Adam Sandberg

ITAS 292

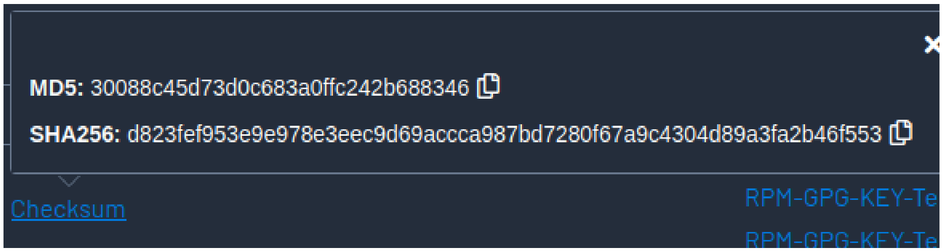
2023-02-06

Lab 5 - Desktop and Server OS vulnerabilities

Table of Contents

Part 1 - Testing commercial vulnerability scanners (Nessus)

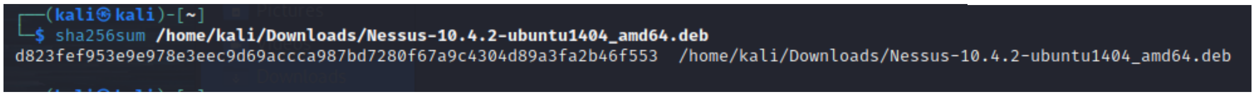
Nessus essentials was installed from Tenable, and the signature was checked.



MD5: 30088c45d73d0c683a0ffc242b688346

SHA256: d823fef953e9e978e3eec9d69accca987bd7280f67a9c4304d89a3fa2b46f553

Figure 1 Nessus Checksum



```
┌──(kali㉿kali)-[~]
└─$ sha256sum /home/kali/Downloads/Nessus-10.4.2-ubuntu1404_amd64.deb
d823fef953e9e978e3eec9d69accca987bd7280f67a9c4304d89a3fa2b46f553  /home/kali/Downloads/Nessus-10.4.2-ubuntu1404_amd64.deb
```

Figure 2 Downloaded File Checksum

As shown in figure one and two, the sha256 signatures match. The program was then installed and launched using its web browser address.

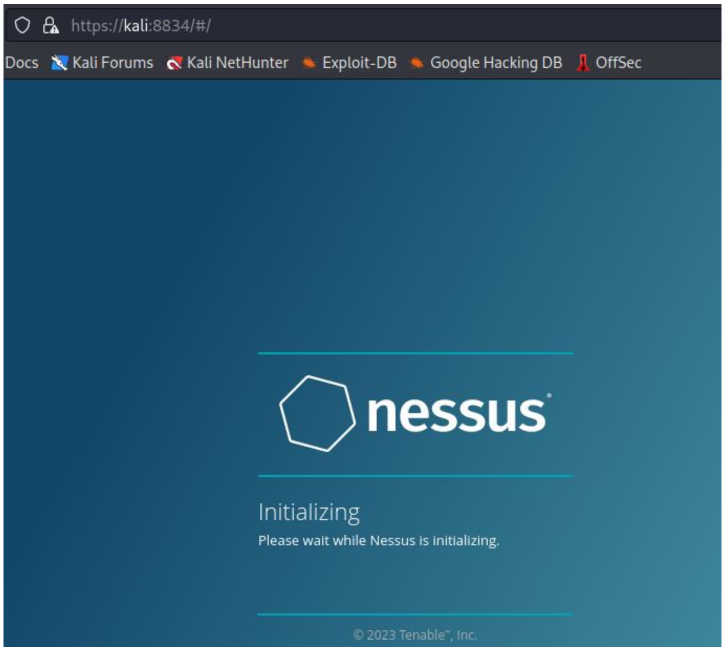
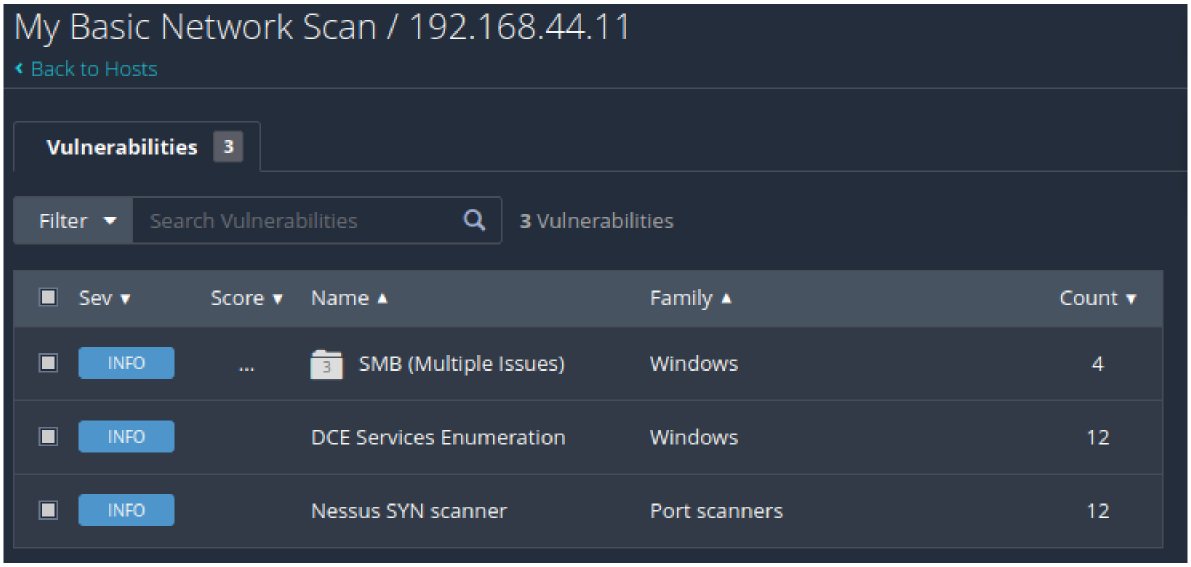


