Swinburne University of Technology

Faculty of Science, Engineering and Technology

ASSIGNMENT AND PROJECT COVER SHEET

Unit Code: COS30015 Unit Title: IT Security

Assignment number and title: Assignment 2 Due date: 30th October 2022

Lab group: Thursday 12:30 EN305. Tutor: Jamie Ooi Lecturer: Mr. Lin Li

Family name: Efaz Identity no: 103494172

Other names: Ekrar Uddin Mohammed

To be completed if this is an INDIVIDUAL ASSIGNMENT

I declare that this assignment is my individual work. I have not worked collaboratively, nor have I copied fromany other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for me by another person.

Signature: Ekrar Efaz.

Marker's comments:

Total Mark:

General Server Hardening

COS30015 IT Security

Assignment 2 Practical Report

Ekrar Efaz (103494172)

Submission Date: 8th September
Submission Due Date-Time: 11:59 pm 30th October

1. Introduction

This paper presents a plan and implementation of secure configuration of a Linux webserver during and after installation. We analyse the current security threats and state affecting a public facing webserver and attempt to follow best practices of a renowned hardening guide to establish security. The effectiveness of the guide is evaluated by performing various attempts of breaching security and analysing how effective the security implementation is.

1.1. Scenario

In this paper we are working with a local job advertisement small business. Recently, to expand their user base to young adults they have employed a web server on Lenovo machine running Oracle Linux that hosts their website using Apache. They have dealt with a few security breaches since their start and now are looking to secure their webserver to protect the data of their users and maintain accessibility.

1.2. General Server Threats

ACSC 20-21 Annual Cyber Threat report points that there was 13% increase of cybercrime reports from the past year. The threat actors are rapidly exploiting new and old unpatched vulnerabilities to gain malicious access to devices and critical information [1]. The most common forms of attack to a webserver are listed below [2]:

a. Denial of Service
b. Probing
c. Directory Traversal Attacks
d. SSH Brute Force Attack
e. Server Application Exploits

The rise of DDoS attacks is exponential in terms of frequency and duration. The Q2 2022 saw an average of 3000 minutes of DDoS attack which pales the Q2 of 2021 attack duration of 30 minutes [3]. Thus, DDoS poses the highest threat to a public facing server. Probing is done as a pre-attack measure to find exploitable services and open ports to conduct an attack. Public facing web servers often forget to deny traversing their internal directories which the attacks exploit to gain privileged access on the server. If the web server is maintained or access remotely via SSH the SSH brute forcing is one of the most popular attacks for armatures and scriptkiddies.

1.3.Threat Actors

Threat actors are those who aim to exploit vulnerabilities in Information Systems. They are categorized by their motivation, and they value access to devices and networks for different reasons [4]. Web servers mostly face a few specific categories of actors with either discontent or profit motivation. Sometimes there are people who do it for satisfaction.

1.4. Small Business Perception of Internet Threat

Small businesses often assume they are too small to come up on an attacker's radar. They think their data although very valuable to them and their clients would spark no significant interest in a malicious attacker. Unfortunately, today's cyber space practices equal opportunity and as such small businesses are on the same radar with more vulnerable

systems. A study suggests, 20% of studies small businesses faced 6-10 cyber incidents while around 12 percent faced more than 10 incidents in a year [5].

2. Security Practices for Server Security

2.1. Scope

This document addresses the servers that provide general services over the network communication as primary service. The servers addressed are the public outward facing servers such as Web services or Email services. The servers that use general Operating Systems such as Linux are the primary target of this document. The aim of the document is to follow the certain guidelines and implement security policies and assess the effectiveness of the security policy through security analysis and evaluation.

2.2. Document Reference

The hardening guidelines followed for this security evaluation is the *NIST Guide to General Server Security* [6] and *NIST Accepted Principles for securing Information Technology Systems* [7].

