5-2021

A Study of Penetration Testing Processes and Tools

Sushmitha Reddy Mamilla

Recommended Citation

Mamilla, Sushmitha Reddy, "A Study of Penetration Testing Processes and Tools" (2021). *Electronic Theses, Projects, and Dissertations*. 1220.
https://scholarworks.lib.csusb.edu/etd/1220

A STUDY OF PENETRATION TESTING PROCESSES AND TOOLS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Information Systems and Technology:

Cybersecurity

by

Sushmitha Reddy Mamilla

May 2021

A STUDY OF PENETRATION TESTING PROCESSES AND TOOLS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Sushmitha Reddy Mamilla

May 2021

Approved by:

Dr. Benjamin Becerra, Committee Chair

Dr. Shayo, Committee Member

Dr. Javad Varzandeh, Committee Member, Chair, IDS Department

ABSTRACT

With cybercrimes on the rise, cybersecurity has become very important today. Every company is eager to avoid cybercrimes like data breaches and hacking. To prepare for such incidents, every company has many protection systems in place. However, the best method to test the strength of these protective measures is penetration testing. Every pen tester must consider many factors like budget, time, and scope of the organization’s penetration testing to choose just the right tool for each phase of the process. This project analyzes the most efficient scanning tool by testing them in a Kali Linux environment.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my family for standing by me during my master's degree, and especially during the time I was invested in my project. It would be least to say that I owe them immense gratitude for their constant support and encouragement. Also, I would like to express my deepest appreciation to my supervisor, Dr. Benjamin Becerra for his guidance and support during my project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Background and Motivation

The Internet has opened many doors that would have been unthinkable at one time. At the same time, the internet has made us wary about what these open doors could bring in. Every organization has strong policies for information security through confidentiality, integrity, and availability. However, every data breach reminds us about how these policies alone might not be all infallible for security of the organization. According to IBM's Cost of Data Breach Report 2020, the average cost of a data breach is $ 3.86 million with the United States being the recipient of the most expensive data breaches at $ 8.64 million ("Cost", 2020). Given that prevention is better than cure, penetration testing is used as an arsenal to protect an organization by discovering secret doors before intruders do by using similar techniques as those intruders and closing these doors by a remediation plan.

Johnson (2021) published a report about the increase in financial damage due to cybercrime in the United States from $ 17.8 million in 2001 to $ 4200 million in 2020. This kind of astronomical increase within a span of 20 years makes the prevention of cybercrime a major goal for everyone in IT.

Given such a scenario, every organization deploys security measures to protect themselves. However, every organization needs to be confident about the effectiveness of these measures against attackers. Penetration Testing is what allows us to test the effectiveness of these security measures in the face of an attack. Penetration testing is a form of stress testing to identify flaws and establish security strength in the Trusted Computing Base (Weissman, 1995). Many tools are available for each stage of the Penetration Testing process. Each of these tools can be further categorized by their compatibility with the host Operating System (Windows, Linux, Unix, MAC OS X). But the real question is about their effectiveness in terms of the number of vulnerabilities discovered in a certain time frame. Every tool's vendor makes long claims about how efficient their tool is compared to other tools. But every pen tester and students of pen testing must rely on their own instinct or prior experience to select the tool. This might not be the most efficient way, especially during a real penetration test.

Objective of the Project:

The objective of this project is to compare some scanning tools in terms of the number of ports discovered and the time taken by the tool to discover those ports. A comparative analysis of the results generated by these tools will be used to identify the most efficient tools. The scanning tools to be tested will be further discussed in the results section of this project. So, this project answers the research question, which scanning tool is the most efficient?

Apart from the above analysis, this project also studies the various types, the process, and the models of penetration testing. Seven different types of Penetration testing will be discussed, along with two models of Penetration Testing: Flaw Hypothesis and Attack Tree. The difference between Vulnerability Analysis and Penetration testing will be discussed in detail.

CHAPTER TWO

LITERATURE REVIEW

Penetration Testing

Penetration Testing is also referred to as pen test. National Institute of Standards and Technology (NIST) defines Penetration Testing as the security testing which imitates cyber-attacks to identify the vulnerabilities of a system or a network before they can be taken advantage of by adversaries in the real world. Weissman (1995) called Penetration Testing “a pseudo-enemy attack by a friendly evaluation team on a computer system of interest to discover ways to breach the system's security controls, to penetrate the security perimeter of protection to obtain sensitive information, to obtain unauthorized services, or to cause damage to the system that denies service to legitimate users”. The UK National Cyber Security Center defines 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" (“Penetration Testing”, 2017). As pointed out by Bishop (2007), the “aspect” being tested need not be a computer system or network. It can also be a building, or a combination of people, an office, and a computer system. Osborne (2006) has defined pen testing in his book as “A test to ensure that gateways, firewalls and systems are appropriately designed and configured to protect against unauthorized access or attempts to disrupt services” (p.257).

Pen Testing is a part of a full information security audit and many companies routinely perform this test.

The History of Penetration Testing

In the 1960s, when multiple users started sharing the same resource, the risk of this resource sharing resulted in the IT industry realizing the need for computer security. It was in 1965 that at a conference for computer system security, the use of penetration testing was formally suggested. It was the US Department of Defense (DoD) that sponsored the “tiger teams'' in the 1970s. “Tiger teams were government and industry-sponsored teams of crackers who attempted to break down the defense of computer systems to uncover, and eventually patch, security holes'' (Russell & Gangemi, 1991, p.29). Although these tiger teams were able to uncover some vulnerabilities, it was apparent very soon that this method had many flaws, including, not being able to prevent a second penetration attack and unreliability due to new vulnerabilities being found by new teams. It became obvious then that a more stringent approach than tiger teams were needed.

It was James P. Anderson who introduced “reference monitors” in the Computer Security Technology Planning Study. A reference monitor “enforces the authorized access relationships between subjects and objects of a system.” (Russell & Gangemi, 1991, p.30). These reference monitors resulted in the development of standards and technologies for secure systems. It was pointed out by Hunt (2012) that after researching and analyzing the security of resource sharing system at the Pentagon, Anderson described a pen test attack in steps:

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

This was the first technique that has been used to assess resource-sharing computer system security. In 1993, a paper called “Improving the Security of Your Site by Breaking into it” was written by Dan Farmer of Sun Microsystems and Wietse Venema of Eindhoven University of Technology. This paper is about the “uebercracker”, the hacker who uses his own hacking programs, as opposed to using the existing scripts. This makes an uebercracker harder to detect and hence posing a very serious threat to security. Famer & Venema further pointed out that a system’s owner must similarly learn to test his own system thinking of himself as a hacker. This was the basis for Penetration testing. In 2003, the OWASP or Open Web Application Security Project introduced the Testing Guide which had the first framework for Penetration testing. In 2014, the OWASP version 4 was released with improvements over the previous versions.

