Shoulder-Surf Users Logging On

Shoulder surfing is looking over someone's shoulder to see what that person is typing, writing, or doing. This can be accomplished by walking along with a target. The testing team should attempt to view the passwords or usernames of these personnel as they log on.

Step 33: Try to Videotape Users Logging On

In this step, testers do the following:

- Try to arrange video cameras in places where people work in such a way that the computer monitor and keyboard are in focus.
- Recording keystrokes can reveal information if the screen is not accessible.

Step 34: Check If Exterior Doors Are Guarded and Monitored

Check if access is restricted to visitors.

Step 35: Check Guard Patrol Routines for Holes in the Coverage

Testers can identify the times that certain places are left unguarded.

Step 36: Intercept and Analyze Guard Communications

Testers can try to monitor conversations for sensitive information.

Step 37: Attempt Piggybacking on Guarded Doors

Piggybacking is the act of following an authorized user through a secure entrance, as when a polite user opens and then holds the door for those following. A tester can attempt to closely follow employees into the building without having to show valid credentials. He or she can then try to get into restricted areas by pretending to be an authorized person.

Step 38: Attempt to Use a Fake ID to Gain Access

It is easy to create an ID that looks authentic. These IDs can be either state- or company-issued IDs. Testers can create different fake IDs to see if access is granted.

Step 39: Test "After Office Hours" Entry Methods

The testing team should check for the following common activities:

- Identify employees entering the company premises after office hours. Check whether they follow normal office hours. The entry register will help evaluate this.
- Determine whether the employees swipe their access cards before gaining entry to the office premises.

Step 40: Identify All Unguarded Entry Points

Testers should identify all unguarded entry points, including doors, windows, the cafeteria, and so on, and then check the locks to see if they can be picked.

Step 41: Check for Unsecured Doors

Penetration testers should do the following:

- Identify if doors are locked properly.
- Check if entry is possible through exit doors.

Step 42: Check for Unsecured Windows

Penetration testers should do the following:

- Check if entry is possible through windows
- Check if they can view or access things in the rooms through windows
- Test that the locks of windows are functioning properly

Step 43: Attempt to Bypass Sensors Configured on Doors and Windows

Testers can see if the sensors are properly attached and functioning, and then attempt to bypass these sensors by detaching them or by using other means.

Step 44: Attempt Dumpster Diving

The testing team should attempt to retrieve any useful information from the company dumpster. This may include printed documents, books, manuals, CDs, floppy disks, invoices, and bank statements.

Step 45: Use Binoculars from Outside the Building to View Activities Inside

Testers should find an adjacent building or parking lot from which the activities within the company building can be viewed. Through the use of high-powered binoculars, sensitive information can be uncovered.

Step 46: Use Active High-Frequency Voice Sensors to Hear Private Conversations Among Company Staff

In this step, penetration testers do the following:

- Identify locations where staff members usually converse in private.
- Place active high-frequency voice sensors in such places to hear private conversations among company staff.

Step 47: Dress Up as a FedEx/UPS Employee and Try to Gain Access to the Building

Employees trust courier companies and usually allow them inside the building. Security guards can be bypassed by dressing up like a courier carrying a package.

Step 48: Document Everything

The testing team should document all findings. The following information should be included in the report:

- Locks used on gates, doors, and closets
- Server rooms, cabling, and wires
- Fire detection and air conditioning systems
- Electromagnetic interception and physical assets
- Physical security policy and risk test
- Valuable documents in the facilities
- Employee access and physical access to facilities
- Documented processes and authorized people

Laptop, PDA, and Cell Phone Theft

If a laptop, PDA, or cell phone is stolen, the following information could be gained:

- *Strategic information*: Examples of this type of information include pending mergers, new product intellectual property, strategies and launch plans, and previously undisclosed financial operating results.
- *Tactical information*: Examples include private compensation information, plans for organizational changes, proposals to clients, and myriads of similar information that can be gained from reading a person's e-mail, calendar, contacts, or collection of documents and spreadsheets.
- *Network or computing infrastructure information*: Examples of this type of information include usernames and passwords, dial-in numbers, IP addressing schemes, DNS naming conventions, ISPs used, primary mail servers, and other networking details related to connecting the laptop to the corporate or Internet environment.
- *Personal information*: This includes credit card numbers, passwords stored in text files, Social Security numbers, birth dates, anniversary dates, family names, personal photographs, private e-mails, and so on.

