
Vulnerability Assessment for Applications Security Through Penetration Simulation and Testing

Petar Lachkov, Lo'ai Tawalbeh* and Smriti Bhatt

*Director of the Cyber Engineering Technology/Cyber Security Research Center,
Department of Computing and Cyber Security, Texas A&M University-San Antonio,
One University Way, San Antonio, TX 78224, USA*

E-mail: plach01@jaguar.tamu.edu; Ltawalbeh@tamusa.edu; sbhatt@tamusa.edu

**Corresponding Author*

Received 23 May 2021; Accepted 08 June 2021;
Publication 28 December 2022

Abstract

Cybersecurity threats and attacks are a critical concern for computing systems as general and specifically in web applications. There are many types and categories of cyberattacks on web applications. Many of these attacks are made possible due to existing vulnerabilities in the networking environments and platforms that host these web applications. So, the vulnerability assessment and attacks simulations on these networking platforms are of extreme importance to protect and secure the top web applications that play a prime role in our daily life. One of the widely used mechanisms to identify vulnerabilities and defend against different attacks on systems and networks is Penetration Testing. It allows us to simulate real-world attacks on a network or a single device to determine the susceptibility and impact of cybersecurity attacks. Pen testing aims to secure a system or network by performing a full-blown attack against it. Several techniques have been used for that, from port scanning, service, and operating system detection to network enumeration, creating specially crafted packets, and modifying

Journal of Web Engineering, Vol. 21.7, 2187–2208.

doi: 10.13052/jwe1540-9589.2178

© 2022 River Publishers

software to exploit vulnerabilities. However, while it is used widely as a defensive technique, some attackers also employ it for malicious intentions utilizing available open-source penetration testing tools. Penetration testing on internal networks such as networks that connect IoT/sensors/web cameras, can be utilized to find vulnerabilities and fix them to secure the networks. In this research, we present a detailed discussion on penetration testing and its seven phases of action and provide a step-by-step procedure with instructions using various open-source tools to conduct penetration testing and vulnerability assessments of a network. We finally demonstrate the process and results of simulated attacks on our network within the testing environment. This research provides a comprehensive introduction to penetration testing and testbed through real-world attack simulation. The IT administrator or security enthusiast can utilize them to secure networks, devices, clients, servers, and applications while enhancing the overall organization's security.

Keywords: Penetration testing, ethical hacking, applications security, firewall, IDS/IPS, server, client, privacy, vulnerability assessment.

1 Introduction

These days the world is comprising of billions of devices, systems, applications, and complex networks. It is critical to effectively and efficiently secure them to defend against various threats and attack vectors. Penetration testing is a mechanism to find the vulnerabilities in a system or a network and analyze multiple security threats to defend against such threats [1]. It is a proactive way to test your system or network to secure it. Mostly, users wonder that maybe the best way to protect your network is to hack it yourself? However, penetration testing is a detailed procedure to unravel all open doors in a system or network. Therefore, ethically hacking a system or network can make them more secure by disclosing all of the vulnerable components and devices within the network. It is critical for any company relying on IT infrastructure to conduct such testing to identify and remediate the vulnerabilities to secure them further.

Mostly, such testing capabilities are provided by companies that specialize in security through attack simulations and have highly skilled professionals who are trained to perform them. There is an agreement between the company requesting the test, and the company performing the test before penetration testing may begin. This Scoping agreement specifies the extent to which the test will be conducted, such as if it will only include port scanning

and service detection along with network enumeration, or if actual attacks need be carried out, which may take down online IT infrastructure and in-turn harm the production environment. Once all the tests are completed, the testing company will provide a detailed report on any vulnerabilities found, including possible mitigation strategies for those vulnerabilities [2]. Testing frequency and extent will vary significantly from company to company. In some cases, simulated attacks can be executed remotely, and in a test environment without the need to disrupt any IT services or production environment.

Penetration testing, also known as ethical hacking, white-hat hacking, or simply as a “pen test” while conducting testing, is the process of performing simulated attacks from the outside in, on a single device or an entire network. Computer security professionals evaluate, hack, and report on a given number of systems, servers, services, or applications within an organization [3]. The report is the most important benefit of conducting these test types, as it provides crucial information to IT infrastructure management. It discloses how the penetration testers can hack critical IT infrastructure and exploit existing vulnerabilities to gain unauthorized access to systems and launch further advanced attacks. The report also provides recommendations on ways to mitigate the vulnerabilities to prevent additional attacks and presents invaluable insight to system administrators in charge of securing networks, services, and IT infrastructure for establishing a more secure environment. The duration and depth of various tests can vary depending on the size of networks being tested or the level of details that are requested. As the complexity and variation in systems, networks, and IT infrastructure increases, so does the length of penetration testing. Sometimes, testing may take several months to complete because of all the different services that need to be tested. Similarly, the cost of testing also scales depending on the extent of testing [4]. Hiring a company to conduct penetration testing can be costly for a small or even large company. Thus, sometimes system administrators would perform these tests internally to save cost as well as secure their infrastructure.

Usually, system administrators who have penetration testing skills are more experienced and are better compensated compared to junior administrators. One of the most important questions for any organization is – what is their desired security level? It depends on various factors, such as the nature of sensitive information being hosted locally or in their cloud, if any critical IT infrastructure is being hosted and run locally or in cloud servers (websites, payment card processing, mail servers), and of course the budget that is available to secure the network and associated services [5]. These factors facilitate an organization’s management in deciding whether

to hire a professional penetration testing company or conduct an internal penetration test.

In this paper, we discuss different phases of penetration testing in detail, then demonstrate the penetration testing and attack simulation process and steps in a testbed environment. We also present comprehensive results for identified threats and vulnerabilities in the network. The main contribution of this work is to provide theoretical information and background on penetration testing. In addition to demonstrating practical test environment setup and penetration testing on our testbed network environment. We first set up our testbed environment and then perform the testing within this network by conducting various tests using open-source tools. Our pen testing architecture aligns within the scope of penetration testing conducted internally within an organization. At the same time, security and IT administrators can understand the testing architecture and quickly extend it on a larger scale based on their needs [6]. This research is an initial step towards providing a penetration testing guideline that can be utilized by individual users as well as IT administrators to test, secure, and monitor their networks. With billions of connected smart devices in the network, this research can enable users of smart devices to understand the security threats in their network, devices, systems, and applications and envision how to secure them. Mostly, users are using devices that store their sensitive personal information such as banking, social security, private files. From an attacker's perspective, they also use readily available open-source pen testing tools to gain knowledge on a target, perform live monitoring of internal network resources, and finally launch attacks against it. Therefore, it is critical to follow the ethical guidelines in conducting penetration testing or ethical hacking.