2.3. Security Practices

This document aims to provide an evaluation of the server security implemented following the hardening guideline. The following security policies are applied [6]:

- Securing Server Hardware
- Securing Server Software
- Securing Server Remote Access

2.4. Used Technologies

- Hardware Security Tools: *RFID Reader/Writer*
- Server OS: *Oracle Linux*
- Server Hardware: *Lenovo ThinkPad*
- Server Software: *Apache 2.5.1*
- Network Analysis: *Nmap, Wireshark, Net-Tools*
- Access Control: *PAM, firewalld, SSHGuard*
- Security Tools: *Metasploit*
- Pentesting OS: *Kali Linux, Ubuntu*

3. Security Policy Implementation

3.1. Securing the Server Hardware

Access Control to server hardware is an important part of security. Protecting Physical assets from outside threats is known as hardware security also refers to protection from harms such as fire, short-circuits etc. There are 3 layers of server hardware security [8].

a. Authorized Building Access and Monitoring.
b. Authorized Server Room Access.
c. Authorized Server Rack Access.

To ensure hardware security in three layers we take the following measures [7]:

3.1.1. Installing Secure Server Rack

***Defence*:**

There are various forms of server racks that prevent unauthorized access to server components. Whether hosting a server in a shared space or private space limiting access to servers even within the organization is important to ensure layered security.
Server racks provide locks that can only be opened using dedicated keys providing controlled access to servers.

Fig: server rack with lock.

3.1.2. Access Controls for server room

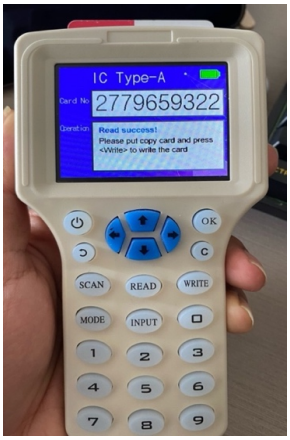
Security of server rooms is an ongoing process. Ensuring only authorized person can access the data centres and multiple layers of authentication are employed to adopt the principle of least privilege.

Defence:

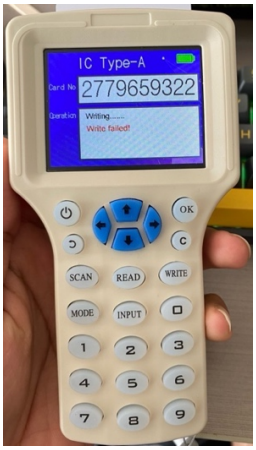
RFID tags and cards are a good way to authenticate users and keep logs on who accessed the data centre to trace back footprints in the advent of a breach. RFID tags emit low frequency radio waves to the antenna which is then converted to data to aid in authentication.
RFID tags although are a convenient way to authenticate users they are often compromised by market RFID readers/writers and Self-programmed RFID listeners. Market RFID readers/writers can only be used to read unencrypted RFID tags making them unsecure.

Attack:

I have tested the security of unencrypted RFID tags by using my own RFID reader/writer. I got hands on to one tag that provides access to my server room and copied the RFID onto my own tag and used it to access the server room.



Encrypted RFIDs are much harder to read with general market reader/writer. Such RFIDs need to be decrypted before reading using special software and often such techniques are employed by attackers with a direct target-based attack.



3.2.Securing Server Software

Server Software: Apache 2.4.51

```
[sysadmin@localhost html]$ httpd -v
Server version: Apache/2.4.51 (Oracle Linux Server)
Server built:   Aug  1 2022 00:00:00
[sysadmin@localhost html]$
```

A partially configured server should never be exposed to the external network. Every server that is insecure would be compromised within minutes after it is put on the network. Although it is not possible to fully harden the software before deployment, so it is best practice to implement incremental security overtime.

To secure server software we employ the following steps:

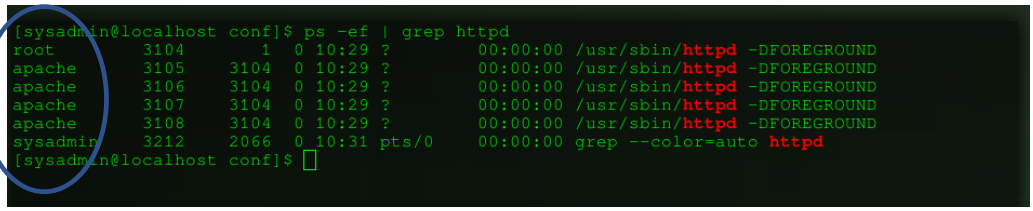
- Install server on a dedicated hardware
- Remove or disable all unneeded default user accounts and have separate user accounts for server process.
- Remove all sample content., scripts and test files in the server.
- Disable directory traversing and version advertisement by the web server.
- Employ DDoS protection.