Penetration Testing Vs Vulnerability Assessment

According to Shinde & Ardhapurkar (2016), vulnerabilities are weaknesses or flaws in the system that could potentially lead to security breach. Adversaries find these vulnerabilities and exploit them as a means of compromising the system. Injection, Broken Authentication, Sensitive Data Exposure, XML External Entities (XXE) and Broken Access Control are some examples of vulnerabilities.

Vulnerability Assessment is the method of testing a system or a network to identify potential vulnerabilities. Penetration Testing encompasses vulnerability assessment with evidence of success with the testing process. Pen testing uses manual testing along with automated processes to simulate the attack on the system, whereas Vulnerability Assessment relies on automated testing processes to discover vulnerabilities (CPNI, 2006).

According to Shinde & Ardhapurkar (2016), the advantages and disadvantages of Vulnerability assessment are:

A. Advantages of Vulnerability Assessment:

   i. Automating thousands of security checks is possible using Vulnerability Assessment
   ii. Can be used to bring together an organization’s threat and vulnerability management program.

B. Disadvantages of Vulnerability Assessment:

i. Automated tools can give a lot of false positives and an unnecessary amount of data.

ii. Vulnerability assessments cannot identify logical attack vectors like reusing of passwords.

iii. Remedies proving by Vulnerability Assessment are usually nonspecific and depend on the output of the tool.

Shinde & Ardhapurkar (2016) further list the advantages and disadvantages of Penetration Testing:

A. Advantages of Penetration Testing:

i. Penetration Testing takes into consideration security mitigating controls.

ii. It looks at vulnerabilities discovered to describe a detailed picture about the issues.

iii. No false positives.

B. Disadvantages of Penetration Testing:

i. Can only be performed by certified pen testers, so, in the absence of an in-house pen tester, organizations avail an outside company for the purpose of pen testing.

ii. Penetration Testing takes more time, effort, and money than vulnerability assessment.

The Goals of Penetration Testing

The goal of a Penetration Test is to certify the effectiveness of the security measures taken by an organization to protect their system. Penetration testing achieves this by discovering vulnerabilities by simulating an attack by adversaries.

The Types of Penetration Testing

According to Osborne (2006), there are different types of pen testing: Network Penetration Tests, Application Penetration Tests, Periodic Network Vulnerability Assessments and Physical Security Tests. Firch (2021) further classifies penetration testing into Client-side Penetration Tests, Wireless Penetration Tests and Social Engineering.

1. Network Penetration Test:

   Network Penetration Test is also referred to as Network Service Penetration Testing or Infrastructure Testing (Firch, 2021). Servers, routers, firewalls, switches, etc. constitute the infrastructure of a network. Testing this infrastructure for vulnerabilities is Network Penetration Testing. DNS level attacks, Man in the Middle (MIM) Attacks, Router and SSH attacks can be combated by scheduling this testing in an organization.

   As pointed out by Agarwal (2019), Network Penetration Tests should be conducted on the client side and the external point, since both ends constitute the access points of a network. According to CPNI (2006), there are three types of network penetration tests: Black-box, white-box, and gray-box penetration testing. In Black- box testing, pen testers are given no prior information about the network that they are going to run the pen test on. This method helps a pen tester understand what an adversary with no prior knowledge of the network might achieve. However, they are given full information in white-box testing. This method helps a pen tester determine most of the vulnerabilities and attack vectors. In contrast, in gray-box testing, they are given some of the information about the system about to be tested. This method allows a pen tester to understand the degree of access that is accessible to an authorized user (Phong, 2014).

2. Application Penetration Tests:

   Vulnerabilities in web- based applications are identified by these Application Penetration Tests (Firch,2021). These web- based applications include web applications, browsers and their components like Plugins, Applets, ActiveX, Scriptlets (Agarwal,2019).

   These tests are complex and very detail oriented. Since the number of threats originating from web applications is big and serious, every web -application's endpoint that communicates with the user must be recognized and tested on a very regular basis (Agarwal, 2019). This makes this test very time consuming and given the importance of this test, it also requires careful planning.

Software companies take this testing very seriously and employ pen testers to perform this test to discover vulnerabilities in their code. Google offers a reward through “Google Vulnerability Reward Program (VRP)” to people who are able to find qualifying vulnerabilities in their web-based applications (“Program rules”, n.d.).

3. Periodic Network Vulnerability Assessments:

   According to Osborne (2006), these assessments are used to “augment a complete penetration test”. Usually encompasses regularly scanning IP ranges and noting the changes.

4. Physical Security Tests:

   These tests simulate an attack on a physical barrier like infrastructure of a company to identify weaknesses that could be exploited. These tests help improve the physical security of an organization.

5. Client-side Penetration Tests:

   As pointed out by Agarwal (2019), these tests are used to discover threats that emerge locally, on the client side. Firch (2021) lists these cyber-attacks as:

   i. Cross-Site Scripting Attacks
   ii. Clickjacking Attacks
   iii. Cross-Origin Resource Sharing
   iv. Form Hijacking

v. HTML Injection

vi. Open Redirection

vii. Malware infection.

6. Wireless Penetration Tests:

All the wireless devices connected to the Wi-Fi on the client’s side are tested by Wireless Penetration tests. These wireless devices can be laptops, phones, iPads, iPods, etc. This test is usually performed on-site. To conduct this test remotely, NUC and Wi-Fi Pineapple are used (Firch, 2021).

Firch (2021) also points out that a pen tester must consider the following to perform this Wireless Penetration test:

i. Identify all access points.

ii. Is the incoming and outgoing data encrypted?

iii. Deploy monitoring systems to identify unauthorized users.

iv. Check if the wireless network has been duplicated or misconfigured.

v. How is the wireless network being protected?

vi. IS WPA protocol being used at all access points?

7. Social Engineering Penetration testing:

This occurs when a pen tester tricks a victim into divulging their sensitive information like passwords. Firch (2021) listed these attacks as:

i. Phishing Attacks

ii. Vishing

iii. Smishing

iv. Tailgating

v. Imposters

vi. Name Dropping

vii. Eavesdropping

viii. Dumpster Diving.

These tests are important to verify the “human network” of an organization (Agarwal,2019).

The Models of Penetration Testing

Many new models of Penetration Testing are being used today. However, the two original models of Penetration Testing are Flaw Hypothesis and Attack Tree (Phong, 2014).

McDermott (2000) defines a flaw as “a demonstrated undocumented capability, which can be exploited to violate some aspect of the security policy”. Phong (2014) states that a Flaw Hypothesis Methodology (FHM) has six stages:

i. Define goals of Penetration Testing: The scope of Penetration testing is defined with clear objectives set for the test. Also, test ground rules of the Penetration test are established.

ii. Conduct Background study: According to McDermott (2000), a background study for penetration testing includes “system design documentation, source code, user documentation, and results of unit and integration testing.”

iii. Flaw Generation: Generate hypothetical (suspected) flaws using brainstorming sessions, or the Delphi technique.

iv. Flaw confirmation: After the flaws generated in the previous stage are analyzed, filtered, and sorted by priority, source code analysis is used to confirm or deny them as categorizing them as true, false or untested.

v. Flaw Generalization: Generalize discovered flaws by analyzing them for patterns of similar mistakes made in the system.

vi. Flaw Elimination: The discovered flaws are recommended for repair or their risks are managed by using external controls.