The rest of the paper is organized as follows: Section 2 discusses a brief background on penetration testing with its different phases and related work. The test environment setup is presented in Section 3. Section 4 illustrates the steps involved and results obtained from penetration testing to secure the network in detail. Finally, Section 5 concludes the paper.

2 Penetration Testing and Vulnerability Assessments

In recent years, penetration testing has become a popular trend in analyzing the security of any system in the industry. It is also referred to as red teaming in academia, where the red team attempts to “break into a system” while the blue team prevents and defends against cyberattacks on a system. However, there are several questions to be answered before attempting penetration

testing on a network or a host machine. It is essential to understand what aspect of the system is being tested and there has to be an agreement between sponsors and testers to identify the specific goals and threats being addressed through penetration testing [1]. This helps to narrow down the scope of testing and focus on desired aspects of the system being tested.

2.1 Penetration Testing Phases

A penetration test has seven phases: (1) *Pre-Engagement Scoping*, (2) *Reconnaissance*, (3) *Threat Modeling/Vulnerability Identification*, (4) *Exploitation*, (5) *Post-Exploitation*, (6) *Reporting*, and (7) *Resolution/Re-Testing* [7].

(1) **Pre-Engagement Scoping:** The first phase of a penetration test is most often overlooked but is of the utmost importance when conducting a penetration test against an organization. In this phase, the test's scope needs to be clearly defined to avoid the disruption of critical services. Usually, the client or sponsor and the testing company will outline and agree on what needs to be tested and how it will be tested. This lays the foundation for the essential aspect of penetration testing, and it is vital to have details on paper to know how to proceed further. Once a Pre-Engagement Scoping agreement has been established and agreed upon by both parties, phase two may begin [7].

(2) **Reconnaissance:** This phase is also known as foot-printing. It is a passive information gathering and preparation step for conducting the actual penetration test. It begins with collecting as much information as possible about the organization (systems and networks) being tested. It is crucial to have a good understanding of the IT infrastructure before beginning the testing. Some of the commonly used methods are, but not limited to, employing a centralized Open-Source Intelligence framework, such as described in OSINT Framework [8], and other specialized tools, such as Nmap, to perform exploration on any one host or network of an organization. More advanced techniques as social engineering, domain name searches, search engine queries, dumpster diving, and website inspection are used to gather information in the reconnaissance phase. Besides, other tools can be used to collect further intelligence. Once this step is completed, and there is a substantial understanding of the target gathered; thus, phase three can be initialized.

(3) **Threat Modeling/Vulnerability Identification:** This phase will utilize all the information gathered in phase two to start designing the threat models based on the context of the organization, its customers, and sensitive data

hosted or stored in the databases. Now, it leads to vulnerability identification, which identifies the vulnerabilities that can be exploited to realize the threat model in a real-world scenario [9]. An important step is to conduct vulnerability assessments through port scanners and fingerprinting to find open ports, enumerate operating systems, and see live hosts to perform testing. Vulnerability scanners such as Nessus or OpenVAS have these capabilities built into them and put the findings into a web interface that is easy to read and understand [10]. For our implementation, we have set up Nessus on a LAN network. We are downloading and installing the Nessus Essentials server and web client on Windows 10 environment. As a penetration tester, it is essential to keep an open mind and always have backup plans. If there is an issue with a mechanism, it is necessary to reevaluate different paths based on identified vulnerabilities and move forward from another feasible route.

(4) **Exploitation:** In this phase, the information gained and analysis from previous stages come into action, where the pen tester will try to exploit the system through various mechanisms. As an example, trying to get admin privileges or getting a shell on the target server. The goal is to exploit all the vulnerabilities and investigate all avenues to compromise the system or network. It is vital to capture screenshots of different stages during the exploit for future phases. Another important aspect of this phase is to identify the depth and scalability of attacks enabled by exploiting the vulnerabilities. The length of this phase is based on the time and resources available for testing.

(5) **Post-Exploitation:** Once the exploitation phase is complete, it is necessary to document the process and methodologies used in the exploitation phase and a list of all the compromised resources (e.g., accounts, systems, applications, etc.). Besides, documenting the nature of compromise and potential damage that was achievable is also essential to capture. For example, the CEO's account credentials being compromised indicates more substantial damage because the organization's sensitive data and information have been accessed. Moreover, a vital step here is to revert the system to baseline configurations by removing any new tools or software used, new accounts or scripts added, or modified settings during pen-testing.

(6) **Reporting:** After the fun activity of breaking into the system, the most critical phase of penetration testing is reporting, especially from a client/sponsor perspective. It involves reporting all the steps and tools used to run exploits during penetration testing and effective mitigation strategies to protect the resources in an organization. A more detailed analysis of security

risks, threats, vulnerabilities, and mitigation strategies can be included in the report as per the client requirements. The client could have an internal or external security team who will then remediate the vulnerabilities and implement the recommended security mechanisms. The overall goal of penetration testing is to find security risks and vulnerabilities in the system and fix them to secure it. Therefore, information and analysis provided in this phase allow clients to secure their organization.

(7) **Resolution and Re-Testing:** Mostly, a penetration test is completed with a final report. However, in some cases, re-testing is done based on the client's need after the security issues and vulnerabilities have been resolved. This phase ensures that the identified vulnerabilities have been fixed.