3.2.1. Separate Accounts for Apache Process

Isolating processes with their own user accounts and own set of permissions to run helps reduce damage caused by break-ins. This configuration will allow for maximum security if we add in least privilege policy.

Defence:

a. *Changing the Apache configurations file, we specified a user and group account for the process to run.*
b. *We implement least privilege policy for the user account by removing it from sudoers group and limiting the commands the user can use.*
c. *We then check our running processes to check if Apache is running using the specified user.*



3.2.2. Disable Directory traversing and Version advertisement

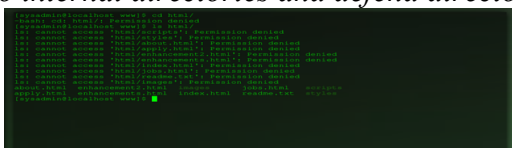
Defence:

a. *Directory traversing and version advertisement banner can be disabled form the Apache config files.*

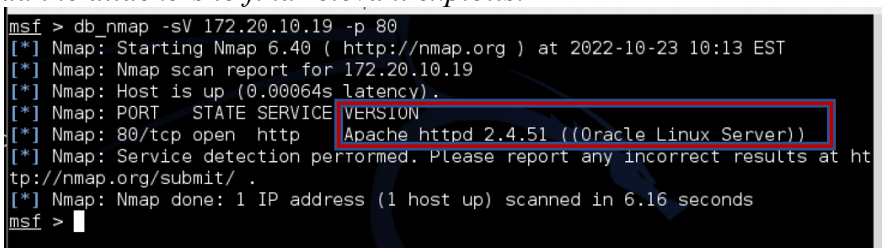
b. *Add the command*

```
<Directory /etc/httpd/htdocs>
Options None
</Directory>
```

to the Apache config files we can disable directory traversing which will deny users access to internal directories and defend directory traversal attacks

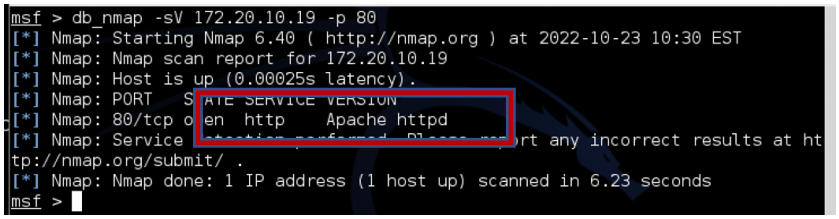


c. *Then we add supplemental configs to the end of the Apache config to disable version advertisement banner. Otherwise, when scanned with network analysis tools the version of the running Apache software and few other details are shown which then might lead the attackers to find relevant exploits.*

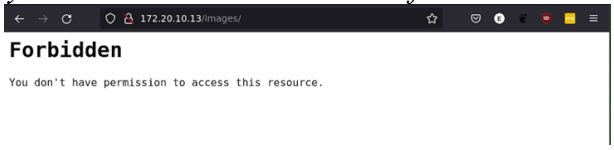


Attack:

a. *I try to grab the banner using **nmap** specifying the version discovery option. Comparing to the previous scan results we can see no version information or the host OS information.*



b. *Then I try to access internal directories from the browser to which I get denied access.*



3.2.3. Employing DDoS protection

DDoS is one of the most common attacks on a web server. Employing DDoS protection has both advantages and disadvantages. It's hard to tell legitimate traffic from illegitimate

traffic and we might end up blocking legitimate users from accessing our services. Still the kind of threat DDoS poses the pros of DDoS protection overweighs the cons.