The other model of Penetration Testing is called “Attack Tree”. Salter et al. (1998) defined attack trees as “a visualization tool to enumerate and weigh different attacks against a system”. The tree shows different potential attacks against the system. The objective of the attacker is represented as the root node and the child nodes or leaf nodes or leaves are the different ways in which that goal is achieved. So, essentially, each leaf node is a specific attack that could happen to achieve the goal of the attacker.

This approach was developed at Sparta and is considered a top-down approach in Penetration Testing (McDermott,2000). The Attack tree model of Penetration testing is preferred when we do not have enough prior knowledge about the system that has to be tested. According to Salter et al. (1998), this methodology has five steps:

1. Attack trees are developed for the system.
2. Weights are applied to each of the leaves. These weights are risk, access, and cost of implementation.
3. Attack tree is pruned in a way that only exploitable leaves remain. The exploitable leaves are the ones that are the closest to an attacker's goals and have a chance of giving him a sufficient return.
4. Corresponding countermeasures are generated for the exploitable leaves.
5. Countermeasure options are optimized by ranking them using attributes like cost of implementation and operation, availability, handiness, profit after system resources and compatibility with existing technology.

Penetration Testing Processes

There are many different processes for penetration testing. Depending on the needs of the entity that requires the pen test, a specific process is chosen. According to Thorsen, Nufryk, & Taylor, (2019), there are eight phases in a traditional Penetration Testing Process:

Phase 1: Planning:

This is the first step in the process of Penetration Testing. Scope of the Pen test is defined in this step. Tiller (2011) stated that the scope and scale of the test is decided based on factors like existing security policies, culture, laws and regulations, best practices and industry requirements.

This is a very important step because it defines the entire test and guides the deliverable of the test.

Phase 2: Reconnaissance:

This step is the information gathering stage where a pen tester gathers all the information he can about the organization or the system that is to be pen tested, in the hopes that this information can be useful during the attack. This information gathering can be passive information gathering and deliberate information gathering. Passive information gathering is collecting publicly available information. Deliberate information gathering is to detect vulnerabilities by scanning ports (Thorsen, Nufryk, & Taylor, 2019).

Phase 3: Scanning:

Also known as vulnerability scanning, this stage is when a pen tester uses scanning tools to scan for vulnerabilities in a target system. (Thorsen, Nufryk & Taylor, 2019).

Phase 4: Gaining Access:

Using the knowledge gained from reconnaissance and exploiting the vulnerabilities discovered in scanning, a pen tester starts attacking the target system to gain access into that system. (Thorsen, Nufryk & Taylor, 2019).

Phase 5: Maintaining Access:

Once the pen testers gain access to the system in the previous stage, they use various mechanisms to continue their access in the system (Thorsen, Nufryk & Taylor, 2019).

Phase 6: Covering tracks:

Pen testers cover their own tracks by deleting the evidence that they were ever inside the system (Thorsen, Nufryk & Taylor, 2019).

Phase 7: Analysis:

In this stage, pen testers analyze all the information acquired during the testing process, along with the vulnerabilities discovered and also suggest remediation measures to counteract the identified vulnerabilities (Thorsen, Nufryk & Taylor, 2019).

Phase 8: Reporting:

This is the stage where all the information collected in the previous stages is formally reported to the company stakeholders. This report usually consists of vulnerabilities discovered, sensitive data accessed, time taken for the pen test and suggested remediation solutions.

CHAPTER THREE

METHODOLOGY

To research various papers that helped me write this project, I used resources like Google Scholar, learning.oreilly.com, ieee xplore.org, researchgate.net and ACM Digital library. I used the services of Google Scholar, researchgate.net and ieeexplore.org because they are known as reputable sources. I further used keywords: Penetration testing, vulnerability, penetration, and security testing while searching for resources in these websites.

I used an online book, CISO's guide to penetration testing from learning.oreilly.com to help me understand the processes in Penetration Testing. I also used ACM Digital library for some journals and articles about penetration testing using the key words penetration and security testing. I took care that most of my research papers were dated after the year 2000. I did find a particularly useful handbook about computer security by Clark Weissman. However, this handbook was published in 1973 and I had to omit this article because most of the material, while being helpful, was outdated considering the year.

Additionally, I performed Google searches for articles about penetration testing and about companies that offer penetration testing. I included the articles of reputable companies and industry veterans in choosing these sources.

My culminating experience project has a lab section about the analysis of penetration testing tools. I used the cyberlab.csusb.edu resource provided by my university to access Kali Linux in the VMWare web console. I selected a few scanning tools for the purpose of this research and tested them in that Kali Linux environment. All the tools I tested were included with Kali Linux and I did not download them from any other sources. The IP address scanned for this lab is the IP address of my virtual machine.

CHAPTER FOUR

RESULTS

Penetration Testing Tools

1. Network Scanning:

According to Wack, Tracy & Souppaya (2003), network scanning involves the use of a port scanner to identify all the active hosts, open ports, switches and routers in the address range. Operating System fingerprinting occurs when the open ports discovered by scanning tools identify the target Operating System. However, OS fingerprinting may not always give the correct answer, because system administrators can use mechanics like firewall filters to disguise their real operating systems.

Although port scanners are completely automated, they do not identify vulnerabilities by themselves. Only the pen tester looking at the results of this port scanning can identify vulnerabilities by interpreting and analyzing those results.

Table 1. List of Network Scanning Tools

Scanning Tools	Description of the Tool	Cost of the Tool
Nmap	Port scanning tool used to discover active hosts and scan for open ports (Wack, Tracy & Souppaya, 2003).	free
OpenVas	Open Vulnerability Assessment System is an open-source software framework for vulnerability management and scanning (Thorsen, Nufryk & Taylor, 2019).	free
Dmitry	Command line port scanner that scans both TCP and UDP ports ("Kali Linux", n.d.).	free
Unicornscan	Port scanner that scans TCP scanning tools ("Kali Linux", n.d.).	free

Sparta	GUI port mapper that scans networks to identify available hosts on the network (“Kali Linux”, n.d.).	free
Netcat	Popularly known as the swiss army utility of a security engineer, it is a port scanner that is also used in reading and writing data across the network (Wilson, 2021).	free
SolarWinds Port Scanner	Scanning tool that generates a list of open closed and filtered ports for an IP address (“Free port”, n.d.).	Free 30 day trial
Angry IP Scanner	Scanning tool that scans ports and IP addresses and is compatible with Linux, Windows, and MAC OS X (“Angry IP”, n.d.).	free

ManageEngine OpUtils	Port scanning tool that also provides network address monitoring and tools for administration (Wilson, 2021).	Free trial

2. Password Cracking:

According to Wack, Tracy & Souppaya (2003), password cracking is used to identify weak passwords. This tool uses the method of creating and storing password hashes for every password that is input by the user. This hash is an encrypted form of the entered password. So, the next time this user enters that password, another hash is created and matched with the stored hash. The user is given authentication only if these two hashes are equal.