Laptop, PDA, and Cell Phone Penetration Testing Steps

- *Step 1*: Identify sensitive data on the devices.
- *Step 2*: Look for passwords.
- *Step 3*: Look for company infrastructure or finance documents.
- *Step 4*: Extract the address book and phone numbers.
- *Step 5*: Extract schedules and appointments.
- *Step 6*: Extract applications installed on these devices.
- *Step 7*: Extract e-mail messages from these devices.
- *Step 8*: Gain access to server resources by using extracted information.
- *Step 9*: Attempt social engineering with the extracted information.
- *Step 10*: Check for BIOS password.
- *Step 11*: Look into encrypted files.
- *Step 12*: Check Web browsers.
- *Step 13*: Attempt to enable wireless services.

Step 1: Identify Sensitive Data on the Devices

Mobile devices such as laptops and PDAs may contain various types of information, including the following:

- *Company finance documents*: These documents may contain information about bank accounts, capital resources, investment policies, budgetary data, etc.
- *Excel spreadsheets*: These documents show information about various presentations, estimations of project plans, scheduling details, program details, etc.
- *Word documents*: These documents contain information in the form of official letters, plan documents, price quotes, vendor lists, etc.
- *E-mail messages*: These can be a rich source of personal and sensitive information.
- *Operational plans*: An operational plan provides information about project scheduling and the various functions and operations involved in the project.

Step 2: Look for Passwords

Testers should look for the following passwords:

- VNC passwords
- E-mail account passwords
- Active Directory passwords
- Web site history passwords
- Passwords stored in the registry
- SSH/telnet passwords
- Application passwords

Step 3: Look for Company Infrastructure or Finance Documents

Sometimes, a laptop might contain the following company infrastructure documents:

- Building plans
- Plan of operations
- Overseas operations and procedures
- Company handbooks or manuals
- Contracts and agreements
- NDA documents
- Bank statements
- Auditing information
- Insurance documents

Step 4: Extract the Address Book and Phone Numbers

PDAs and laptops contain address books that could contain the following useful information:

- Names
- Addresses
- Telephone numbers
- Fax numbers
- E-mail addresses
- Birth dates
- Notes
- Pictures

Step 5: Extract Schedules and Appointments

The following schedule and appointment information may be stored on a PDA or laptop:

- Times and dates of meetings
- Attendees
- Locations
- Agendas
- Meeting confirmations
- Meeting lengths

Step 6: Extract Applications Installed on These Devices

Various applications can be installed on mobile devices. These applications can reveal sensitive data. For example, an ERP (enterprise resource planning) application could reveal all the planning and coordinating strategies of an enterprise or could even reveal confidential information regarding any product of that enterprise. Such applications, if accessed, could provide the most sensitive information of an organization or an enterprise.

In this step, penetration testers trace all the applications installed on a particular device and then look for the data in those installed applications. The best example of an application that could provide information is finance software, such as Quicken and Microsoft Money. Finance software contains details regarding financial aspects of the organization or individual. This includes the budget strategy for the year or for a particular project, the sources of income for the company, expenditures, salary details, bank balances, company credits, debits, shares, quotations, profits, and losses.

Step 7: Extract E-Mail Messages from These Devices

E-mail messages can provide a lot of sensitive information. Passwords and access codes can be uncovered if e-mail content is scanned thoroughly.

Step 8: Gain Access to Server Resources by Using Extracted Information

In this step, the penetration testers gather all the information obtained in previous steps and use that information to try to gain access to server resources. For example, if the testers discover a password for accessing a server through e-mail scanning, they can gain access to that server. The testers may also find other information stored on the server that will allow them to further penetrate the network. Once an attacker is able to gain access to a server, he or she can share, copy, destroy, or modify those resources, making this a very important step in the penetration testing process.

Step 9: Attempt Social Engineering with the Extracted Information

Extracted information such as names, phone numbers, or e-mail addresses could be used for social engineering.

Step 10: Check for BIOS Password

Testers should determine if the BIOS password is enabled. If it is not, an attacker could easily alter the BIOS settings. The testers should also check whether the BIOS lists the hard disk as the first bootable device.

Step 11: Look into Encrypted Files

Encrypted files can be decrypted through the use of cryptography tools, and testers may find useful information in these files.

Step 12: Check Web Browsers

Testers should check the following items in the Web browser for information:

- Cookies
- History file
- Temporary files
- Password file

Step 13: Attempt to Enable Wireless Services

In this step, testers perform the following procedure to attempt to enable wireless services:

- Switch on wireless or Bluetooth near the company campus.
- Scan for the LAN network of the company.
- Locate the LAN network and search for the SSID on the laptop.
- Check whether the network requires a password for connection.
- Check the password strength and try to break it using password-cracking techniques.