Defence:

a. *I install the **mod_evasive** Apache module that offers a strong protection against DDoS and brute force attacks* [9].
b. *mod_evasive module traces the IP and pages requested by that IP to the server and once the set threshold limit is reached that IP is blocked.*
c. *I configure the mod_evasive configuration file and add the threshold values to protect the server against DDoS. I configure the following mod_evasive parameters.*

DOSPageCount = *threshold number of page request per IP.*
DOSSiteCount = *threshold number of concurrent request per IP per site.*
DOSPageInterval = *interval for page count .*
DOSSiteInterval = *interval for site count.*
DOSBlockingPeriod = *duration for which IPs are blocked from site.*

```
# This is the threshhold for the number of requests for the same page (or
# URI) per page interval.  Once the threshhold for that interval has been
# exceeded, the IP address of the client will be added to the blocking
# list.
DOSPageCount        2

# This is the threshhold for the total number of requests for any object by
# the same client on the same listener per site interval.  Once the
# threshhold for that interval has been exceeded, the IP address of the
# client will be added to the blocking list.
DOSSiteCount        50

# The interval for the page count threshhold; defaults to 1 second
# intervals.
DOSPageInterval     1

# The interval for the site count threshhold; defaults to 1 second
# intervals.
DOSSiteInterval     1
```

c. *I enable email notification in an event of DDoS I get notified via email.*

```
# The blocking period is the amount of time (in seconds) that a client will
# be blocked for if they are added to the blocking list.  During this time,
# all subsequent requests from the client will result in a 403 (Forbidden)
# and the timer being reset (e.g. another 10 seconds).  Since the timer is
# reset for every subsequent request, it is not necessary to have a long
# blocking period; in the event of a DoS attack, this timer will keep
# getting reset.
DOSBlockingPeriod   10

# If this value is set, an email will be sent to the address specified
# whenever an IP address becomes blacklisted.  A locking mechanism using
# /tmp prevents continuous emails from being sent.
#
# NOTE: Requires /bin/mail (provided by mailx)
DOSEmailNotify      ekrar.efaz@gmail.com
```

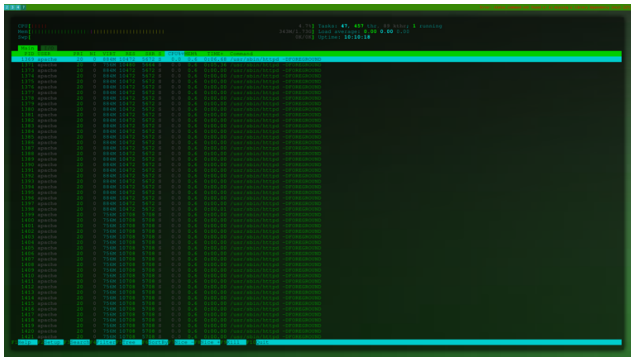
Attack:

I attempt a DDoS attack using a python script and monitor the processes before and after mod_evasive is activated and try to access the website during the DDoS attack. The noticed differences were significant.

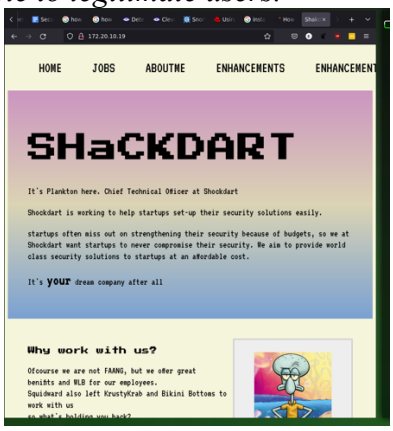
a. *I use a python script to perform a DDoS on the unprotected server with 30000 socket requests.*

```
┌─[ekrar]──[longislandicetea]  [/home/ekrar/exploits/ddos-slowloris/SlowLoris] [22-10-23 10:
04] git:(master)
└──> python3 slowloris.py --host 172.20.10.19 --port 80 --max-sockets 30000
```

b. *The server gets overwhelmed with requests and tries to process every request thus the website becomes inaccessible for legitimate users.*



c. *Attacking the same server with mod_evasive results in a very different output. The concurrent DDoS requests get denied after the set threshold. Thus, leaving the website accessible to legitimate users.*



3.3.Securing Server Remote Access

Remote administration bears a lot of risks and should be enabled considering the risks involved. Remote administration should be allowed on via VPNs and computers on the internal network. If it is determined that it is necessary to remotely access a server then the following steps, ensure that the access is implemented in a secure mechanism [6]:

3.3.1. Use strong authentication mechanism (asymmetric cryptography)

Defence:

Basic defence measures were taken via configuration of SSH config files. I changed the default configurations of the SSH config file and denied password authentication. Authentication is only possible via asymmetric keys. The following steps are taken to ensure strong authentication using asymmetric key.

