Penetration test

A **penetration test**, colloquially known as a **pen test** or **ethical hacking**, is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system;[1][2] this is not to be confused with a vulnerability assessment.[3] The test is performed to identify weaknesses (also referred to as vulnerabilities), including the potential for unauthorized parties to gain access to the system's features and data,[4][5] as well as strengths,[6] enabling a full risk assessment to be completed.

The process typically identifies the target systems and a particular goal, then reviews available information and undertakes various means to attain that goal. A penetration test target may be a white box (about which background and system information are provided in advance to the tester) or a black box (about which only basic information—if any—other than the company name is provided). A gray box penetration test is a combination of the two (where limited knowledge of the target is shared with the auditor).[7] A penetration test can help identify a system's vulnerabilities to attack and estimate how vulnerable it is.[8][6]

Security issues that the penetration test uncovers should be reported to the system owner.[9] Penetration test reports may also assess potential impacts to the organization and suggest countermeasures to reduce the risk.[9]

The UK National Cyber Security Center describes penetration testing as: "A method for gaining assurance in the security of an IT system by attempting to breach some or all of that system's security, using the same tools and techniques as an adversary might."[10]

The goals of a penetration test vary depending on the type of approved activity for any given engagement, with the primary goal focused on finding vulnerabilities that could be exploited by a nefarious actor, and informing the client of those vulnerabilities along with recommended mitigation strategies.[11]

Penetration tests are a component of a full security audit. For example, the Payment Card Industry Data Security Standard requires penetration testing on a regular schedule, and after system changes.[12] Penetration testing also can support risk assessments as outlined in the NIST Risk Management Framework SP 800-53 (https://csrc.nist.gov/projects/risk-management/sp800-53-controls/release-search#!/control?version=4.0&number=CA-8) .[13]

Several standard frameworks and methodologies exist for conducting penetration tests. These include the Open Source Security Testing Methodology Manual (OSSTMM), the Penetration Testing Execution Standard (PTES), the NIST Special Publication 800-115, the Information System Security Assessment Framework (ISSAF) and the OWASP Testing Guide.

Flaw hypothesis methodology is a systems analysis and penetration prediction technique where a list of hypothesized flaws in a software system are compiled through analysis of the specifications and documentation for the system. The list of hypothesized flaws is then prioritized on the basis of the estimated probability that a flaw actually exists, and on the ease of exploiting it to the extent of control or compromise. The prioritized list is used to direct the actual testing of the system.

There are different types of penetration testing, depending upon the goal of the organization which include: Network (external and internal), Wireless, Web Application, Social Engineering, and Remediation Verification.

History

By the mid 1960s, growing popularity of time-sharing computer systems that made resources accessible over communication lines created new security concerns. As the scholars Deborah Russell and G. T. Gangemi Sr. explain, "The 1960s marked the true beginning of the age of computer security."[14]:27

In June 1965, for example, several of the U.S.'s leading computer security experts held one of the first major conferences on system security—hosted by the government contractor, the System Development Corporation (SDC). During the conference, someone noted that one SDC employee had been able to easily undermine various system safeguards added to SDC's AN/FSQ-32 time-

sharing computer system. In hopes that further system security study would be useful, attendees requested "...studies to be conducted in such areas as breaking security protection in the time-shared system." In other words, the conference participants initiated one of the first formal requests to use computer penetration as a tool for studying system security.[15]:7–8

At the Spring 1967 Joint Computer Conference, many leading computer specialists again met to discuss system security concerns. During this conference, the computer security experts Willis Ware, Harold Petersen, and Rein Turn, all of the RAND Corporation, and Bernard Peters of the National Security Agency (NSA), all used the phrase "penetration" to describe an attack against a computer system. In a paper, Ware referred to the military's remotely accessible time-sharing systems, warning that "Deliberate attempts to penetrate such computer systems must be anticipated." His colleagues Petersen and Turn shared the same concerns, observing that online communication systems "...are vulnerable to threats to privacy," including "deliberate penetration." Bernard Peters of the NSA made the same point, insisting that computer input and output "...could provide large amounts of information to a penetrating program." During the conference, computer penetration would become formally identified as a major threat to online computer systems.[15]:8