During a Penetration Test, three types of password cracking attacks can be used: dictionary attack, hybrid attack and brute force method. Dictionary attacks are fastest, and attack is by using all the words listed in a dictionary. However, this attack is weak. Hybrid attacks use numbers and symbols along with words from a dictionary. Brute force attacks are the strongest attacks and use a trial-and-error method of generating passwords and hashes.

Password Cracking tools are also called Credential Testing Tools. Wack, Tracy & Souppaya (2003) further asserted that a strong Linux or Unix password has more than 10 characters and contains upper cases, lower cases, special characters, and numbers. Usually, organizations use password cracking tools monthly to ensure that their passwords cannot be cracked easily. However, when they discover after using these tools that a lot of their passwords can be cracked, they modify their policies to decrease the number of passwords that can be cracked.

Table 2. List of Password Cracking Tools

Password Cracking Tool	Description of the Tool	Cost of the Tool
John the Ripper	Password recovery tool available for Linux, Unix (11 Versions), DOS, Win32, and OpenVMS (Thorsen, Nufryk & Taylor, 2019).	free
IMP 2.0	NetWare password cracking tool that facilitates a user to get passwords	free

	through various attack methods (Wack, Tracy & Souppaya, 2003).	
L0pht Crack	Password cracking tool compatible with Windows NT, Windows 2000, and Windows XP (Wack, Tracy & Souppaya, 2003).	Pro Version-$295 Admin- $595 Consultant-$1195 (Rubens, 2009)
Crack 5	Unix password cracker used to identify weak passwords in Unix (Wack, Tracy & Souppaya, 2003).	free
Cain and Abel	Password recovery tool compatible with Windows (Thorsen, Nufryk & Taylor, 2019).	free

3. Vulnerability assessment:

Also known as Vulnerability scanning tools, they scan for vulnerabilities. They differ from network scanning tools in that, unlike network scanners, they do not require a human to interpret the results of scanning to discover vulnerabilities (Wack, Tracy & Souppaya, 2003).

Table 3. List of Vulnerability Assessment Tools

Vulnerability Assessment Tool	Description of the Tool	Cost of the Tool
Nessus	Vulnerability scanner that scans for vulnerabilities, misconfigurations, default passwords and susceptibility to DoS or Denial of Service attacks (Thorsen, Nufryk & Taylor, 2019).	free
SARA	Vulnerability scanner that identifies gaps in security on networks (Wack, Tracy & Souppaya, 2003).	free

SATAN	Vulnerability scanning tool that helps system administrators by discovering and reporting problems with network security (Wack, Tracy & Souppaya, 2003).	free

4. Other Miscellaneous Tools:

Table 4. List of Miscellaneous Tools

Tool	Description of the Tool	Cost of the Tool
Wireshark	Open-source network protocol analyzer used to sniff and monitor traffic on a network (Thorsen, Nufryk & Taylor, 2019).	free
Metasploit Framework	Pen testing framework that is command line based and is used to find and exploit	free

	vulnerabilities (Thorsen, Nufryk & Taylor, 2019).	
Recon-ng	Web reconnaissance tool compatible with Kali Linux and used to automate OSINT. Can be used to file search, identify hosts, geolocation, search password hashes and look for VPN (Thorsen, Nufryk & Taylor, 2019).	free
Peach	It provides dynamic application security testing or DAST for pen testing which is an automated testing tool that helps avoid zero-day attacks. (Thorsen, Nufryk & Taylor, 2019).	Free trial version

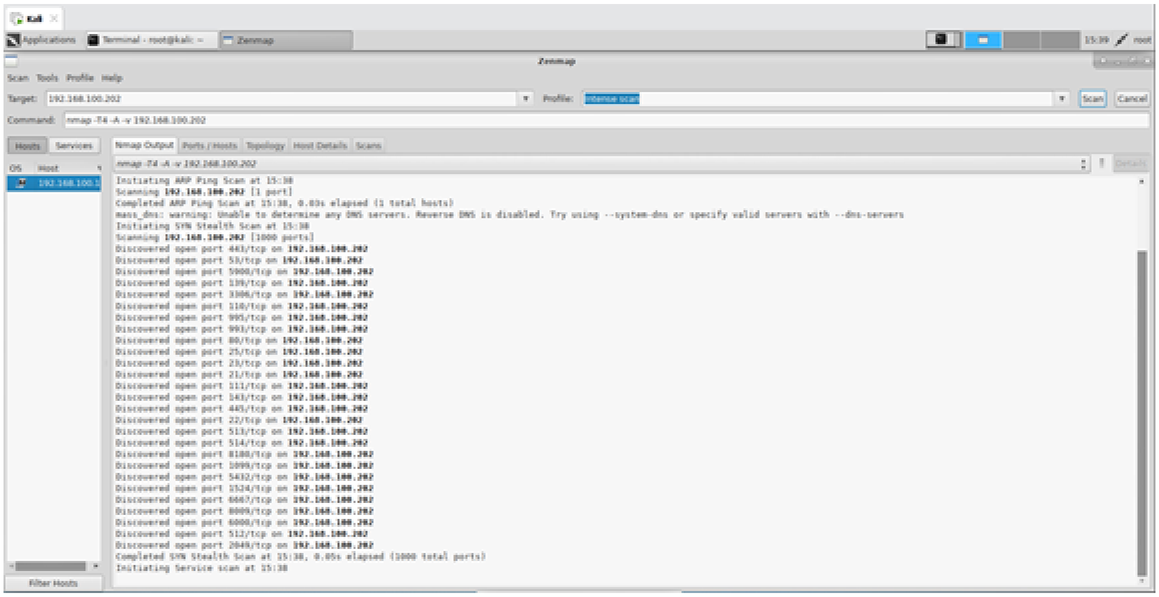
Analysis of Scanning Tools

This following section shows an analysis of some of the scanning tools chosen from the list of scanning tools mentioned in the previous section. This exercise was conducted using the cyberlab.csusb.edu provided by the university. This cyberlab was used to launch a VMware Remote Console in which the tools were tested.

- Tools: Nmap, Dmitry, Unicornscan
- OS System: Kali Linux (Virtual Machine)
- IP address: 192.168.100.202

1.Nmap

Kali Linux has a Graphic User Interface (GUI) for Nmap. After entering the Target IP address (192.168.100.202) and then clicking on the “scan”, Nmap gives the complete details about the open ports on the host as shown below.