2.2 Related Work

There is prior work in the literature that discusses penetration testing in different contexts. Here, we present a brief analysis of relevant early work on pen-testing. A detailed introduction of penetration testing was presented in [1], where the author discusses different aspects of penetration testing and what it means. It also discusses several use-case scenarios analogous to penetration testing to show the relevance; for example, a new car is protected by trying to break into the car. Besides, there are different types of penetration testing as well. Penetration Testing in the context of software testing and quality assurance to develop secure software is presented in [11]. Similarly, penetration testing to find security bugs in applications by incorporating specific tests within application-security testing techniques is given in [12], where the author presents different types of testing to find security vulnerabilities. Another early work on penetration testing [13] presented the benefits of using Petri net together with flaw hypothesis and attack tree approaches and proposed an attack net penetration testing approach. The flaw hypothesis means "a flaw (such as vulnerability) as a demonstrated undocumented capability which can be exploited to violate some aspect of the security policy" [14], and attack trees are used in testing when there is limited information on the target system. In [15], authors presented penetration testing from two aspects: science and art, and discussed 5Ws of pen testing – why, who, what, where, and when. Penetration testing has also been applied in various domains, such as advanced metering infrastructure (AMI) [16]. The authors developed an archetypal attack tree approach to guide penetration testing across multiple-vendor implementations of intelligent AMI systems. In this research, we mainly focus on describing the

penetration testing process. Illustrates its phases and benefits, simulating and demonstrating an actual penetration testing with its various phases on the network.

3 Penetration Test Environment

In this section, we discuss the penetration test environment setup along with specific devices and tools used for conducting the penetration test. Here, we develop a semi-automated system that can help system administrators and penetration testers to perform scanning and reporting of vulnerabilities on a network. The authors in [17] illustrate and define such a system. This setup also can be used to conduct full-blown attacks for assessing incident response readiness. The operating system and all tools which are used with this setup are open-source, where users have the permission granted to use and modify the program and its source code as needed. It adds a certain degree of flexibility to the test architecture, and minor changes can be applied to get the desired functionality. This test setup first requires a penetration testing box to be deployed within the intranet of the organization.

In this setup, we have a Raspberry Pi running Kali Linux with the following accessories: storage media, a wireless adapter (built-in for this model), and interface devices ($2 \times$ RJ45 Ethernet). This device sits on the LAN behind the firewall, where it has access to all devices on the network. It continually monitors, scans, and reports on vulnerabilities within all devices. An SSH (Secure Shell) tunnel is established to an external VPS (Virtual Private Server) to enable remote access for further automating the testing process. The security professional can connect to the VPS through the SSH tunnel for any remote location and get inside the network past the firewall into the penetration testing box, which allows us to read any reports about current vulnerabilities and provide the ability to perform further penetration testing. Now, security attacks can be launched to compromise systems and to assess the readiness of system administrators to respond to, mitigate, and recover from attacks. This test environment utilizes an externally accessed internal vulnerability assessment that is conducted on the inside of a network. In contrast, the penetration test is done from outside of a network simulating a real attack. The difference between a penetration test and a vulnerability assessment is that a pen test is conducted from the outside in, and an assessment is done from within the network. Both can provide substantial information on security risks and improvements needed and should be performed regularly to secure an organization's IT infrastructure

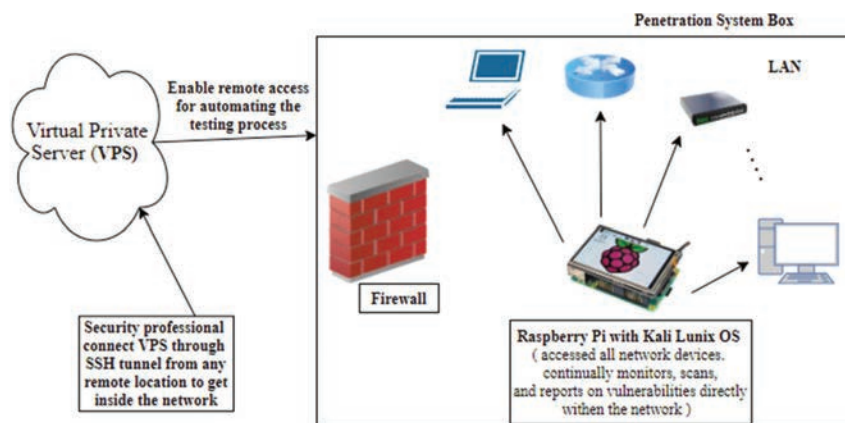


Figure 1 Penetration testing environment.

and sensitive information effectively. Figure 1 shows the penetration testing environment.

4 Penetration Test Execution and Results

This section discusses how we executed the seven phases of penetration testing, as discussed earlier in our testbed architecture. It also presents a detailed step-by-step pen testing through a simulated attack on our internal LAN. In our testing environment, penetration testing is performed on an Internal LAN, and no testing is conducted outside of the network. It is critically important to understand that hacking with criminal intent and without proper permission is illegal, and we do not condone it in any way, shape, or form in our experiments. Some of the mechanisms used in our testing and attack simulations are port scanning, reconnaissance, enumeration, exploitation, etc. using different open-source tools.

As discussed earlier, a successful penetration test can be broken down in seven phases: (1) Pre-Engagement Scoping, (2) Reconnaissance, (3) Threat Modeling/Vulnerability Identification, (4) Exploitation, (5) Post-Exploitation, (6) Reporting, and (7) Resolution/Re-Testing [7]. In the pre-engagement and scoping phase of a penetration test, the testing scope needs to be clearly defined to avoid the disruption of critical services. Usually, the client or sponsor and the testing company will outline and agree on *what needs to be tested and how it will be tested?* So, it is important to describe our penetration testing scope, which laid the foundation for the testing to

be done. In the reconnaissance phase, passive information on the network and connected devices are gathered through various tools in preparation for conducting the actual penetration test. It encompasses collecting as much information as possible about the network being tested using multiple tools.

Next, in the threat modeling and vulnerability identification phase, all the data gathered during the reconnaissance phase will pay off and come into good use. We used port scanners and vulnerability assessments to find open ports, enumerate operating systems, services, and discovered live hosts to perform testing. Vulnerability scanners such as Nessus, or OpenVAS with these capabilities built into them and can be viewed in a web interface, as mentioned earlier [10]. We have setup Nessus on the LAN by downloading and installing the Nessus Essentials server and web client on a Windows 10 Home installation. This service performs vulnerability assessment triggered through the web client. It is started through an administrator shell on Windows with the command – “*net start ‘Tenable Nessus’*”. The web client can be accessed by going to “*localhost:8834*” address in the web browser [18], which opens its web interface. We performed the “*Basic Network Scan*” vulnerability scan on our entire test environment (192.168.1.0/24) subnet. This scan not only identified open ports and services running on those ports but provided detailed information on vulnerabilities that are discovered on those ports by identifying specific service version running on that port.