Chapter Summary

- Many companies spend a large amount of time and resources protecting their network resources from attack over the Internet, while neglecting simple physical security measures such as locking doors.
- The safety of any facility and the items it contains relies greatly upon locking devices.
- The server is the most important part of any network, so usually it is given a higher level of security.
- Air conditioning systems of large organizations use wide ducts, which attackers can use to sneak into an organization and steal sensitive information.
- Data encryption can be very effective in protecting information from being intercepted. Secure communication protocols such as WPA are used for the protection of information while in transmission.
- Regular monitoring of the fire alarm system and a planned mock test of emergency planning ensures a better response in case of any real emergency.
- An ERP application could reveal all the planning and coordinating strategies of an enterprise or could even reveal confidential information regarding any product of that enterprise.

Chapter 6

E-Mail Security Penetration Testing

Objectives

After completing this chapter, you should be able to:

- Test e-mail security
- Use antiphishing tools
- Use antispam tools

Key Terms

Blind relayer a server that resends spam e-mails to hide the origin of the messages

Spambot a program that harvests e-mail addresses from Web pages

Introduction to E-Mail Security Penetration Testing

E-mail accounts store private, confidential information intended only for select parties. This makes them a prime target for attackers, who use a wide variety of techniques to gain unauthorized access to these accounts. E-mail encryption software and secure connections between users and e-mail servers are necessary to protect against these attacks. This chapter teaches you how to test the security of e-mail servers.

Obtaining an E-Mail ID

Before beginning e-mail penetration testing, it is necessary to obtain a valid e-mail ID for the system to be tested. There are several ways to do this, including the following:

- When testing a free e-mail service, simply sign up for a free account.
- Software programs, called spiders, can be used to search Web pages for e-mail addresses. These can be configured to look for addresses from a specific domain.
- E-mail extraction software can search for e-mail IDs based on keywords.

Steps for E-Mail Penetration Testing

Step 1: E-Mail ID and Password

In the first step, testers attempt a few basic techniques to obtain e-mail IDs and passwords. Social engineering is one effective method, through which the tester tries to directly fool users into revealing private information. The tester may try this over the telephone, through e-mail, a chat room, a social networking Web site, or any other means of communication.

Even if the social engineer cannot directly extract a password, he or she might be able to gather other information that will lead to the password. Many e-mail sites ask for a security question, such as "What is the name of the city in which you were born?" If a tester knows this information, and that information is much easier to obtain in normal conversation, he or she can use it to retrieve the password.

Different password-cracking tools can also be used to obtain passwords, although using these tools can be very time consuming. The following are some common techniques used in password cracking:

- *Dictionary attack*: A dictionary file is loaded in a cracking application, and all words in that file are attempted
- *Hybrid attack*: Adds numbers or symbols to the words in the dictionary, because many people change their password by simply adding a number at the end of their present password
- *Brute-force attack*: All possible combinations of characters are attempted as the password, which takes an enormous amount of time

The following are some popular password-cracking tools:

- John the Ripper
- RainbowCrack
- Brutus
- WebCracker
- ObiWaN

Step 2: Antiphishing Software

Phishing is an Internet scam used to trick users into revealing personal and confidential information. This attack is a type of social engineering where the attacker poses as a legitimate organization to fool the user. For instance, the victim may receive an authentic-looking e-mail claiming to be from his or her bank, but once the victim enters his or her account information, that information is forwarded directly to the phisher.

The attacker sends e-mails to many potential victims at once, using forged e-mail headers to make them appear to originate from elsewhere. Phishing e-mails may contain the following characteristics:

- Look professional and legitimate
- Based on authorized corporate e-mails with minimal changes
- Use of HTML to hide the target URLs of links
- Attached viruses and worms
- Personalized or unique

To test antiphishing software, the testing team can try to send a phishing e-mail to an e-mail ID in the organization and check whether the software detects it. If the message passes through without the software generating an alert, the software must be updated, reconfigured, or replaced.

Step 3: Antispam Tools

Spam or junk e-mails are unsolicited messages sent in bulk. Spammers use rogue ISPs and multiple domain names to avoid spam-blocking software, and they typically switch ISPs when caught. Spam messages are sent through ***blind relayers***, servers that resend spam e-mails to hide the origin of the messages.