Target: 192.168.100.202

Profile: Intense scan

Command: nmap -T4 -A -v 192.168.100.202

```
nmap -T4 -A -v 192.168.100.202
```

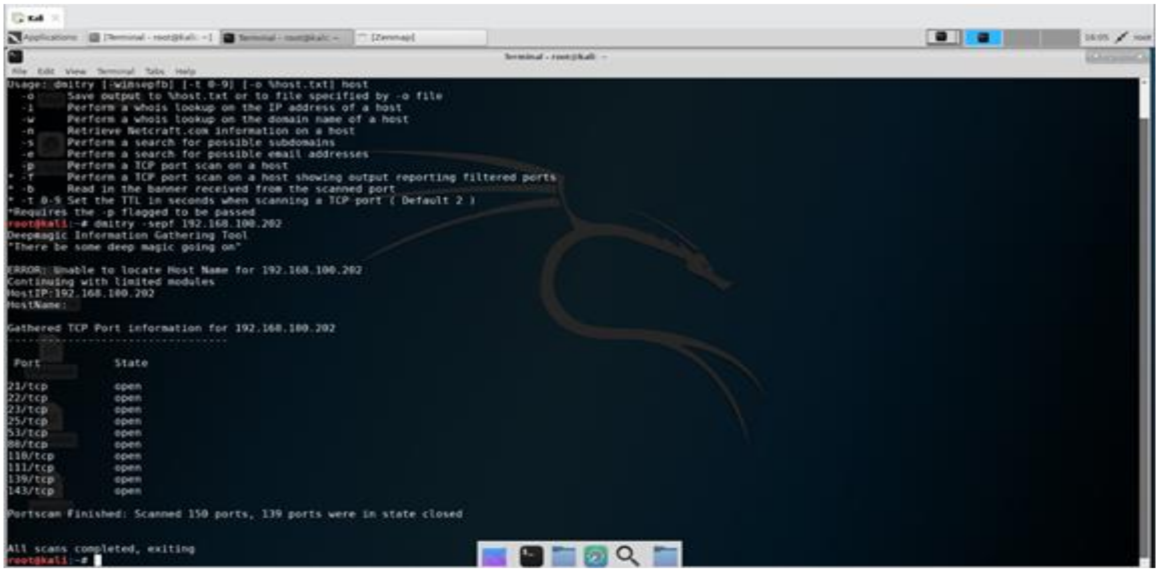
```
Initiating ARP Ping Scan at 15:38
Scanning 192.168.100.202 [1 port]
Completed ARP Ping Scan at 15:38, 0.03s elapsed (1 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Initiating SYN Stealth Scan at 15:38
Scanning 192.168.100.202 [1000 ports]
Discovered open port 443/tcp on 192.168.100.202
Discovered open port 53/tcp on 192.168.100.202
Discovered open port 5900/tcp on 192.168.100.202
Discovered open port 139/tcp on 192.168.100.202
Discovered open port 3306/tcp on 192.168.100.202
Discovered open port 110/tcp on 192.168.100.202
Discovered open port 995/tcp on 192.168.100.202
Discovered open port 993/tcp on 192.168.100.202
Discovered open port 80/tcp on 192.168.100.202
Discovered open port 25/tcp on 192.168.100.202
Discovered open port 23/tcp on 192.168.100.202
Discovered open port 21/tcp on 192.168.100.202
Discovered open port 111/tcp on 192.168.100.202
Discovered open port 143/tcp on 192.168.100.202
Discovered open port 445/tcp on 192.168.100.202
Discovered open port 22/tcp on 192.168.100.202
Discovered open port 513/tcp on 192.168.100.202
Discovered open port 514/tcp on 192.168.100.202
Discovered open port 8180/tcp on 192.168.100.202
Discovered open port 1099/tcp on 192.168.100.202
Discovered open port 5432/tcp on 192.168.100.202
Discovered open port 1524/tcp on 192.168.100.202
Discovered open port 6667/tcp on 192.168.100.202
Discovered open port 8009/tcp on 192.168.100.202
Discovered open port 6000/tcp on 192.168.100.202
Discovered open port 512/tcp on 192.168.100.202
Discovered open port 2049/tcp on 192.168.100.202
Completed SYN Stealth Scan at 15:38, 0.05s elapsed (1000 total ports)
Initiating Service scan at 15:38
```

Figure 1. Nmap Port Scanning Results

2. Dmitry

Dmitry is included with Kali Linux. The command used to scan the ports of the given IP address is: dmitry -sepf 192.168.100.202

The above command immediately gave the open ports as shown below:



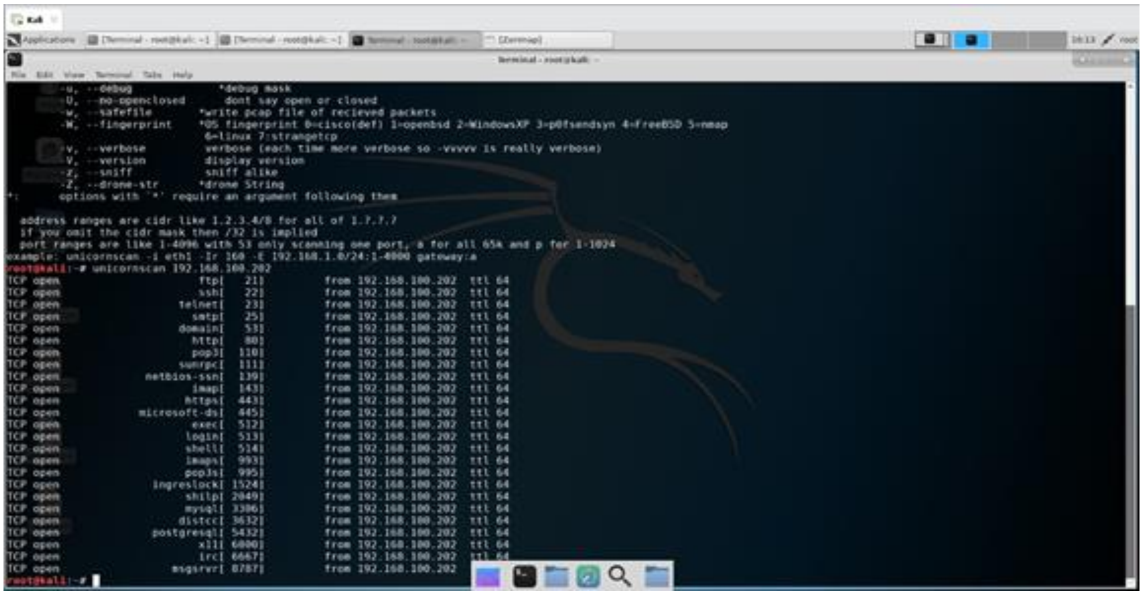
```
Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o      Save output to %host.txt or to file specified by -o file
  -i      Perform a whois lookup on the IP address of a host
  -w      Perform a whois lookup on the domain name of a host
  -n      Retrieve Netcraft.com information on a host
  -s      Perform a search for possible subdomains
  -e      Perform a search for possible email addresses
  -p      Perform a TCP port scan on a host
* -f      Perform a TCP port scan on a host showing output reporting filtered ports
* -b      Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
root@kali:~# dmitry -sepf 192.168.100.202
Deepmagic Information Gathering Tool
"There be some deep magic going on"

ERROR: Unable to locate Host Name for 192.168.100.202
Continuing with limited modules
HostIP:192.168.100.202
HostName:

Gathered TCP Port information for 192.168.100.202
---------------------------------

 Port           State

21/tcp          open
22/tcp          open
23/tcp          open
25/tcp          open
53/tcp          open
80/tcp          open
110/tcp         open
111/tcp         open
139/tcp         open
143/tcp         open

Portscan Finished: Scanned 150 ports, 139 ports were in state closed


All scans completed, exiting
root@kali:~#
```