Figure 1 displays the results of a completed scan by Nessus and the discovered vulnerabilities along with their severity level. The most severe one found is in the router, which has an outdated Apple Filing Protocol (AFP) service, which allows remote code execution (remote shell vulnerability). We used an updated Netgear router, yet this vulnerability was discovered. The router’s firmware is re-flashed with a custom one such as OpenWRT or DD-WRT to secure it. This technique will significantly improve the router’s security, functionality, and reliability while providing a solution for this critical vulnerability. There is another software that we used to scan for open ports, service, and operating system detection on the local network.

The steps involve starting a Kali Linux Virtual Machine on Windows 10 host machine, and then running NMAP (Network Map) in Command Line Interface (CLI) in Kali Linux, and performing a manual scan with the flags: “*nmap -A -sV -sT 192.168.1.0/24*”. The findings are listed in Figure 2 (Host details: 192.168.1.1, the Netgear Router), where live hosts are identified and enumerated in a report. We have also created a table combining the results from both the Nmap and Nessus scans which is shown in Table 1. It illustrates all the network devices and their details. The CLI report by NMAP lists hosts

```

root@kali:~# nmap -A -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-19 13:42 EST
Nmap scan report for 192.168.1.1
Host is up (0.0026s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       dnsmasq 2.78
|_ dns-nsid:
|_   NSID: res310.mia.rrdns.pch.net (7265733331302e6d69612e7272646e732e7063682e)
|_   id.server: res310.mia.rrdns.pch.net
|_   bind.version: dnsmasq-2.78
80/tcp    open  http         Apache/2.4.18 (Ubuntu)
|_ http-auth:
|_   HTTP/1.0 401 Unauthorized\x0D
|_   Basic realm=NETGEAR R8300
|_ http-title: 401 Unauthorized
548/tcp   open  afp          Netatalk 2.2.5 (name: kaiz3network; protocol 3.3)
|_ afp-serverinfo:
|_   Server Flags:
|_     Flags hex: 0x8f79
|_     Super Client: true
|_     UUIDs: true
|_     UTF8 Server Name: true
|_     Open Directory: true
|_     Reconnect: false
|_     Server Notifications: true
|_     TCP/IP: true

```

Figure 2 Kali Linux – NMAP CLI.

Table 1 Consolidated NMAP + Nessus table for Internet Network 192.168.1.0/24

IP	OPEN PORTS	SERVICES	HOSTNAME/OS	VALNERABLE
192.168.1.1 (Router)	53, 80, 139, 445, 548, 631, 5000, 9100, 9101, 9102, 9103, 20005	DNS, HTTP, SMBD, AFP, IPP, UPnP, HP, JetDirect, NetUSB	KAIZ3N/ LINUX	2 LOW (DHCP, ETHERLEAK) 1 MEDIUM (SMB SIGNING), 1 HIGH (NETATALK)
192.168.1.2	1080, 8888	SOCKS5 Proxy, ALHTTTPD	AMAZON/ LINUX	None
192.168.1.3	1080, 8888	SOCKS5 Proxy, ALHTTTPD	AMAZON/ LINUX	None
192.168.1.4	None	None	ROOMBA-980/ iRobot embedded	None
192.168.1.6	None	None	UNKNOWN iOS	None
192.168.1.9	80, 443, 631, 8080, 9100, 9220	HTTP, HTTPS, IPP, HT, TP Proxy, HP, JetDirect, HP-GSG	HP/ Wind River VXWORKS	None
192.168.1.10	None	None	SAMSUNG ELECTRONICS	None
192.168.1.22	135, 139, 445, 5357	MSRPC, NETBIOS-SSN, M-DS, HTTP	KAIZ3N/ TOP/ WINDOWS	1 MEDIUM (SMB SIGNING)
192.168.1.24	None	None	UNKNOWN/ LINUX	None
192.168.1.253	139, 445, 2691	SMBD, SSH	KAIZENRIG, LINUX	1 MEDIUM (SSH SIGNING) 1 LOW (SSH CBC)
192.168.1.254	53, 80, 2691	DNS (PIHOLE), HTTP, SSH	KAIZ3NPI, LINUX	None

that are discovered in the subnet along with open ports and service including their versions that are running on each device, and their operating systems as well. This tool is very convenient, easy to set up, and implement on any network, and can help secure the LAN by enumerating the devices. These early phases of penetration testing are intended to help us collect as much information about the target network as possible and use this information to identify the threats and vulnerabilities. Some of the critical questions in these phases are: *How many hosts are in the network? What ports are open? What services are running, and what version of each service is running? Is there a firewall or an IDS/IPS (Intrusion Detection/Prevention System) system?*

The next phase is **exploitation**, and this is when the real hacking begins. Now, we must start to think like a hacker and start asking questions like – *What assets are within the organization?, How many people work there, what are their roles?, How many different departments are there? Can any employees be used to social engineer our way into the network? What are the customers of this organization, and what kind of data can we gather on them, and do they have any access to the organization's systems?* At this point, we can build some charts and document all the information gathered on the organization, and possible attack vectors to exploit further for penetration testing.