Antispam software can sometimes be fooled using these techniques:

- If the subject line in the e-mail starts with *FW:* or *Re:*, the filter may believe the message is in reply to a trusted e-mail.

- Spam filters cannot catch text in image form, so an e-mail just containing an image may pass.

The testing team can send an obvious spam e-mail to an e-mail ID in the organization to test antispam tools.

Step 4: E-Mail Bombing

E-mail bombing is a technique in which an attacker fills a victim's mailbox with a huge number of large messages. This may cause a denial of service, especially if multiple accounts are hit by this technique. Penetration testers should attempt this on an e-mail ID in the organization and check if the messages are marked or blocked.

Step 5: CLSID Extension Vulnerability

Attachments with a class ID (CLSID) file extension do not show the actual extension of the file. Files with this type of extension may look like harmless JPG or WAV files, when they are in fact malicious executables. Many e-mail content filters do not catch this.

To perform a CLSID extension vulnerability test, testers send an attachment with a CLSID file extension to an e-mail ID in the organization. They then open the attachment in the target mailbox. If it runs, the target is vulnerable to this attack.

Step 6: VBS Attachment Vulnerability

VBS files consist of commands that, when executed, can do anything to a computer, including running malicious code such as viruses and worms. Users must run the files manually, but many users are not aware that these files are dangerous. Most e-mail security programs will warn against opening executable attachments, but many do not catch VBS files.

To perform a VBS extension vulnerability test, the team sends a VBS file as an attachment to an e-mail ID in the organization. They then open the attachment in the target mailbox. If it runs, the target is vulnerable to this attack.

Step 7: Double File-Extension Vulnerability

Some attackers may attempt to hide the true nature of a file by giving it multiple extensions. For instance, a file with the following filename may look like a harmless JPEG image at first glance, but it is actually an executable file:

```
EXAMPLE..JPG.exe
```

Windows users are especially vulnerable to this attack if they choose to hide known file extensions in Windows Explorer. This will hide the *.exe* portion entirely.

To perform a double file-extension vulnerability test, testers can send a file with a double extension as an attachment to an e-mail ID in the organization. They can then open the attachment in the target mailbox. If it runs, the target is vulnerable to this attack.

Step 8: Long-Filename Vulnerability

Some e-mail security programs can only test filenames up to a certain length. If an attacker sends a malicious file with an extremely long filename, it may go undetected. To perform a long-filename vulnerability test, testers can send a file with a long filename as an attachment to an e-mail ID in the organization. They can then open the attachment in the target mailbox. If it runs, the target is vulnerable to this attack.

Step 9: ActiveX Vulnerability

Because of vulnerabilities in Internet Explorer and Outlook, ActiveX controls contained within HTML content can sometimes bypass security measures and run code in the Microsoft Virtual Machine (MVM), even when that code should not be allowed.

To perform an ActiveX vulnerability test, testers send an HTML-based e-mail message to an e-mail ID in the organization. These tests are designed to detect whether the e-mail system is protected. Some of the tests may execute automatically. Others require the end user to run an attachment. If the file attachment gfi-test.txt appears on the desktop, the system is vulnerable to the attack being tested. In this case, gfi-test.txt is created automatically and contains vital system information.

Step 10: Iframe Remote Vulnerability

This attack does not require any attachments. In this type of attack, an HTML e-mail contains a tag pointing to a file on a remote HTTP site. This will download the file to the recipient's machine, without the victim taking any action. To perform an Iframe vulnerability test, the testing team sends an e-mail containing an Iframe pointing to a file residing on an HTTP server. They then open the mail message and see if the file downloads.

Step 11: MIME Header Vulnerability

MIME (Multipurpose Internet Mail Extensions) exploits use a malformed MIME header and an Iframe tag to run a VBS file in Outlook. This VBS file will execute automatically when the e-mail message is opened. To test for the MIME header vulnerability, testers send an HTML e-mail message with an executable attachment and modified MIME header information. Hackers often include a wrong, or incorrectly specified, to or from address. After sending the test e-mail, testers then access the e-mail and try to read it. If the executable attachment runs, the mail program is vulnerable to this attack.

Step 12: Malformed File-Extension Vulnerability

Files with HTA (HTML Application) extensions contain commands that can run malicious code, including viruses and worms. These files can bypass many security programs, including many versions of Outlook. To test for the malformed file-extension vulnerability, testers send a file with the .HTA extension to an e-mail ID in the organization. They then open the message and try to open the file. If the file executes, the system is vulnerable.