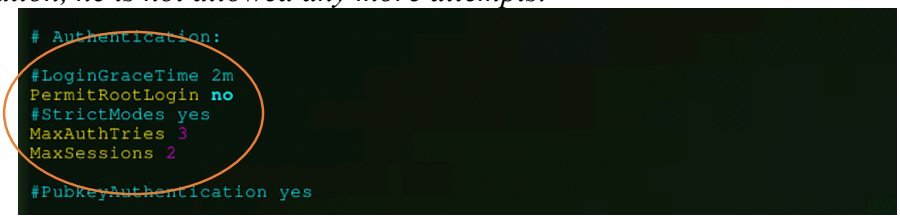
a. *I changed default SSH port by editing the configuration files.*

```
#semanage port -a -t ssh_port_t -p tcp 2025
#
#Removed Default Port
#Port 2025
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

b. *I disabled password authentication to protect against brute forcing. Only asymmetric key authentication is permitted.*

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
PermitEmptyPasswords no
```

c. *I restricted root remote access to ensure least privilege policy is maintained.*
d. *I limited max authentication attempts = 3 and max session = 2. If any user fails authentication, he is not allowed any more attempts.*

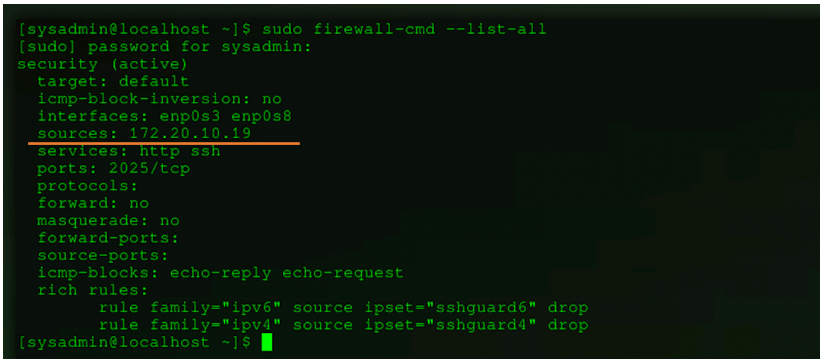


3.3.2. Restrict which hosts are allowed remote access

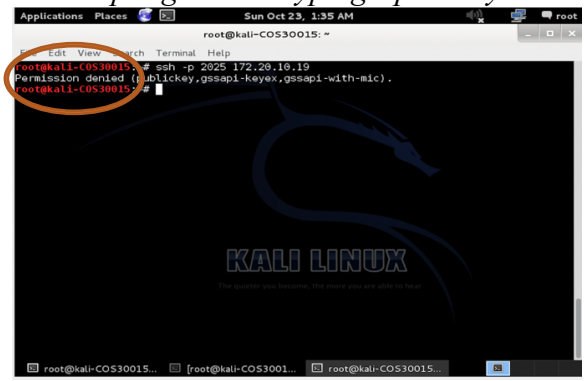
Defence:

Limiting the users that are allowed to remotely administer the server is a strong security measure. This can be implemented using firewall services on the server OS.
I use firewalld service to configure a default zone which allows connections based on whitelisted IP address. The users that are allowed access should have their corresponding private key to the public key stored in their device because password authentication is also not allowed.

a. *I use firewalld service to allow SSH service. Firewalld allows us to whitelist IP using* ***–add-source <IP>*** *and only connections from those IPs are allowed. Any other IP connection is dropped.*
b. *I added trusted IP addresses using firewalld and only those IPs are granted connection.*



c. *Any other IP attempting to use cryptographic keys would be refused connection.*