The threat that computer penetration posed was next outlined in a major report organized by the United States Department of Defense (DoD) in late 1967. Essentially, DoD officials turned to Willis Ware to lead a task force of experts from NSA, CIA, DoD, academia, and industry to formally assess the security of time-sharing computer systems. By relying on many papers presented during the Spring 1967 Joint Computer Conference, the task force largely confirmed the threat to system security that computer penetration posed. Ware's report was initially classified, but many of the country's leading computer experts quickly identified the study as the definitive document on computer security.[15] Jeffrey R. Yost of the Charles Babbage Institute has more recently described the Ware report as "...by far the most important and thorough study on technical and operational issues regarding secure computing systems of its time period."[16] In effect, the Ware report reaffirmed the major threat posed by computer penetration to the new online time-sharing computer systems.

To better understand system weaknesses, the federal government and its contractors soon began organizing teams of penetrators, known as *tiger teams*, to use computer penetration to test system security. Deborah Russell and G. T. Gangemi Sr. stated that during the 1970s "...'tiger teams' first emerged on the computer scene. Tiger teams were government and industry-sponsored teams of crackers who attempted to break down the defenses of computer systems in an effort to uncover, and eventually patch, security holes."[14]:29

A leading scholar on the history of computer security, Donald MacKenzie, similarly points out that, "RAND had done some penetration studies (experiments in circumventing computer security controls) of early time-sharing systems on behalf of the government."[17][18] Jeffrey R. Yost of the Charles Babbage Institute, in his own work on the history of computer security, also acknowledges that both the RAND Corporation and the SDC had "engaged in some of the first so-called 'penetration studies' to try to infiltrate time-sharing systems in order to test their vulnerability."[16] In virtually all these early studies, tiger teams successfully broke into all targeted computer systems, as the country's time-sharing systems had poor defenses.

Of early tiger team actions, efforts at the RAND Corporation demonstrated the usefulness of penetration as a tool for assessing system security. At the time, one RAND analyst noted that the tests had "...demonstrated the practicality of system-penetration as a tool for evaluating the effectiveness and adequacy of implemented data security safeguards." In addition, a number of the RAND analysts insisted that the penetration test exercises all offered several benefits that justified its continued use. As they noted in one paper, "A penetrator seems to develop a diabolical frame of mind in his search for operating system weaknesses and incompleteness, which is difficult to emulate." For these reasons and others, many analysts at RAND recommended the continued study of penetration techniques for their usefulness in assessing system security.[15]:9

Presumably the leading computer penetration expert during these formative years was James P. Anderson, who had worked with the NSA, RAND, and other government agencies to study system security. In the early 1971, the U.S. Air Force contracted Anderson's private company to study the security of its time-sharing system at the Pentagon. In his study, Anderson outlined a number of major factors involved in computer penetration. Anderson described a general attack sequence in steps:

1. Find an exploitable vulnerability.
2. Design an attack around it.
3. Test the attack.
4. Seize a line in use.
5. Enter the attack.
6. Exploit the entry for information recovery.

Over time, Anderson's description of general computer penetration steps helped guide many other security experts, who relied on this technique to assess time-sharing computer system

security.[15]:9

In the following years, computer penetration as a tool for security assessment became more refined and sophisticated. In the early 1980s, the journalist William Broad briefly summarized the ongoing efforts of tiger teams to assess system security. As Broad reported, the DoD-sponsored report by Willis Ware had "...showed how spies could actively penetrate computers, steal or copy electronic files and subvert the devices that normally guard top-secret information. The study touched off more than a decade of quiet activity by elite groups of computer scientists working for the Government who tried to break into sensitive computers. They succeeded in every attempt."[19]

While these various studies may have suggested that computer security in the U.S. remained a major problem, the scholar Edward Hunt has more recently made a broader point about the extensive study of computer penetration as a security tool. Hunt suggests in a recent paper on the history of penetration testing that the defense establishment ultimately "...created many of the tools used in modern day cyberwarfare," as it carefully defined and researched the many ways that computer penetrators could hack into targeted systems.[15]:5

Tools

A wide variety of security assessment tools are available to assist with penetration testing, including free-of-charge, free software, and commercial software.