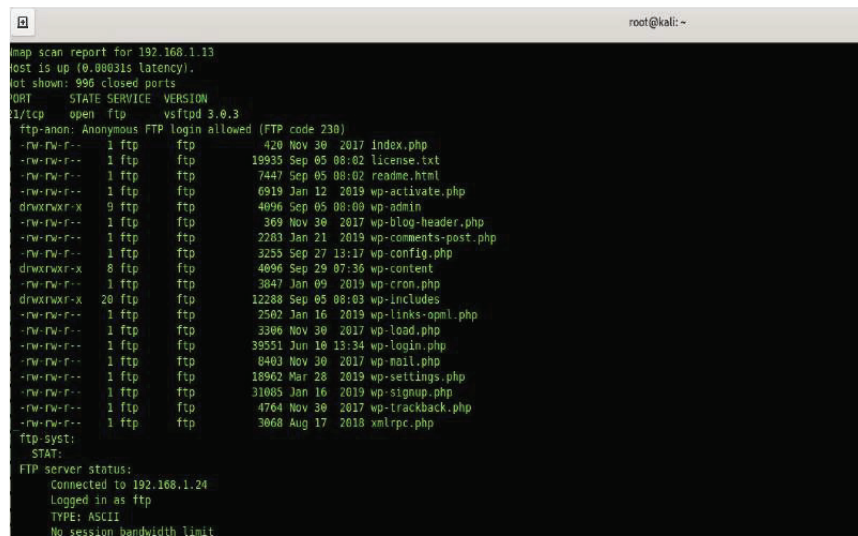
We can now begin exploiting vulnerabilities and trying to gain access to a system by using various tools, techniques, and methods. Next, we look up for an instance, the version of AFP that is running on the vulnerable router and how to exploit it. For this, we use a search engine such as Google, or it's more privacy-focused counterpart, DuckDuckGo to search for the CVE (Common Vulnerabilities & Exposures) number which was discovered during the Nessus vulnerability scan: *CVE-2018-1160*. Furthermore, another web page ("*NIST/NVD*") was found, which describes this vulnerability in further detail, including about ten different links to resources that explain its exploitation and remediation.

From the network and vulnerability scan results, we found that there is a machine on the internal network sitting at 192.168.1.253 named 'kaiz3nrig' which is running the service Samba (smbd) on ports 139 and 445. Next, there is another machine at 192.168.1.4 named 'kaiz3ntop' that is running Samba on port 445 as well. Furthermore, exploring the Nessus scan shows that there is a 'SMB Signing not required' vulnerability involving digital SMB signing in medium severity, on both of these machines. It would allow an unauthenticated, remote attacker to exploit this vulnerability and conduct a man-in-the-middle attack against the SMB server. This vulnerability could

be exploited further but requires an extensive setup that is outside the scope of this research paper. To further simulate, perform, and report on penetration testing on our internal network, we deployed a vulnerable VM that is already misconfigured, similar to a machine in a production environment would be with many open ports and vulnerabilities.

Here, the goal is to demonstrate how to exploit a known vulnerability by exploiting this VM. For this exploitation, we utilize a popular open-source tool, Metasploit, which is relatively easy to setup and takes an intermediate technical knowledge to use. While Metasploit is all CLI based, there is another software known as Armitage, which is precisely the same but with a GUI. There are various resources available on Metasploit [19]. To begin exploiting this vulnerable VM, first, we need to find its IP address in the internal network, which can be done through NMAP as shown earlier. Using the same command, we scan the whole subnet 192.168.1.0/24 and find the IP address and all information we can gather on it. The NMAP scan results show the VM's IP address as 192.168.1.13, with the following running services and open ports: *vsftpd on port 21, OpenSSH on 22, Apache httpd on 80, and Miniserv webserver on 10000* as shown in Figure 3.

From the scan report, we also discovered that the FTP server allows anonymous guest logins, which when probed, has disclosed that there is a WordPress website being hosted by it. There are several references in the

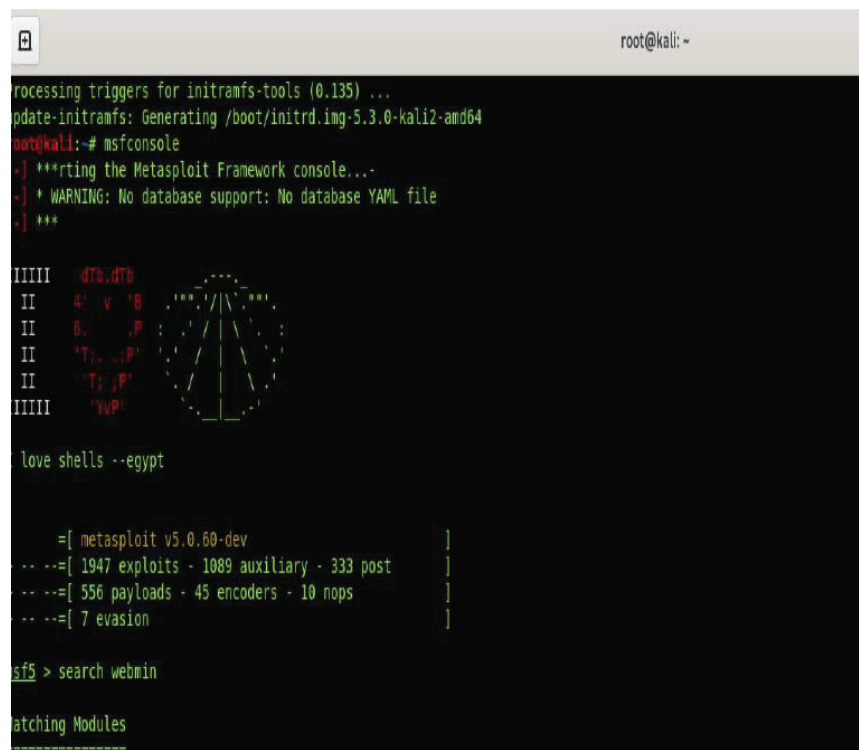


```

root@kali: ~
nmap scan report for 192.168.1.13
Host is up (0.00031s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
ftp-anon: Anonymous FTP login allowed (FTP code 230)
-rw-rw-r-- 1 ftp      ftp      420 Nov 30 2017 index.php
-rw-rw-r-- 1 ftp      ftp      19935 Sep 05 08:02 license.txt
-rw-rw-r-- 1 ftp      ftp      7447 Sep 05 08:02 readme.html
-rw-rw-r-- 1 ftp      ftp      6319 Jan 12 2019 wp-activate.php
drwxrwxr-x 9 ftp      ftp      4096 Sep 05 08:09 wp-admin
-rw-rw-r-- 1 ftp      ftp      369 Nov 30 2017 wp-blog-header.php
-rw-rw-r-- 1 ftp      ftp      2283 Jan 21 2019 wp-comments-post.php
-rw-rw-r-- 1 ftp      ftp      3255 Sep 27 13:17 wp-config.php
drwxrwxr-x 8 ftp      ftp      4096 Sep 29 07:36 wp-content
-rw-rw-r-- 1 ftp      ftp      3847 Jan 09 2019 wp-cron.php
drwxrwxr-x 20 ftp     ftp      12288 Sep 05 08:03 wp-includes
-rw-rw-r-- 1 ftp      ftp      2502 Jan 16 2019 wp-links-opml.php
-rw-rw-r-- 1 ftp      ftp      3306 Nov 30 2017 wp-load.php
-rw-rw-r-- 1 ftp      ftp      39551 Jun 10 13:34 wp-login.php
-rw-rw-r-- 1 ftp      ftp      8403 Nov 30 2017 wp-mail.php
-rw-rw-r-- 1 ftp      ftp      18962 Mar 28 2019 wp-settings.php
-rw-rw-r-- 1 ftp      ftp      31085 Jan 16 2019 wp-signup.php
-rw-rw-r-- 1 ftp      ftp      4764 Nov 30 2017 wp-trackback.php
-rw-rw-r-- 1 ftp      ftp      3068 Aug 17 2018 xmlrpc.php
ftp-syst:
STAT:
FTP server status:
Connected to 192.168.1.24
Logged in as ftp
TYPE: ASCII
No session bandwidth limit