Figure 3 Nessus WebUI

I scanned my known vulnerable windows server first.



My Basic Network Scan / 192.168.44.11

‹ Back to Hosts

Vulnerabilities 3

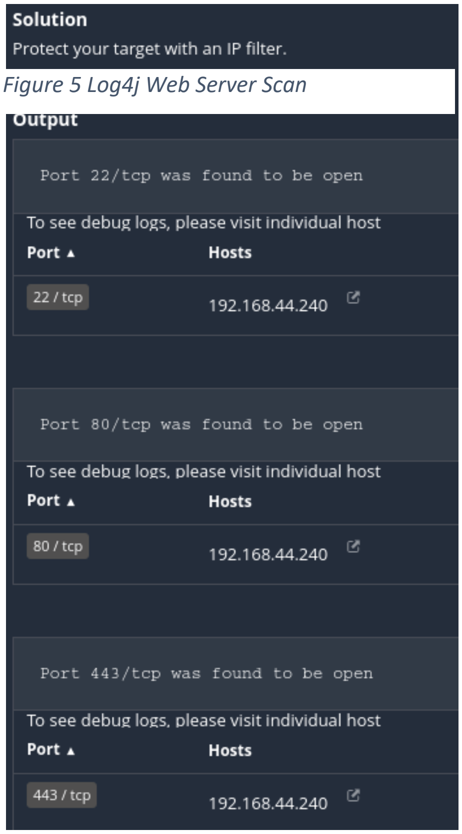
Filter ▾ Search Vulnerabilities 3 Vulnerabilities

□	Sev ▾	Score ▾	Name ▴	Family ▴	Count ▾
□	INFO	...	3 SMB (Multiple Issues)	Windows	4
□	INFO		DCE Services Enumeration	Windows	12
□	INFO		Nessus SYN scanner	Port scanners	12

Figure 4 Scan of Unpatched Windows Server

Multiple SMB vulnerabilities were detected as expected.

Next one of my Linux web servers was scanned. This server was my Apache server. My Apache server was used since Nessus offers a Log4j checker which is meant to be tested against Apache web servers.



Solution

Protect your target with an IP filter.

Figure 5 Log4j Web Server Scan

Output

```
Port 22/tcp was found to be open
```

To see debug logs, please visit individual host

Port ▴	Hosts
22 / tcp	192.168.44.240

```
Port 80/tcp was found to be open
```

To see debug logs, please visit individual host

Port ▴	Hosts
80 / tcp	192.168.44.240

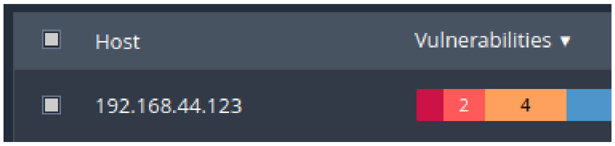
```
Port 443/tcp was found to be open
```

To see debug logs, please visit individual host

Port ▴	Hosts
443 / tcp	192.168.44.240

My server was not vulnerable to Log4j, but it did report on the fact that my webserver ports were open. Which is to be expected. The port 22 being open is only open to the private net as well so that's fine.

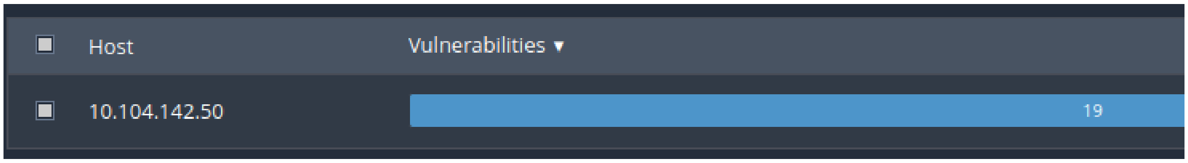
Next my Metasploitable VM was targeted. This showed many results, as expected.