Specialized OS distributions

Several operating system distributions are geared towards penetration testing.[20] Such distributions typically contain a pre-packaged and pre-configured set of tools. The penetration tester does not have to hunt down each individual tool, which might increase the risk of complications—such as compile errors, dependency issues, and configuration errors. Also, acquiring additional tools may not be practical in the tester's context.

Notable penetration testing OS examples include:

- BlackArch based on Arch Linux
- BackBox based on Ubuntu
- Kali Linux (replaced BackTrack December 2012) based on Debian

- Parrot Security OS based on Debian
- Pentoo based on Gentoo
- WHAX based on Slackware

Many other specialized operating systems facilitate penetration testing—each more or less dedicated to a specific field of penetration testing.

A number of Linux distributions include known OS and application vulnerabilities, and can be deployed as *targets* to practice against. Such systems help new security professionals try the latest security tools in a lab environment. Examples include Damn Vulnerable Linux (DVL), the OWASP Web Testing Environment (WTW), and Metasploitable.

Software frameworks

- BackBox
- Hping
- Metasploit Project
- Nessus
- Nmap
- OWASP ZAP
- SAINT
- w3af

Penetration testing phases

The process of penetration testing may be simplified into the following five phases:

1. Reconnaissance: The act of gathering important information on a target system. This information can be used to better attack the target. For example, open source search engines can be used to find data that can be used in a social engineering attack.
2. Scanning: Uses technical tools to further the attacker's knowledge of the system. For example, Nmap can be used to scan for open ports.
3. Gaining access: Using the data gathered in the reconnaissance and scanning phases, the attacker can use a payload to exploit the targeted system. For example, Metasploit can be

used to automate attacks on known vulnerabilities.

4. Maintaining access: Maintaining access requires taking the steps involved in being able to be persistently within the target environment in order to gather as much data as possible.
5. Covering tracks: The attacker must clear any trace of compromising the victim system, any type of data gathered, log events, in order to remain anonymous.[21]

Once an attacker has exploited one vulnerability they may gain access to other machines so the process repeats i.e. they look for new vulnerabilities and attempt to exploit them. This process is referred to as pivoting.

Vulnerabilities

Legal operations that let the tester execute an illegal operation include unescaped SQL commands, unchanged hashed passwords in source-visible projects, human relationships, and old hashing or cryptographic functions. A single flaw may not be enough to enable a critically serious exploit. Leveraging multiple known flaws and shaping the payload in a way that appears as a valid operation is almost always required. Metasploit provides a ruby library for common tasks, and maintains a database of known exploits.

When working under budget and time constraints, fuzzing is a common technique that discovers vulnerabilities. It aims to get an unhandled error through random input. The tester uses random input to access the less often used code paths. Well-trodden code paths are usually free of errors. Errors are useful because they either expose more information, such as HTTP server crashes with full info trace-backs—or are directly usable, such as buffer overflows.

Imagine a website has 100 text input boxes. A few are vulnerable to SQL injections on certain strings. Submitting random strings to those boxes for a while will hopefully hit the bugged code path. The error shows itself as a broken HTML page half rendered because of an SQL error. In this case, only text boxes are treated as input streams. However, software systems have many possible input streams, such as cookie and session data, the uploaded file stream, RPC channels, or memory. Errors can happen in any of these input streams. The test goal is to first get an unhandled error and then understand the flaw based on the failed test case. Testers write an automated tool to test their understanding of the flaw until it is correct. After that, it may become obvious how to package the payload so that the target system triggers its execution. If this is not viable, one can hope that another error produced by the fuzzer yields more fruit. The use of a fuzzer saves time by not checking adequate code paths where exploits are unlikely.

Payload

The illegal operation, or payload in Metasploit terminology, can include functions for logging keystrokes, taking screenshots, installing adware, stealing credentials, creating backdoors using shellcode, or altering data. Some companies maintain large databases of known exploits and provide products that automatically test target systems for vulnerabilities:

- Metasploit
- Nessus
- Nmap
- OpenVAS
- W3af

Standardized government penetration test services

The General Services Administration (GSA) has standardized the "penetration test" service as a pre-vetted support service, to rapidly address potential vulnerabilities, and stop adversaries before they impact US federal, state and local governments. These services are commonly referred to as Highly Adaptive Cybersecurity Services (HACS) and are listed at the US GSA Advantage website.[22]

This effort has identified key service providers which have been technically reviewed and vetted to provide these advanced penetration services. This GSA service is intended to improve the rapid ordering and deployment of these services, reduce US government contract duplication, and to protect and support the US infrastructure in a more timely and efficient manner.