```

Figure 3 NMAP discovery of vulnerable VM “Hackerfest 2019”.



```

root@kali: ~
processing triggers for initramfs-tools (0.135) ...
update-initramfs: Generating /boot/initrd.img-5.3.0-kali2-amd64
root@kali:~# msfconsole
[*] Starting the Metasploit Framework console...
[*] * WARNING: No database support: No database YAML file
[*] ***

IIIII  dTb.dTb
II    4:  v  'B
II    6:  .  'P
II    'T:  'P'
II    'T:  'P'
IIIII  'VVP'

love shells --egypt

    =[ metasploit v5.0.60-dev
  -- ==[ 1947 exploits - 1089 auxiliary - 333 post
  -- ==[ 556 payloads - 45 encoders - 10 nops
  -- ==[ 7 evasion

msf5 > search webmin

atching Modules
=====

```

Figure 4 Metasploit framework console: “Webmin” vulnerability search.

scan report that disclose the operating system of the VM, such as “*OpenSSH 7.4p1 Debian 10+deb9u7*”. The system is most likely running Debian 9 Stretch, if not that particular version, but for sure, we know that it is running Debian Linux. Now, when we know the services running, next, we can look for specific vulnerabilities by browsing available resources, such as website “<https://www.cvedetails.com/>” and enter particular software and its version. Here, we checked MiniServ v1.890 against the database and found a backdoor vulnerability with a CVSS (Common Vulnerability Scoring System) score of 10. Moreover, there seems to be a Metasploit module that exploits this same vulnerability; thus, we try and execute it to gain access to the machine. The steps are as follows. (i) To start Metasploit, type ‘‘msfconsole’’ in the Kali CLI. (ii) The framework begins, and we can type ‘‘search webmin’’ to see all of the available exploits for the keyword ‘Webmin’ that are loaded in the database of Metasploit. (iii) The results show five exploits, as shown in Figure 4. (iv) From the results, one of the

```

[*] 192.168.1.13 - Command shell session 1 closed. Reason: User exit
msf5 exploit(unix/webapp/webmin_backdoor) > exploit

[*] Started reverse TCP handler on 192.168.1.24:4444
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 2 opened (192.168.1.24:4444 -> 192.168.1.13:50166) at 2019-11-20 14:34:53 -0500

```

Figure 5 Gaining a shell on the target system 192.168.1.13: “Webmin”.

vulnerabilities in the CVE-2019-15107, which we tried to exploit. We can say that this is the correct exploit which we want to run from the description of it in the Metasploit console as “*Webmin password_change.cgi Backdoor*”.

In order to exploit this vulnerability, the following command is used from inside the framework: “*use exploit/unix/webapp/webmin_backdoor*”. First, we need to check all options that can be used with this module by typing “*options*”, then we use “*rhosts 192.168.1.13*” to target the web server. After conducting the reconnaissance phase, we know that “*Webmin*” uses HTTPS communications on the web front end, so we need to execute “*set ssl true*”, to match this setting in our module [19].

Next, we need to set the host where our backdoor will open and be accessed from by typing the command “*set lhost 127.0.0.1*”. After the module is configured correctly, the next step is to exploit the target system by typing “*exploit*”. After executing the command, we successfully exploited vulnerability. Figure 5 shows the module being executed for the exploit to target the IP address of our vulnerable VM at 192.168.1.13. We have gained an administrator, root-level shell on the target system. This can be confirmed by typing “*id*”, and checking what user is currently logged in.

Figure 6 lists the root directory “*/*” with administrator access of the Virtual Machine that was just exploited.