3.3.3. Enabling SSHGuard to protect against SSH brute force attacks.

Defence:

a. *I install SSHGuard on my server OS and activate it. SSHGuard is essentially a brute force defence that drops incoming multiple failed attempts to authenticate from a single IP.*

b. *I make necessary changes to the configuration files for SSHGuard.*

```
#### REQUIRED CONFIGURATION ####
# Full path to backend executable (required, no default)
BACKEND="/usr/lib/x86_64-linux-gnu/sshg-fw-iptables"

# Shell command that provides logs on standard output. (optional, no default)
# Example 1: ssh and sendmail from systemd journal:
LOGREADER="LANG=C /bin/journalctl -afb -p info -n1 -o cat SYSLOG_FACILITY=4 SYSLOG_F

#### OPTIONS ####
# Block attackers when their cumulative attack score exceeds THRESHOLD.
# Most attacks have a score of 10. (optional, default 30)
THRESHOLD=30

# Block attackers for initially BLOCK_TIME seconds after exceeding THRESHOLD.
# Subsequent blocks increase by a factor of 1.5. (optional, default 120)
BLOCK_TIME=120

# Remember potential attackers for up to DETECTION_TIME seconds before
# resetting their score. (optional, default 1800)
DETECTION_TIME=1800

# IP addresses listed in the WHITELIST_FILE are considered to be
# friendlies and will never be blocked.
WHITELIST_FILE=/etc/sshguard/whitelist
```

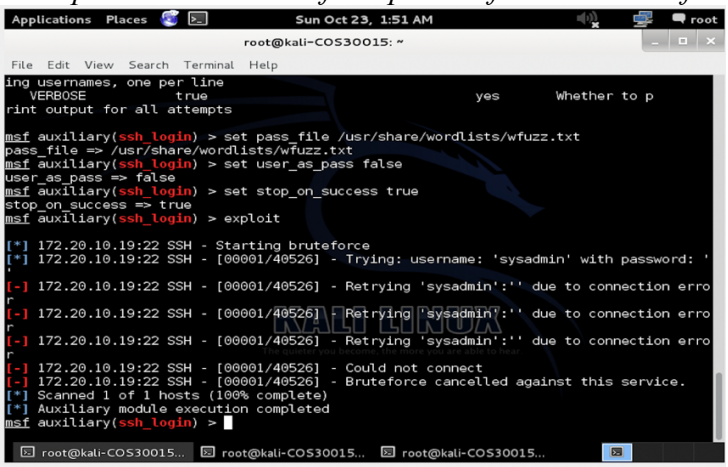
Attack:

a. *SSH brute force is done to verify the implemented security policies are effective in protecting the service. Bruteforcing can be done using various pentesting tools available such as **hydra** and **Metasploit** modules.*

b. *Using Metasploit Module **ssh_login** and a password dictionary available for kali called **rockyou.***

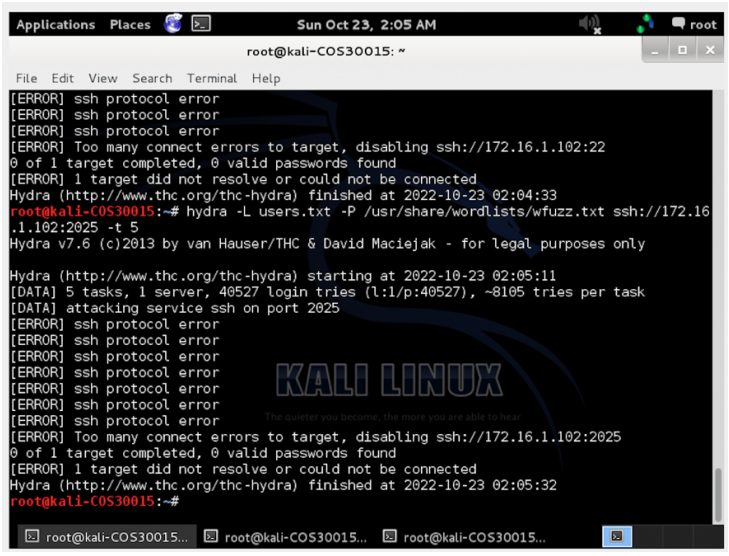
c. *Then I set the target IP and password, username file for the execution.*

d. *The default exploit options attack the default port 22 for SSH which fails to connect.*



e. *Changing the default attack port to 2025 we attempt the attack again with the following unsuccessful results.*

f. *Further attempts are made to brute force **ssh** server using a different tool named **hydra**. After specifying the port, ip and password dictionary the exploit is executed only be denied access again.*



4. Analysis

4.1.Hardware Security

Employing layered security for hardware is a good approach for security. Employing security racks for servers are a good way to ensure that no malicious actor gains access to the terminal. But as we know the server rack is protected using a physical lock which can be access using a key. Physical locks are not much secure if a person with lock picking skill can reach the server by passing all other security it would be the easiest job for him. Server locks should employ biometric authentication method which increases the security exponentially. Biometrics are unique to a person and copying biometric signature is a more complex skill than lockpicking. Keys are easily lost or stolen which might pave way to future unauthorised access, but biometrics are a part of a person which isn't lost or stolen.