Step 13: Access Exploit Vulnerability

Files containing VBA (Visual Basic for Applications) code are automatically executed without any security warning, regardless of the target machine's security settings. This is dangerous for any computer with Internet Explorer installed. To perform an Access exploit vulnerability test, the testing team sends a Microsoft Access file with VBA code to an e-mail ID in the organization. They then open the message and try to read the attachment. If the attachment opens, the system is vulnerable.

Step 14: Fragmented-Message Vulnerability

Splitting a large file into multiple smaller messages is called message fragmentation. A client that supports this feature can transparently reassemble these small messages into the single large file; however, many security measures only scan the smaller files and miss any threats in the complete file.

To perform a fragmented-message vulnerability test, testers send fragmented messages to an e-mail ID in the organization. They then open the message and try to read the attachment. If the message is contained in a single e-mail with the attachment, the system is vulnerable.

Step 15: Long Subject-Name Attachment

Some security programs are fooled by e-mail messages with long subject names, along with attachments of the same name. To perform a long subject-name attachment test, testers send a message with a long subject name and attach an executable file with the same name as the e-mail subject, and give that file a .DAT extension. They then open the message. If the attachment is executed, the system is vulnerable.

Antiphishing Tools

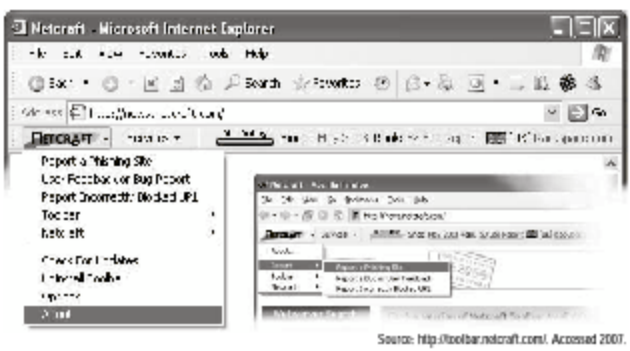
PhishTank SiteChecker

PhishTank SiteChecker is an extension for Firefox, SeaMonkey, Internet Explorer, Opera, Mozilla, and Flock. It compares Web sites to an online database of known phishing sites. If the site is in the database, SiteChecker blocks it.

PhishTank SiteChecker is shown in Figure 6-1, and its features include the following:

- Displays a page detailing the reason the site was blocked, allowing users to ignore the warning if they choose
- Comes in many languages
- Includes several options

Figure 6-1 PhishTank SiteChecker blocks known phishing sites.



Source: http://toolbar.netcraft.com/. Accessed 2007.

Figure 6-2 The Netcraft Toolbar allows users to report phishing sites.

Netcraft Toolbar

Users of the Netcraft Toolbar can report phishing sites, which are then stored in an online database. When any subsequent users attempt to access those sites, the toolbar warns those users and blocks the site. The Netcraft Toolbar is shown in Figure 6-2, and its features include the following:

- Catches suspicious URLs containing characters commonly used to deceive
- Forces pop-up windows to display navigational controls
- Clearly displays sites' hosting locations, including country, to help catch fraudulent sites

GFI MailEssentials

GFI MailEssentials' antiphishing module detects and blocks phishing e-mails. In addition to comparing messages to a constantly updated blacklist, it looks for typical phishing keywords in every e-mail received. GFI MailEssentials is shown in Figure 6-3.

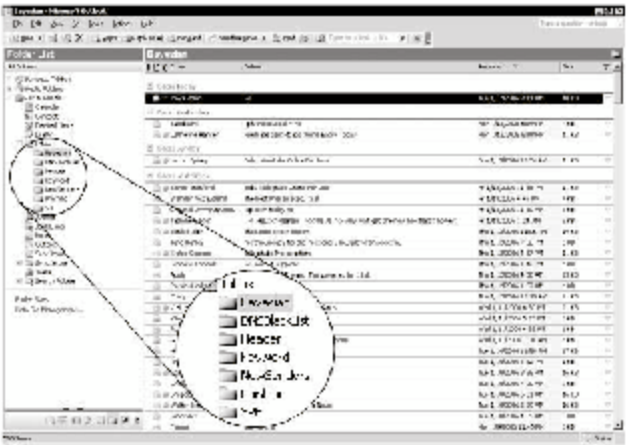
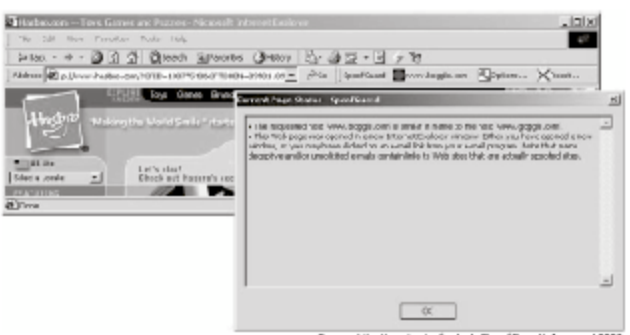


Figure 6-3 GFI MailEssentials compares messages to a blacklist and looks for typical phishing keywords.