The system is compromised at this point, and it is possible to retrieve sensitive information stored on the database running the WordPress installation or simply erase everything and kill the server leading to a denial of service attack for this server. Another approach would be to plant a backdoor for continued access and listen to all activity and communication associated with this system. It can also be used to collect more information, unknown to the administrator, to launch future attacks on the network or other systems. Similarly, the tester has the option to try and exploit the network further by performing a lateral movement into another VLAN, system, or possibly into the router where any system connected on the network could be accessed.

```

root@kali: ~
drwxr-xr-x  4 root root  4096 Sep  9 13:32 xinetd
-rwxr-xr-x  1 root root 7709 Jul 15  2018 xmlrpc.cgi
ls -la /
total 00
drwxr-xr-x 22 root root 4096 Oct  7 07:31 .
drwxr-xr-x 22 root root 4096 Oct  7 07:31 ..
-rw-r--r--  1 root root 1024 Sep 10 08:13 .rnd
drwxr-xr-x  2 root root  4096 Oct  7 07:33 bin
drwxr-xr-x  3 root root  4096 Oct  7 07:31 boot
drwxr-xr-x 17 root root 3140 Nov 20 18:24 dev
drwxr-xr-x 80 root root 4096 Nov 20 18:24 etc
drwxr-xr-x  4 root root  4096 Sep 10 08:58 home
lrwxrwxrwx  1 root root   36 Oct  7 07:06 initrd.img -> boot/initrd.img-4.19.0-0.bpo.6-amd64
lrwxrwxrwx  1 root root   36 Oct  7 07:31 initrd.img.old -> boot/initrd.img-4.19.0-0.bpo.6-amd64
drwxr-xr-x 13 root root  4096 Oct  7 07:03 lib
drwxr-xr-x  2 root root  4096 Aug  6 08:25 lib64
drwx-----  2 root root 16384 Aug  6 08:24 lost+found
drwxr-xr-x  2 root root  4096 Aug  6 08:25 media
drwxr-xr-x  4 root root  4096 Sep  9 12:19 mnt
drwxr-xr-x  3 root root  4096 Oct  7 07:03 opt
dr-xr-xr-x 95 root root    0 Nov 20 18:24 proc
drwx-----  2 root root  4096 Oct  7 07:38 root
drwxr-xr-x 19 root root   580 Nov 20 18:24 run
drwxr-xr-x  2 root root  4096 Nov 20 18:24 sbin
drwxr-xr-x  3 root root  4096 Sep 10 08:48 srv

```

Figure 6 Inside a admin-level shell on target system listing volumes of root directory “/”.

After the exploitation, the next phase is Post-Exploitation, and this is when all methods that were used are documented. It is recommended that while the tests are being conducted, keep a chronology of everything performed with screenshots, such as *the device being exploited, type of test, the time it ran*, and other factors. Therefore, at the end, when a report is being created, all of the information is readily accessible. At this time, it necessary to revert the system or network to its baselines configuration by cleaning the system of any new scripts or files that were used on the hosts, or new accounts added, etc. It essential to include written detailed information on all the methods that were tried and all the ones that worked or even didn't work for documentation purposes. This helps to prepare for the next phase: Reporting. At this point, a very detailed report needs to be written on how the vulnerabilities were discovered, and exactly how they were exploited so that IT management personnel could patch them and prevent an actual attacker from exploiting these vulnerabilities. It is critical to write this report with precision, resiliency, and exactness to make sure it is understandable and legible. In this paper, we document and demonstrate our results based on the outcome of all the testing done up to this point. This is similar to the report that company management will see when the findings are turned in, but possibly with more detailed analysis based on the client's requirements.

On the other hand, the authors in [20] investigated the security and penetration of Open SSH on Raspberry Pi2. They introduced very detailed

testing and penetration process that exploited in details the possible vulnerabilities associated with Open SSH on constrained environments such as Raspberry Pi's. Moreover, the authors in [21] analyzed the D-Wave Quantum Macro Assembler Security. Their analysis is considered important at the area of Macro Assemblers. Their findings and results can be utilized by the developers to enhance the security of such assemblers.

The objective of this research is to enable system administrators and IT management to enhance the security posture of the company employing the penetration testing, specially when these networks are implemented as Software Defined Networks [22]. In the final pen testing report, every little detail needs to be included along with the steps for each successful exploitation, and the actions taken to get inside the host also need to be disclosed. The client/sponsor should be able to easily understand what is most critical and what needs to be addressed depending on the scope and extent of the testing. Next, the penetration test is concluded with the last but not least Resolution and Re-Testing phase. During this phase, written recommendations on remediation for the vulnerabilities discovered need to be compiled. The more detailed this phase is, the better the solution for remediation will be and will be most useful for the management. In an organization, depending on the extent of testing being done, time can be granted for administrators to address these issues, after which a re-testing of the hosts would be done to make sure the patches are useful. Furthermore, based on organization's request, the penetration testers can also guide the company and IT administration staff throughout the process of securing vulnerable systems as needed and in accordance with NICE cybersecurity framework [23].

5 Conclusion

In this paper, we discussed a penetration test from start to finish and disclosed all the necessary phases in conducting pen testing successfully and efficiently. It is relatively easy to enumerate a network, perform scans to find vulnerable hosts, and then exploit them to compromise a system. The information presented in this paper can be used to secure networks, firewalls, servers, clients, and applications. At the same time, it can also be used for malicious purposes; however, it is not the intent of this work, and any hacking or even scanning done without permission is not ethical and legal and may result in felony convictions, restitution fees, and prison terms. Penetration tests, also known as ethical hacking, are always conducted with proper permissions and

agreement with client and tester organizations with a beneficial outcome of securing the client organization and its resources, Penetration testing is done ethically and correctly is an exceptional method of testing network or system posture and security. There are numerous benefits to organizations that have such testing done on a regular basis. Most companies will have such testing done regularly such as quarterly, bi-annually, or annually. It is recommended to have it done at least quarterly, if not monthly, as the dynamics of IT infrastructure within an organization change and become re-configured very frequently today with new technologies. Penetration Testing is an essential part of keeping an organization secure and resilient to networks attacks and should be conducted often. To truly secure a network, system, service or application, we need to think from both defensive and offensive perspectives, and this paper discusses an offensive mechanism, penetration testing, in detail, and how it can be employed to achieve efficient and effective defensive security for an organization.

Acknowledgement

This work was supported by the Texas A&M University System Chancellor Research Initiative (CRI) grant awarded to Texas A&M University-San Antonio, TX, USA.