Figure 2. Dmitry Port Scanning Results

3. Unicornscan

Unicorn scan is available with Kali Linux. The terminal command to scan ports for a given IP address: unicornscan 192.168.100.202

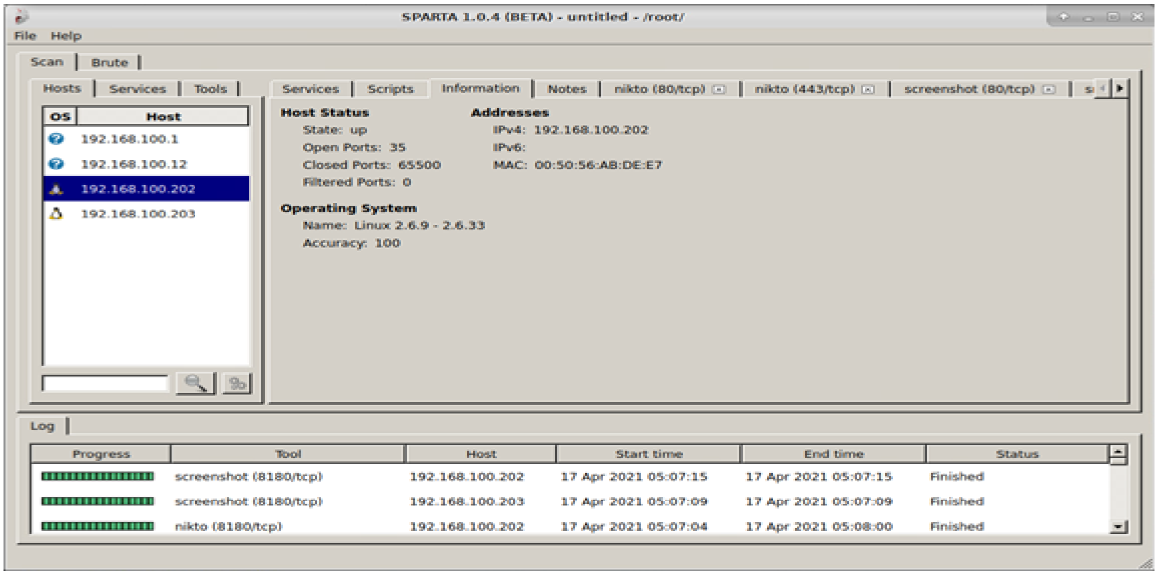
The above command gives the open ports as shown in the picture below.



```
     -u, --debug              *debug mask
     -U, --no-openclosed       dont say open or closed
     -w, --safefile          *write pcap file of recieved packets
     -W, --fingerprint       *OS fingerprint 0=cisco(def) 1=openbsd 2=WindowsXP 3=p0fsendsyn 4=FreeBSD 5=nmap
                              6=linux 7:strangetcp
     -v, --verbose            verbose (each time more verbose so -vvvvv is really verbose)
     -V, --version            display version
     -z, --sniff              sniff alike
     -Z, --drone-str         *drone String
*:       options with '*' require an argument following them

  address ranges are cidr like 1.2.3.4/8 for all of 1.?.?.?
  if you omit the cidr mask then /32 is implied
  port ranges are like 1-4096 with 53 only scanning one port, a for all 65k and p for 1-1024
example: unicornscan -i eth1 -Ir 160 -E 192.168.1.0/24:1-4000 gateway:a
root@kali:~# unicornscan 192.168.100.202
TCP open                     ftp[   21]         from 192.168.100.202  ttl 64
TCP open                     ssh[   22]         from 192.168.100.202  ttl 64
TCP open                  telnet[   23]         from 192.168.100.202  ttl 64
TCP open                    smtp[   25]         from 192.168.100.202  ttl 64
TCP open                  domain[   53]         from 192.168.100.202  ttl 64
TCP open                    http[   80]         from 192.168.100.202  ttl 64
TCP open                    pop3[  110]         from 192.168.100.202  ttl 64
TCP open                  sunrpc[  111]         from 192.168.100.202  ttl 64
TCP open             netbios-ssn[  139]         from 192.168.100.202  ttl 64
TCP open                    imap[  143]         from 192.168.100.202  ttl 64
TCP open                   https[  443]         from 192.168.100.202  ttl 64
TCP open            microsoft-ds[  445]         from 192.168.100.202  ttl 64
TCP open                    exec[  512]         from 192.168.100.202  ttl 64
TCP open                   login[  513]         from 192.168.100.202  ttl 64
TCP open                   shell[  514]         from 192.168.100.202  ttl 64
TCP open                   imaps[  993]         from 192.168.100.202  ttl 64
TCP open                   pop3s[  995]         from 192.168.100.202  ttl 64
TCP open              ingreslock[ 1524]         from 192.168.100.202  ttl 64
TCP open                   shilp[ 2049]         from 192.168.100.202  ttl 64
TCP open                   mysql[ 3306]         from 192.168.100.202  ttl 64
TCP open                  distcc[ 3632]         from 192.168.100.202  ttl 64
TCP open              postgresql[ 5432]         from 192.168.100.202  ttl 64
TCP open                     x11[ 6000]         from 192.168.100.202  ttl 64
TCP open                     irc[ 6667]         from 192.168.100.202  ttl 64
TCP open                 msgsrvr[ 8787]         from 192.168.100.202
root@kali:~# 
```

Figure 3. Unicornscan Port Scanning Results

4. Sparta



SPARTA 1.0.4 (BETA) - untitled - /root/

OS	Host
	192.168.100.1
	192.168.100.12
	192.168.100.202
	192.168.100.203

Host Status
State: up
Open Ports: 35
Closed Ports: 65500
Filtered Ports: 0

Addresses
IPv4: 192.168.100.202
IPv6:
MAC: 00:50:56:AB:DE:E7

Operating System
Name: Linux 2.6.9 - 2.6.33
Accuracy: 100

Progress	Tool	Host	Start time	End time	Status
	screenshot (8180/tcp)	192.168.100.202	17 Apr 2021 05:07:15	17 Apr 2021 05:07:15	Finished
	screenshot (8180/tcp)	192.168.100.203	17 Apr 2021 05:07:09	17 Apr 2021 05:07:09	Finished
	nikto (8180/tcp)	192.168.100.202	17 Apr 2021 05:07:04	17 Apr 2021 05:08:00	Finished

Figure 4. Sparta Port Scanning Results

Sparta is included with Kali Linux and is available as a GUI. When the target IP address is entered and scanned, it shows open ports as shown above.