We also must keep in mind that *Availability* is also important. For example, we can't use 5 locks on the server rack which would delay our response time in case of any emergency maintenance that is affecting clients or server uptime.

4.2.Server Software Security

Server software is one of the largest attack surfaces for attackers. Server software if not securely configured poses the greatest threat to server security. By updating the Apache configuration files I manage to somewhat secure the server software from probing. One of the pre-attack measures attackers take is to discover existing vulnerabilities on running services in the target machine. Most services running on the machine advertise their version and build information which the attacker might find useful during exploitation. From a defender's point of view, it's much easier to disable version advertisement than to defend against software vulnerabilities and zero-day exploits.

DDoS attacks are very common among web servers and there exists various open-source scripts to use for DDoS. Protection against DDoS is as easy as installing extension modules provided by the software vendors. It is very easy to setup for small businesses and provides excellent protection against small to medium less sophisticated DDoS attacks. There is always the possibility of sophisticated directed attacks that would take down a server even with DDoS protection, but they are rare for small businesses.

Fully securing a server software on the part of system administrators is not possible. Software security is also dependent on the vendor of software. There might be software vulnerabilities in the code which a sysadmin might not be responsible for fixing and similarly there might be a zero-day for the software for which no defence would work.

4.3.Remote Administration Security

The security landscape of Remote administration is very bright. Remote administration is a method that is widely used since remote jobs are getting more and more popular. Work-from-home workers need to access company data and tools via remote access every day and it certainly draws attention from a security standpoint. If remote access is not secured there a malicious user might gain access to company network and perform destructive activities.

By ensuring proper strong authentication methodologies are adopted and connection to the company network/computer is provided following the principle of least privilege and zero-trust models we still can't guarantee there won't be hacks.

Further measures should be taken to protect company data and network by employing VPNs for remote access users and setting up brute force/exploit preventive measures such IDS/IPS and honeypots [10].

From an attacker point of view, I will always attempt to brute force and using phishing to get access remotely. Phishing is one of the most powerful tools to employ against human vulnerabilities. Even though all security policies are strict if an employee gives away his computer access to a RAT everything else fails. Therefore, it is also important to train employee in security awareness and social engineering techniques.

5. Evaluation

The documented attempt to harden a public facing server deals with major threats in the current cyberspace. This paper states how general server security withholds against everyday cyber threats but doesn't address specialized attacks such as social engineering and zero-day exploits.

Hardware Security: *The foundation of any security measure should involve securing the platform the service is hosted on. In scope of this document the hardware platform is the server. Hardware security provides a stronger foundation for the software security. The attempted hardware security in this document was through securing the server component with secure racks, securing the server room with RFID/Biometric access, monitoring and logging of hardware access.*

Servers lock racks are not the technology of the future, they have several vulnerabilities. There are more secure options where enclosed racks with biometric authentication are being developed. Generic server racks are prone to attacks by lock pickers and professional pickers wouldn't need much time to figure out the combination of the lock. RFID tags can also be cloned without the permission or knowledge of the authorized owner and the duplicate can be used to gain unauthorized access to hardware. Although there are newer security protocols in place which make new RFID a better option than traditional RFID. There have been many advances in the RFID security landscape which is making it much harder to clone and read. The proposed A-SRAC protocol provides security for RFID tags [11]*. Additional data is inserted into RFID tags to uniquely identify the beholder and any clones that are made.*

Biometric are the best option among all access control security protocol. Biometrics are unique to individuals and copying biometrics is not very simple as cloning a RFID. But even biometrics also has its own flaws. There have been reports of biometrics not working

well in some conditions such as wet fingers for fingerprint scanners and people with dry fingertips peeling not being able to authenticate.