Source: http://crypto.stanford.edu/SpoofGuard/. Accessed 2007.

Figure 6-4 SpoofGuard displays a traffic light indicating dangerous Web pages.

SpoofGuard

SpoofGuard places a traffic light in the user's browser toolbar that indicates increasingly dangerous Web pages by turning from green to yellow to red. If the user tries to enter sensitive information into a form from a phishing site, SpoofGuard saves the data and warns the user. It also displays warnings according to the user's preferences. SpoofGuard is shown in Figure 6-4.

Antispam Tools

AEVITA Stop SPAM Email

The AEVITA Stop SPAM Email tool replaces all e-mail addresses on a Web page with specially encoded e-mail addresses. These e-mail addresses appear normal, and mailto: links still work. However, programs that harvest e-mail addresses for spammers, called ***spambots***, are unable to decipher them. The tool can even generate large lists of fake e-mail addresses. This will overload spambots and give spammers an excess of useless data, in order to make things more difficult for them. AEVITA Stop SPAM Email is shown in Figure 6-5.

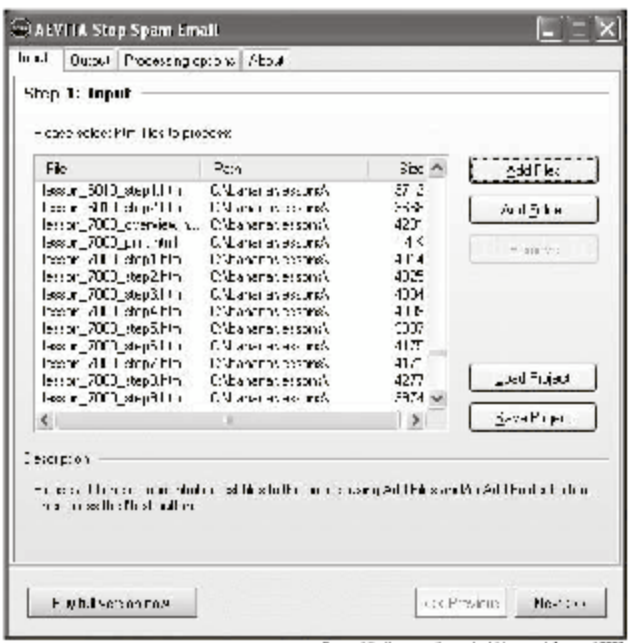
SpamExperts Desktop

SpamExperts Desktop filters POP3/IMAP e-mails for both spam and viruses. It intercepts all e-mail before it reaches the client, and it filters spam in the background by learning from the user what is and is not spam. After installing it, the user manually classifies at least 10 messages as safe and 10 messages as spam, and the tool will begin learning.

SpamEater Pro

SpamEater Pro connects to a user's e-mail account and removes spam messages from the server, before the local e-mail client downloads them. If the user chooses, SpamEater Pro can locally archive these junk messages. It runs a number of checks on the messages, including whitelists and blacklists for sender, country of origin, and server. The program will even send a complaint to the sender's ISP using SpamCop.

SpamEater Pro removes 95 percent of spam messages and is shown in Figure 6-6.



Source: http://www.aevita.com/webtoolspam/. Accessed 2007.

Figure 6-5 AEVITA Stop SPAM Email encodes e-mail addresses in Web pages to thwart spambots.