132-45A Penetration Testing[23] is security testing in which service assessors mimic real-world attacks to identify methods for circumventing the security features of an application, system, or network. HACS Penetration Testing Services typically strategically test the effectiveness of the organization's preventive and detective security measures employed to protect assets and data. As part of this service, certified ethical hackers typically conduct a simulated attack on a system, systems, applications or another target in the environment, searching for security weaknesses. After testing, they will typically document the vulnerabilities and outline which defenses are effective and which can be defeated or exploited.

In the UK penetration testing services are standardized via professional bodies working in collaboration with National Cyber Security Centre.

The outcomes of penetration tests vary depending on the standards and methodologies used. There are five penetration testing standards: Open Source Security Testing Methodology Manual[24] (OSSTMM), Open Web Application Security Project (OWASP), National Institute of Standards and Technology (NIST00), Information System Security Assessment Framework (ISSAF), and Penetration Testing Methodologies and Standards (PTES).

See also

- IT risk
- ITHC
- Tiger team
- White hat (computer security)

General references

- Long, Johnny (2011). *Google Hacking for Penetration Testers*, Elsevier[25]
- The Definitive Guide to Penetration Testing[26]

References

1. *"What Is Penetration Testing?" (https://www.doi.gov/ocio/customers/penetration-testing) . Retrieved 2018-12-18.*
2. *"Penetration Testing overview" (https://cybersguards.com/web-application-penetration-testing-checklist-updated-2019/) . Retrieved 2019-01-25.*
3. *"What's the difference between a vulnerability assessment and a penetration test?" (https://www.pgitl.com/blog/whats-the-difference-between-a-vulnerability-assessment-and-a-penetration-test/) . Retrieved 2020-05-21.*
4. *The CISSP® and CAPCM Prep Guide: Platinum Edition. John Wiley & Sons. 2006-11-06. ISBN 978-0-470-00792-1. "A penetration test can determine how a system reacts to an attack, whether or not a system's defenses can be breached, and what information can be acquired from the system"*

5. Kevin M. Henry (2012). Penetration Testing: Protecting Networks and Systems. IT Governance Ltd. ISBN 978-1-849-28371-7. "Penetration testing is the simulation of an attack on a system, network, piece of equipment or other facility, with the objective of proving how vulnerable that system or "target" would be to a real attack."

6. Cris Thomas (Space Rogue), Dan Patterson (2017). Password Cracking is easy with IBM's Space Rogue (https://www.techrepublic.com/videos/video-password-cracking-is-easy-with-ibms-space-rogue/) (Video). CBS Interactive. Event occurs at 4:30-5:30. Retrieved 1 December 2017.

7. "Pen Testing Types explained" (https://www.ncsc.gov.uk/guidance/penetration-testing) . 2017-06-09. Retrieved 2018-10-23.

8. "Penetration Testing: Assessing Your Overall Security Before Attackers Do" (https://www.sans.org/reading-room/analysts-program/PenetrationTesting-June06) . SANS Institute. Retrieved 16 January 2014.

9. "Writing a Penetration Testing Report" (http://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343) . SANS Institute. Retrieved 12 January 2015.

10. "Penetration Testing" (https://www.ncsc.gov.uk/guidance/penetration-testing) . NCSC. Aug 2017. Retrieved 30 October 2018.

11. Patrick Engebretson, The basics of hacking and penetration testing (http://store.elsevier.com/The-Basics-of-Hacking-and-Penetration-Testing/Patrick-Engebretson/isbn-9780124116443/) Archived (https://web.archive.org/web/20170104104648/http://store.elsevier.com/The-Basics-of-Hacking-and-Penetration-Testing/Patrick-Engebretson/isbn-9780124116443/) 2017-01-04 at the Wayback Machine, Elsevier, 2013

12. Alan Calder and Geraint Williams (2014). PCI DSS: A Pocket Guide, 3rd Edition. ISBN 978-1-84928-554-4. "network vulnerability scans at least quarterly and after any significant change in the network"

13. "NIST Risk Management Framework" (https://csrc.nist.gov/projects/risk-management/sp800-53-controls/release-search#!/control?version=4.0&number=CA-8) . NIST. 2020.