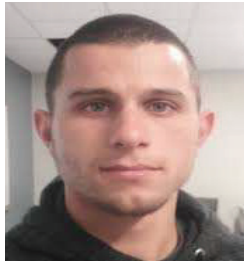
References

- [1] Bacudio, Aileen G., Xiaohong Yuan, Bei-Tseng Bill Chu, and Monique Jones. "An overview of penetration testing." *International Journal of Network Security & Its Applications* 3, no. 6 (2011): 19.
- [2] Zaldivar, David, A. Tawalbeh Lo'ai, and Fadi Muheidat. "Investigating the security threats on networked medical devices." In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0488–0493. IEEE, 2020.
- [3] Al-Haija, Qasem Abu. "Autoregressive modeling and prediction of annual worldwide cybercrimes for cloud environments." In *2019 10th International Conference on Information and Communication Systems (ICICS)*, pp. 47–51. IEEE, 2019.
- [4] "Current CVSS Score Distribution For All Vulnerabilities." CVE Security Vulnerability Database. Security Vulnerabilities, Exploits, References and More. Last accessed April 6th, 2021. <https://www.cvedetails.com/cve/CVE-2019-15107/>

- [5] Lo'ai, A. Tawalbeh, Hala Tawalbeh, Houbing Song, and Yaser Jararweh. "Intrusion and attacks over mobile networks and cloud health systems." In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 13–17. IEEE, 2017.
- [6] AlDairi, Anwaar. "Cyber security attacks on smart cities and associated mobile technologies." *Procedia Computer Science* 109 (2017): 1086–1091.
- [7] "Seven Penetration Testing Phases to Achieve Amazing Results." CyberX,. Last accessed April 9th, 2021. <https://cyberx.tech/penetration-testing-phases/>.
- [8] OSINT Framework." OSINT Framework, <https://osintframework.com/>.
- [9] Jararweh, Yaser, Haythem A. Bany Salameh, Abdallah Alturani, Loai Tawalbeh, and Houbing Song. "Anomaly-based framework for detecting dynamic spectrum access attacks in cognitive radio networks." *Telecommunication Systems* 67, no. 2 (2018): 217–229.
- [10] "NVD/NIST" CVE-2018-1160 Detail, 12/20/2018, <https://nvd.nist.gov/vuln/detail/CVE-2018-1160>
- [11] Arkin, Brad, Scott Stender, and Gary McGraw. "Software penetration testing." *IEEE Security & Privacy* 3, no. 1 (2005): 84–87.
- [12] Thompson, Herbert H. "Application penetration testing." *IEEE Security & Privacy* 3, no. 1 (2005): 66–69.
- [13] McDermott, James P. "Attack net penetration testing." In *Proceedings of the 2000 workshop on New security paradigms*, pp. 15–21. 2001.
- [14] WEISSMAN, C. Penetration Testing. In Handbook for the Computer Security Certification of Trusted Systems. Naval Research Laboratory Technical Memorandum 5540:082a, 24 January 1995.
- [15] Geer, Daniel, and John Harthorne. "Penetration testing: A duet." In *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, pp. 185–195. IEEE, 2002.
- [16] McLaughlin, Stephen, Dmitry Podkuiko, Sergei Miadzvezhanka, Adam Delozier, and Patrick McDaniel. "Multi-vendor penetration testing in the advanced metering infrastructure." In *Proceedings of the 26th Annual Computer Security Applications Conference*, pp. 107–116. 2010.
- [17] Epling, Lee, Brandon Hinkel, and Yi Hu. "Penetration testing in a box." In *Proceedings of the 2015 Information Security Curriculum Development Conference*, pp. 1–4. 2015.
- [18] Security Focus Netatalk CVE-2018-1160 Arbitrary Code Execution Vulnerability, last accessed April 1st, 2021. <https://www.securityfocus.com/bid/106301>.

- [19] Petters, Jeff. “What Is Metasploit? The Beginner’s Guide – Varonis.” Inside Out Security, Last Accessed April 4th, 2021, <https://www.varonis.com/blog/what-is-metasploit/>.
- [20] H. H. Alsaadi, M. Aldwairi, M. Al Taei, M. AlBuainain and M. AlKubaisi, “Penetration and Security of OpenSSH Remote Secure Shell Service on Raspberry Pi 2,” *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018, pp. 1–5, doi: 10.1109/NTMS.2018.8328710.
- [21] Alsaadi H.H., Aldwairi M., Muller-Stuler EM. (2019) Analyzing D-Wave Quantum Macro Assembler Security. In: Latifi S. (eds) 16th International Conference on Information Technology-New Generations (ITNG 2019). Advances in Intelligent Systems and Computing, vol. 800. Springer, Cham. https://doi.org/10.1007/978-3-030-14070-0_19
- [22] AlEroud, Ahmed, and Izzat Alsmadi. “Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach.” *Journal of Network and Computer Applications* 80 (2017): 152–164.
- [23] Easttom C. (2020) Vulnerability Assessment and Management. In: The NICE Cyber Security Framework. Springer, Cham. https://doi.org/10.1007/978-3-030-41987-5_12

Biographies



Petar Lachkov Graduated from Department of computing and Cybersecurity at Texas A&M University with honors degree. His research interests include web applications security, Privacy, Cyber attacks simulations, vulnerabilities assessment.



Lo'ai Tawalbeh (IEEE SM): Completed his PhD degree in Electrical & Computer Engineering from Oregon State University in 2004, and MSc in 2002 from the same university with GPA 4/4. Dr. Tawalbeh is currently a tenured Associate professor at the department of Computing and Cyber Security at Texas A&M University-San Antonio. He also worked as R&D engineer at the leading digital design company SYNOPSYS, OR, USA. Before that he was a visiting researcher at University of California-Santa Barbra. Since 2005 he taught/developed more than 30 courses in different disciplines of computer engineering and science with focus on cyber security for the undergraduate/graduate programs at: NewYork Institute of Technology (NYIT), DePaul's University, and Jordan University of Science and Technology. Dr. Tawalbeh won many research grants and awards with over than 2 Million USD. He is supervised more than 30 Graduate students (PhD and MSc). He has over 130 research publications in refereed international Journals and conferences. <https://orcid.org/0000-0002-2294-9829>



Smriti Bhatt is an Assistant Professor of Computer Science in the Department of Computing and Cyber Security at Texas A&M University-San Antonio. Dr. Bhatt is teaching cybersecurity and computer science courses in the department. She has received her Ph.D. in Computer Science from the University of Texas at San Antonio and did her doctoral research at the Institute for Cyber Security (ICS) and Center for Security and Privacy Enhanced Cloud Computing (C-SPECC). Dr. Bhatt's research expertise is in the field of Cyber Security, mainly focused on Access Control and Communication Control models, and Security and Privacy in Cloud Computing and Internet of Things (IoT). Her current research projects focus on developing secure access control and communication control models for Cloud-Enabled Internet of Things architecture applicable to various IoT domains, such as Smart Home, Smart Health, and Wearable IoT.