Spytech SpamAgent

SpamAgent has more than 1,500 configurable filters to block unwanted e-mail messages and attachments. The program is shown in Figure 6-7, and its features include the following:

- Filters incoming messages by sender, recipient, subject, text in its body, and header
- Includes special filters such as attachment filtering, filtering if no subject is present, filtering if sender equals recipient, and filtering if the incoming message is a forward

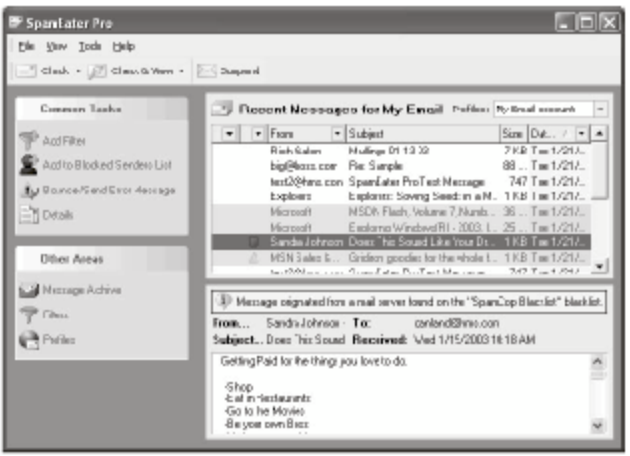


Figure 6-6 SpamEater Pro deletes spam messages directly from the server.

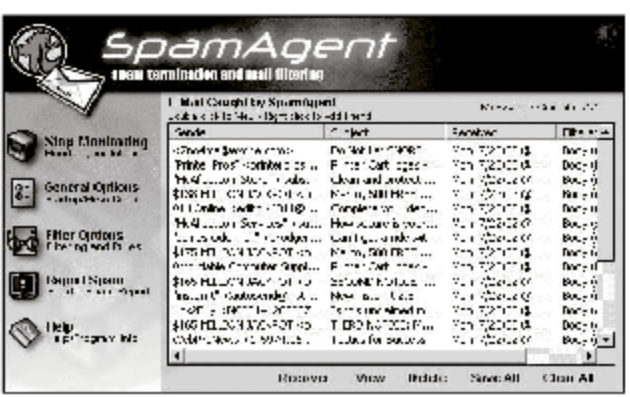


Figure 6-7 Spytech SpamAgent has over 1,500 configurable e-mail filters.

- Allows a whitelist to accept all mail from certain users
- Sends a customizable unsubscribe notice to the sender of the spam or an abuse report to the spammer's ISP
- Blocks either all incoming attachments or just executables that may potentially damage the system

Chapter Summary

- E-mail accounts store private, confidential information intended only for select parties.
- Phishing is an Internet scam in which attackers try to fool users into revealing personal and confidential information.
- Social engineering techniques can be used to get hints to passwords.
- Spamming is the process of sending unsolicited or junk e-mails in bulk.
- Mail bombing is the act of sending excessive e-mails to fill the recipient's mailbox.
- VBS files and HTA (HTML Application) files contain commands that, once executed, can perform malicious actions on a computer.
- E-mail messages with long subject names can bypass a network's security settings.
- Several tools are available to guard against phishing and spam.

Chapter 7

Security Patches Penetration Testing

Objectives

After completing this chapter, you should be able to:

- Understand the concept of patch management
- Describe a patch and vulnerability group (PVG)
- Understand the steps involved in security patches penetration testing
- Enumerate some of the more popular patch management tools

Key Terms

Patch and vulnerability group (PVG) a team in an organization that uses OS patching, application patching, and configuration changes to eradicate vulnerabilities

Patch management a part of system management that involves acquiring patches from vendors, testing patches, and installing patches to an administered computer system

Introduction to Security Patches Penetration Testing

Security patches are an important part of maintaining system security. Managing these patches and making sure that all relevant patches are installed is one of the most crucial roles of a system administrator. This chapter describes the concept of patch management and enumerates tools that administrators can use to automate the process. It also discusses PVGs, teams in organizations who are responsible for patch management. This chapter also teaches you the steps involved in performing penetration testing of security patches.

Patch Management

Patch management is a part of system management that involves:

- Acquiring patches from vendors
- Testing patches
- Installing patches to an administered computer system

Patch management tasks include the following:

- Maintaining information about patches that are available
- Deciding which patches are appropriate for particular systems
- Ensuring that patches are installed properly
- Testing the systems after installation
- Preparing documentation of all the associated procedures

Patch and Vulnerability Group (PVG)

A *PVG* uses OS patching, application patching, and configuration changes to eradicate vulnerabilities. Each and every organization must have a PVG team that looks after the patches and vulnerabilities program throughout the organization.

The following are some of the responsibilities of a PVG:

- Conduct testing of patches and nonpatch remediation
- Create a database of remediation
- Distribute vulnerability and improvisation information to local administrators
- Configure automatic updates of applications
- Monitor security sources for vulnerability announcements like patch and nonpatch improvisation

Penetration Testing Steps

The following are the steps involved in security patches penetration testing:

- *Step 1*: Check if the organization has a PVG in place.
- *Step 2*: Check whether the security environment is updated.
- *Step 3*: Check whether the organization uses automated patch management tools.
- *Step 4*: Check the last date of patching.
- *Step 5*: Check the patches on nonproduction systems.
- *Step 6*: Check the vendor authentication mechanism.
- *Step 7*: Check whether downloaded patches contain viruses.
- *Step 8*: Check for dependencies on new patches.