14. Russell, Deborah; Gangemi, G.T. (1991). Computer Security Basics (https://archive.org/details/computersecurity00russ) . O'Reilly Media Inc. ISBN 9780937175712.

15. Hunt, Edward (2012). "US Government Computer Penetration Programs and the Implications for Cyberwar". IEEE Annals of the History of Computing. **34** (3): 4–21. doi:10.1109/MAHC.2011.82 (https://doi.org/10.1109%2FMAHC.2011.82) . S2CID 16367311 (https://api.semanticscholar.org/CorpusID:16367311) .

16. Yost, Jeffrey R. (2007). de Leeuw, Karl; Bergstra, Jan (eds.). A History of Computer Security Standards, in The History of Information Security: A Comprehensive Handbook. Elsevier. pp. 601–602.

17. Mackenzie, Donald; Pottinger, Garrel (1997). "Mathematics, Technology, and Trust: Formal Verification, Computer Security, and the U.S. Military" (https://www.computer.org/csdl/mags/an/1997/03/man1997030041-abs.html) . IEEE Annals of the History of Computing. **19** (3): 41–59. doi:10.1109/85.601735 (https://doi.org/10.1109%2F85.601735) .

18. *Mackenzie, Donald A. (2004). Mechanizing Proof: Computing, Risk, and Trust (https://books.google.com/books?id=QiMS8t4V_0cC) . Massachusetts Institute of Technology. p. 156. ISBN 978-0-262-13393-7.*
19. *Broad, William J. (September 25, 1983). "Computer Security Worries Military Experts", The New York Times*
20. *Faircloth, Jeremy (2011). "Chapter 1:Tools of the Trade" (http://zempirians.com/ebooks/Jeremy%20Faircloth-Penetration%20Tester's%20Open%20Source%20Toolkit,%20Third%20Edition%20%20-Elsevier%20Science%20(2011).pdf) (PDF). Penetration Tester's Open Source Toolkit (Third ed.). Elsevier. ISBN 978-1597496278. Retrieved 4 January 2018.*
21. *"Summarizing The Five Phases of Penetration Testing - Cybrary" (https://www.cybrary.it/2015/05/summarizing-the-five-phases-of-penetration-testing/) . Cybrary. 2015-05-06. Retrieved 2018-06-25.*
22. *"GSA HACS SIN 132-45 Services" (https://web.archive.org/web/20190323122222/https://www.gsaadvantage.gov/advantage/s/search.do?q=0:2132-45&db=1&searchType=2) . 1 March 2018. Archived from the original (https://www.gsaadvantage.gov/advantage/s/search.do?q=0:2132-45&db=1&searchType=2) on 23 March 2019. Retrieved 1 March 2018.*
23. *"Pen Testing Services" (https://web.archive.org/web/20180626140009/https://www.gsaelibrary.gsa.gov/ElibMain/sinDetails.do?executeQuery=YES&scheduleNumber=70&flag=&filter=&specialItemNumber=132+45A) . 1 March 2018. Archived from the original (https://www.gsaelibrary.gsa.gov/ElibMain/sinDetails.do?executeQuery=YES&scheduleNumber=70&flag=&filter=&specialItemNumber=132+45A) on 26 June 2018. Retrieved 1 March 2018.*
24. *"Open-Source Security Testing Methodology Manual - an overview | ScienceDirect Topics" (https://www.sciencedirect.com/topics/computer-science/open-source-security-testing-methodology-manual) . www.sciencedirect.com. Retrieved 2021-10-13.*
25. *Long, Johnny (2011). Google Hacking for Penetration Testers. Elsevier Science. ISBN 978-0-08-048426-6.*
26. *"Definitive Guide to Penetration Testing | Core Sentinel" (https://www.coresentinel.com/definitive-guide-penetration-testing/) . Core Sentinel. Retrieved 2018-10-23.*

Last edited 2 months ago **by** Discospinster