Table 5. Comparison of Open Ports Scanned in all the Tools

Tool	Number of open ports scanned
Nmap	27
Dmitry	10
unicornscan	25
sparta	35

The above analysis of port scanning tools shows that sparta was the most efficient tool by identifying the greatest number of open ports. It was also very user friendly given the GUI as opposed to typing in a command in the bash shell like Dmitry and unicornscan.

However, it is to be noted that this project is limited by the number of tools tested. In the future, more tools can be tested in a similar way across varying environments to further this research.

Benefits of Penetration Testing

According to Naik, Kurundkar, Khamitkar, & Kalyankar (2009), Penetration Testing offers many benefits:

1. Identifies vulnerabilities in the system.

2. Regular Pen testing in an organization results in a drastic reduction of security incidents while also validating the effectiveness of the current security measures employed by the organization. This also results in increased trust about an organization's security policy.

3. Pen Testing also results in an organization meeting their compliance and security requirements as might be necessary under state and federal regulations.

4. Vulnerabilities discovered by the pen test are prioritized based on factors like their likelihood of occurrence and threat level. This list helps an organization direct their resources towards remediation according to the level of priority.

5. The Reconnaissance step of Penetration Testing helps an organization understand how much of their information is publicly available.

6. Penetration Testing in an organization makes their executives become more knowledgeable about their corporate liability.

7. The risk associated with internal systems and confidential information is quantified.

Cost of Penetration Testing

The cost of Penetration Testing can be between $4000 to $100,000, depending on several factors (“Average cost”,2020):

1. Size:

The larger the organization, more is the cost of Penetration Testing.

2. Complexity:

Complexity refers to the number of systems and IP addresses. Hence, more complex the organization, more expensive is the Pen Test.

3. Scope:

Scope of a Pen Test defines all the boundaries of the Pen Test and so, must be clearly defined to keep the costs in control.

4. Methodology:

Depending on the scope of the Pen Test, some of the tools needed to be used for the test could be expensive, impacting the total cost of the Pen Test.

5. Experience:

Hiring experienced Pen testers could prove more expensive for an organization.

6. External/Internal Testing:

Usually, pen testing is conducted off-site (External testing), but sometimes, depending on the requirements of the organization, it might be necessary for the pen test to be conducted on-site (Internal testing). On-site testing could include the cost of travel, relocation etc. for pen testers and hence is more expensive than off-site testing.

7. Remediation:

If the Pen tester is also expected to provide remediation for the vulnerabilities discovered during the testing process, then this impacts the cost of penetration testing.

Limitations of Penetration Testing

Despite the many benefits of pen testing, there are some major limitations to consider before starting the test (“Major Limitations”, 2020):

1. Limitation of Time:

Penetration testing is a simulation of real-world attack by attackers. However, one constraint that cannot be replicated in the test is time. Attackers could have months or may be even years of planning and scheduling the attacks, but Pen Testers have a very limited time frame to give the report of the test to their employers.

2. Limitation of Scope:

Scope of a pen test is thoroughly defined in the first stage of penetration testing. Since the scope depends on how much and what an organization wants to test, this makes the pen test limited to scope. So, if the scope only defined certain systems or networks within an organization to be tested and vulnerabilities happen to be on the systems or networks that were not tested, then they will not be identified in the pen test making an organization vulnerable to breaches, despite the pen test.

3. Limitation of Access:

Pen testing teams that have limited access to their target systems have difficulty in testing the parts of the targeted system for which they do not have access to. This limitation can be overcome by using white box testing in addition to the ongoing penetration test. This is because of the different angles from which network is attacked in white box testing.

4. Limitation of Methods:

Although theoretically, pen testers are supposed to simulate the exact conditions of an attack, they may be limited on their methods of attack that could potentially cause the system crash. Since the system is failed to have been thoroughly tested, this could leave a lot of vulnerabilities to be exploited by attackers, who do not have such limitations of methods.

5. Limitation of Skill Sets of Pen Testers:

The experience and skill set of pen testers is directly proportional to the quality of pen tests. So, hiring inexperienced pen testers or pen testers with limited knowledge of the pen test is a major limitation to the success of the pen test.

6. Limitation of Custom Exploits:

Pen testers may need to write their own scripts called custom exploits to create a custom path of attack to the target system. However, pen testers are under time and budget constraints set by the organization that hired them. Custom exploits are time consuming and more budget than regular tests, making them inefficient.

7. Limitation to Experiment:

Pen testers are bound to comply with the tools and framework approved by the organization that hired them. This limits their ability to experiment with the test because they can only use these approved tools. However, adversaries are not bound by such limitations.

CHAPTER FIVE

DISCUSSION AND RECOMMENDATIONS

Future Scope

Given the limitation of scope of Penetration testing, a test that has been poorly scoped fails to achieve the goal of pen testing, even if it did meet a compliance or government requirement. However, some organizations with genuine problems like budget which make them compromise on their scope really suffer from getting the full benefit from the pen test. So, instead, if some or most of the tasks in pen testing were automated in the future, requiring very little to no human interaction, it could greatly benefit everyone who has a problem with this limitation. Given the minimum human element in this technology, it can also help overcome the limitation of the skills set of pen testers (“What”,2018). Using Artificial Intelligence and Machine learning as a part of this automation can further increase the efficiency of the pen test in the future (Farao, 2021).

Recommendations

Of all the scanning tools tested in this project, sparta was the most efficient and easy to use. It can also be recommended because it is available in Kali Linux (although some lite versions of Kali Linux may require a download) and is a free tool, making it ideal for small businesses (less than 10 employees).

However, larger businesses with more complex systems might require a tool that is able to scan a range of IP addresses. Nmap is more recommended for these businesses.

Conclusion

Reliability in a tool is the most important aspect in penetration testing, given how each phase is executed with the appropriate tools. Hence this project focused on the comparison of four different port scanning tools to demonstrate their effectiveness against the same target.