Step 1: Check If the Organization Has a PVG in Place

The penetration tester should ask the management team about teams that deal with the following:

- Acquiring patches from vendors
- Testing patches
- Installating patches to an administered computer system

If any authorized person or team deals with the above, it means the company has a PVG.

Step 2: Check Whether the Security Environment Is Updated

With the installation of the latest patches, new types of vulnerabilities may also arise. The new patch may affect the security environment. To avoid this, it is essential to update the security environment along with the installation of a new patch.

To check if the security environment is updated, testers can try any malicious action on the system and check whether the security environment (such as the firewall, antivirus software, and security software tools) detects such actions.

Step 3: Check Whether the Organization Uses Automated Patch Management Tools

Patch management tools help the PVG install patches automatically on many computers simultaneously. Manual installation is a time-consuming process and in many cases, before the patching can be completed for all systems, either new patches are released or a new vulnerability is discovered. Therefore, the penetration tester should check if the organization uses automated patch management tools.

Step 4: Check the Last Date of Patching

To check the last date of patching, a tester can follow these steps:

1. Check whether the PVG maintains a database of patches.
2. Check the last date when patches were installed.

Step 5: Check the Patches on Nonproduction Systems

It is important to test patches on nonproduction systems, because if there are any vulnerabilities present in a patch, an organization does not want those vulnerabilities to affect production systems and interrupt normal business. There are a wide number of system configurations, and the vender cannot test all cases, so it is always better to test patches before installation.

Step 6: Check the Vendor Authentication Mechanism

The vendor provides an authentication mechanism when a PVG downloads a patch. Authentication helps to ensure that the patch is from the right source and has not been tampered with.

The authentication method can be:

- Cryptographic checksums
- Pretty Good Privacy (PGP) signatures
- Digital certificates

Step 7: Check Whether Downloaded Patches Contain Viruses

There are chances that patches contain viruses. Installing such patches can affect the system. Therefore, before installing any patches, the PVG should scan them with an antivirus tool.

The following are the steps for testing patches for viruses:

1. Check whether the virus signature database and antivirus program are up to date.
2. Download the patch to a nonproduction system.
3. Run a virus scan on all the patches after they've been downloaded.
4. After installing the patches, run a virus scan again.

Step 8: Check for Dependencies on New Patches

In many cases, there is a dependency between patches such that they should be installed in a sequence or one patch is removed when the other is installed. In some cases, installation of the latest patch without the previous version will not work, while in some other cases, installation of new patches uninstalls or disables another patch.

Therefore, it is necessary to do the following:

- Make sure patches are installed in sequence if there is a dependency.
- Check whether the installation of new patches unintentionally uninstalls or disables another patch.

Patch Management Tools

The following are some patch management tools:

- BigFix Patch Management
- BindView Patch Management

- CS Enterprise Vulnerability Management Suite
- Ecora Patch Manager
- eTrust Vulnerability Manager
- GFI LANguard Network Security Scanner
- Hercules
- HFNetChkPro
- HP OpenView Patch Manager using Radia
- Kaseya Patch Management
- LANDesk Patch Manager
- LiveState Patch Manager
- ManageSoft Security Patch Management
- Marimba Patch Management
- NetIQ Vulnerability Manager
- Opsware Server Automation System
- PatchLink Update
- PolicyMaker Software Update
- Prism Patch Manager
- SecureCentral PatchQuest
- Security Update Manager
- Service Pack Manager
- Sitekeeper (Patchkeeper module)
- Software Update Services
- Systems Management Server
- SysUpdate
- UpdateEXPERT
- Windows Server Update Services
- ZENworks Patch Management

Chapter Summary

- Patch management is a part of system management that involves the acquisition, testing, and installation of patches to an administered computer system.
- With the installation of the latest patches, new types of vulnerabilities may also arise.
- A patch and vulnerability group (PVG) must be created by each organization.
- A PVG uses OS patching, application patching, and configuration changes to eradicate vulnerabilities.
- Organizations should use automated patch management tools.
- Before installing patches, administrators should check the vendor authentication mechanism.
- It is important to test patches on nonproduction systems.

Index