Software Security: *Software security is shouldered on two parties; one is the software vendor who is responsible for deploying patches and fixing reported bugs and vulnerabilities; other is the system administrator who is responsible for securely configuring the server to minimise security holes. Securing the server software involves a lot of steps which can only ensure security to a certain point.*
Software vulnerabilities are always being discovered and zero-day exploits are the worst of nightmares for small businesses with limited services. The documented security policy and implementation as seen by the exploits attempted can secure the software from generic everyday attacks and pre-attack measures taken by hackers. Further advanced securing would render the software hard to use as we know security has its trade-offs.

Remote Administration Security: *The systems responsible for remote administration should be able to withstand the diverse ever emerging attacks. For security of remote administration, this paper suggests the use of secure shell (SSH) because it provides encrypted communication between two connections. Configuration of SSH for security also has been done keeping in mind that password authentication is much less secure than cryptographic authentication. Although there are exploits that can bypass cryptographic authentication, its best to employ the more secure option.*
Securing SSH against brute force attacks is a prime security policy which is undertaken by most sysadmins. A securely configured remote administration system is very hard to break into and provides excellent functionality for the users.
There have been major developments in SSH security considering it has much popular use. SSH certificates although have been around for a few years hasn't been widely adopted. It's like cryptographic keys by which users can be identified by the server but provides one big advantage that is validity of certificates. The advantage of validity can be seen in assigning short time access to contractors or if an employee loses his certificate, its validity expires which mitigates the risk of compromised key [12].

Works Cited

[1] Australian Cyber Security Center, "ACSC Annual Cyber Threat Report 20-21," Australian Cyber Security Center, 2021.

[2] T. Panhalkar, "InfosSavy," 2019. [Online]. Available: https://info-savvy.com/web-server-attacks/. [Accessed 15 October 2022].

[3] A. Gutnikov, O. Kupreev and Y. Shmelev, "Securelist by Kaspersky," 2022. [Online]. Available: https://securelist.com/ddos-attacks-in-q2-2022/107025/. [Accessed 19 October 2022].

[4] Canadian Center for Cybersecurity, " An introduction to the cyber threat environment," Government of Canada, 28 October 2022. [Online]. Available: https://cyber.gc.ca/en/guidance/introduction-cyber-threat-environment. [Accessed 28 October 2022].

[5] S. Keller, A. Powell, B. Horstmann, C. Predmore and M. Crawford, "INFORMATION SECURITY THREATS AND PRACTICES IN SMALL BUSINESSES," *Information Systems Management,* vol. 22, no. 2, pp. 7-19, 2005.

[6] S. Karen, W. Jansen and M. Tracy, "Guide to General Server Security," *National Institute of Standards and Technology,* vol. 800, no. 123, 2008.

[7] S. Marianne and G. Barbara, "Generally Accepted Principles and Practices for Securing Information Technology Systems," National Institute of Standards and Technology, Washington, 1996.

[8] T. James, "International Security Journal," 29 May 2020. [Online]. Available: https://internationalsecurityjournal.com/physical-security-for-servers-is-more-important-than-ever/. [Accessed 20 October 2022].

[9] L. Vijayakumar, "Prevent DDoS in Apache – Steps to safeguard your web server from DDoS," bobcare, 7 January 2019. [Online]. Available: https://bobcares.com/blog/apache-prevent-ddos/. [Accessed 8 October 2022].

[10] R. Watson, "SSH Hardening Tips to Prevent Brute-Force Attacks," 14 January 2022. [Online]. Available: https://goteleport.com/blog/ssh-hardening-to-prevent-brute-force-attacks/. [Accessed 03 October 2022].

[11] A. W. Sr., L.-S. Tsay, I. A. Kateeb and L. Burton, "Solutions for RFID Smart Tagged Card Security Vulnerabilities," in *AASRI Conference on Intelligent Systems and Control*, North Carolina, 2013.

[12] A. Morris, "Using SSH Certificates Instead of SSH Keys," Venafi, 26 September 2022. [Online]. Available: https://www.venafi.com/blog/what-future-ssh-certificates. [Accessed 26 October 2022].

[13] B. Michael and S. Murugiah, "Hardware-Enabled Security for Server Platforms," NIST.