REFERENCES

Aar, P., & Sharma, A. (2017). Analysis of Penetration Testing Tools. *International Journals of advanced research in Computer Science and Software Engineering, 7*(9), 36-41. Retrieved March 01, 2021, from https://www.researchgate.net/profile/Palak-Aar/publication/326077274_Analysis_of_Penetration_Testing_Tools/links/5f0fcd2f45851512999e55f5/Analysis-of-Penetration-Testing-Tools.pdf

Agarwal, M. (2019). Five types of Penetration Test to zero in potential vulnerabilities. Retrieved April 05, 2021, from https://www.techbeamers.com/penetration-test-and-types/

Alisherov, F. A., & Sattarova, F. Y. (2009). Methodology for Penetration Testing. *International Journal of grid and distributed computing, 2*(2), 43-50. Retrieved April 6, 2021, from http://article.nadiapub.com/IJGDC/vol2_no2/5.pdf

Angry IP scanner. (n.d.). Retrieved April 01, 2021, from https://angryip.org/

Average cost of penetration testing. (2020, April 03). Retrieved April 05, 2021, from https://blog.rsisecurity.com/what-is-the-average-cost-of-penetration-testing/

Best advanced online port scanners in 2021. (2021, April 01). Retrieved March 16, 2021, from https://www.softwaretestinghelp.com/port-scanners/

Bishop, M. (2007). About penetration testing. *IEEE Security & Privacy Magazine, 5*(6), 84-87. doi:10.1109/msp.2007.159

Commercially available Penetration Testing. (2006). *Center for the Protection of National Infrastructure (CPNI).*

Cost of a data breach study. (2020). Retrieved April 05, 2021, from https://www.ibm.com/security/data-breach

Farao, E. (2021, March 23). The future of penetration testing in IT security, cost & service. Retrieved March 17, 2021, from https://ermprotect.com/blog/how-artificial-intelligence-will-drive-the-future-of-penetration-testing/

Farmer, D., & Venema, W. (1993). Improving the security of your site by breaking into it. Retrieved from https://www.semanticscholar.org/paper/Improving-the-Security-of-Your-Site-by-Breaking-it-Farmer-Venema/639aa62428bf7c91c5af743c138bfc44a3bed4b4

Firch, J. (2021, March 25). What are the different types of penetration testing? Retrieved April 05, 2021, from https://purplesec.us/types-penetration-testing/#Network

Free port scanner. (n.d.). Retrieved April 05, 2021, from https://www.solarwinds.com/free-tools/port-scanner

Geer, D., & Harthorne, J. (2002). Penetration testing: A duet. *18th Annual Computer Security Applications Conference, 2002. Proceedings.* doi:10.1109/csac.2002.1176290

Johnson, J. (2021, March 18). Amount of monetary damage caused by reported cybercrime to the IC3 from 2001 to 2020. Retrieved April 05, 2021, from https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us/

Hunt, E. (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

Kali Linux penetration testing tools. (n.d.). Retrieved March 18, 2021, from https://tools.kali.org/

Ke, J., Yang, C., & Ahn, T. (2009). Using w3af to achieve automated penetration testing by live dvd/live usb. *Proceedings of the 2009*

International Conference on Hybrid Information Technology - ICHIT '09. doi:10.1145/1644993.1645078

Major limitations of penetration testing you need to know. (2020, May 14). Retrieved April 09, 2021, from https://towardsdatascience.com/major-limitations-of-penetration-testing-you-need-to-know-3f99d2b72c47

Manship, R. (n.d.). What is the primary purpose of penetration testing? Retrieved April 05, 2021, from https://www.redteamsecure.com/blog/the-purpose-of-penetration-testing#:~:text=The%20Main%20Objective%20Of%20A,before%20hostile%20parties%20discover%20them

McDermott, J. P. (2000). Attack net penetration testing. *Proceedings of the 2000 Workshop on New Security Paradigms - NSPW '00,* 15-21. doi:10.1145/366173.366183

Midian, P. (2002). Perspectives on penetration testing — black box vs. white box. *Network Security, 2002*(11), 10-12. doi:10.1016/s1353-4858(02)11009-9

Naik, N. A., Kurundkar, G. D., Khamitkar, S. D., & Kalyankar, N. V. (2009, December). Penetration testing: A Roadmap to Network Security. Retrieved April 07, 2021, from https://arxiv.org/pdf/0912.3970

Osborne, M. (2006). *How to cheat at managing information security*. Rockland, MA: Syngress. Retrieved March 31, 2021, from https://ebookcentral.proquest.com/lib/csusb/reader.action?docID=26620

Penetration testing. (2017, August). Retrieved April 04, 2021, from https://www.ncsc.gov.uk/guidance/penetration-testing

Phong, C. T. (2014). A study of Penetration testing Tools and Approaches. Retrieved April 6, 2021, from http://openrepository.aut.ac.nz/bitstream/handle/10292/7801/ChiemTP.pdf?sequence=3&isAllowed=y

Program rules – application security. (n.d.). Retrieved April 06, 2021, from https://www.google.com/about/appsecurity/reward-program/index.html

Rubens, P. (2009, September 28). L0phtcrack provides industrial Strength Password AUDITING. Retrieved April 07, 2021, from https://www.enterprisenetworkingplanet.com/netsecur/article.php/3841256/L0phtcrack-Provides-Industrial-Strength-Password-Auditing.htm#:~:text=Pricing%20is%20currently%20%24295%20for,%241195%20for%20the%20Consultant%20version.

Salter, C., Saydjari, S. O., & Wallner, J. (1998). Towards A Secure System Engineering Methodology. In proceedings of New Security Paradigms Workshop, Charlottesville, Virginia.

Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Technical guide to information security testing and assessment. doi:10.6028/nist.sp.800-115

Shebli, H. M., & Beheshti, B. D. (2018). A study on penetration testing process and tools. *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. doi:10.1109/lisat.2018.8378035

The history of penetration testing. (2020, November 18). Retrieved April 04, 2021, from https://alpinesecurity.com/blog/history-of-penetration-testing/

Thorsen, C., Nufryk, J., & Taylor, P. J. (2019). *The Official CompTIA PenTest+ Student Guide (Exam PTO-001)*. Downers Grove, IL: CompTIA.

Tiller, J. S. (2016). *CISO'S Guide to Penetration Testing: A framework to plan, manage, and maximize benefits*. Boca Raton, FL: CRC Press.

Wack, J. P., Tracy, M. C., & Souppaya, M. P. (2003). Guideline on network security testing. doi:10.6028/nist.sp.800-42

Weissman, C. (1995, January 24). Handbook for the computer security certification of trusted systems. Retrieved March 16, 2021, from http://www.windowsecurity.com/uplarticle/12/SPM.pdf

What is the future of penetration testing? (2018, March 08). Retrieved March 17, 2021, from https://medium.com/secjuice/theres-been-increasing-debate-online-and-in-the-cybersecurity-sector-recently-over-both-the-future-e931cff5c68d

Wilson, M. (2021, January 15). Best port scanning software & tools for Windows, Linux and Online. Retrieved April 05, 2021, from https://www.pcwdld.com/best-port-scanner-tools#Netcat