□	Host	Vulnerabilities ▾
□	192.168.44.123	2 4

Figure 6 Metasploit VM Vulnerabilities

And finally, a production windows domain controller was scanned.



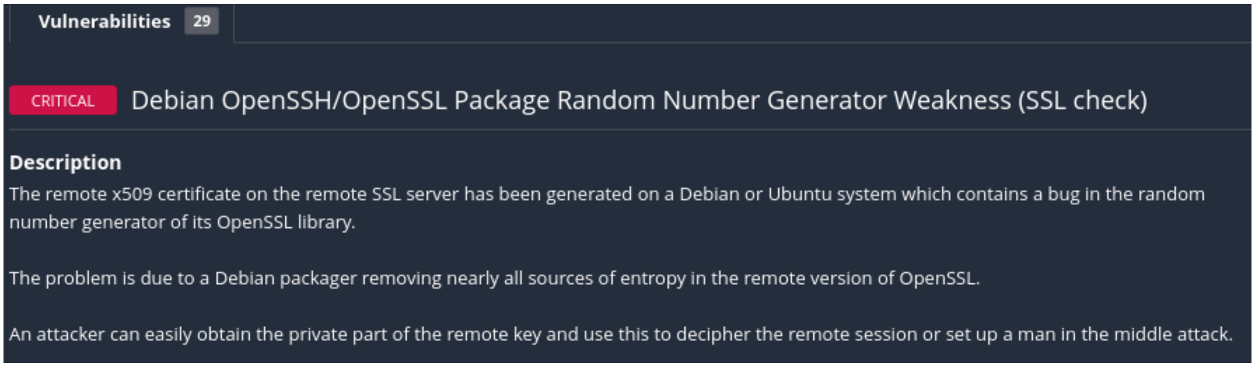
□	Host	Vulnerabilities ▾
□	10.104.142.50	19

Figure 7 Production Win DC Scan

This scan showed no vulnerabilities, which is good since this is a patched and up to date windows server.

Briefly document some of the critical/high risk notifications your scans find:

Most of the critical/high risk notifications came from my Metasploit VM so they will be looked at.



Vulnerabilities 29

CRITICAL Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)

Description

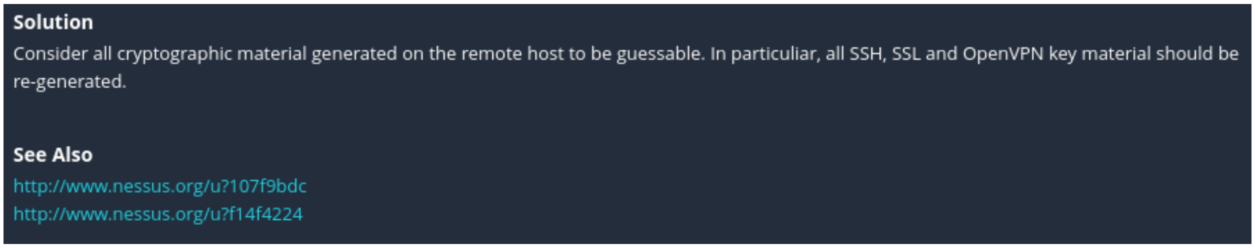
The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a man in the middle attack.

Figure 8 OpenSSH Critical Vulnerability

This was a critical alert that was made for my Metasploit VM. Some remediation links were provided as well as the solution to the critical issue.



Solution

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

See Also

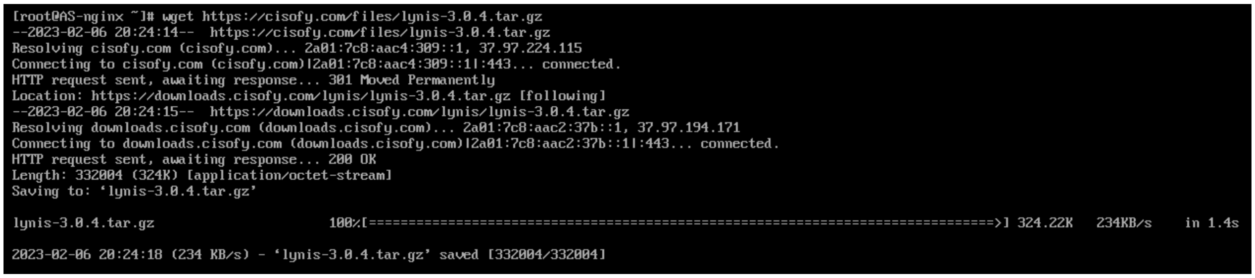
http://www.nessus.org/u?107f9bdc
http://www.nessus.org/u?f14f4224

Figure 9 Vulnerability Patch Solution and Links

I don’t think any of my scans for this VM were false positives the ones I would maybe consider to be were flagged appropriately either with medium or low risk.

Part 2 - Working with Lynis to scan Linux

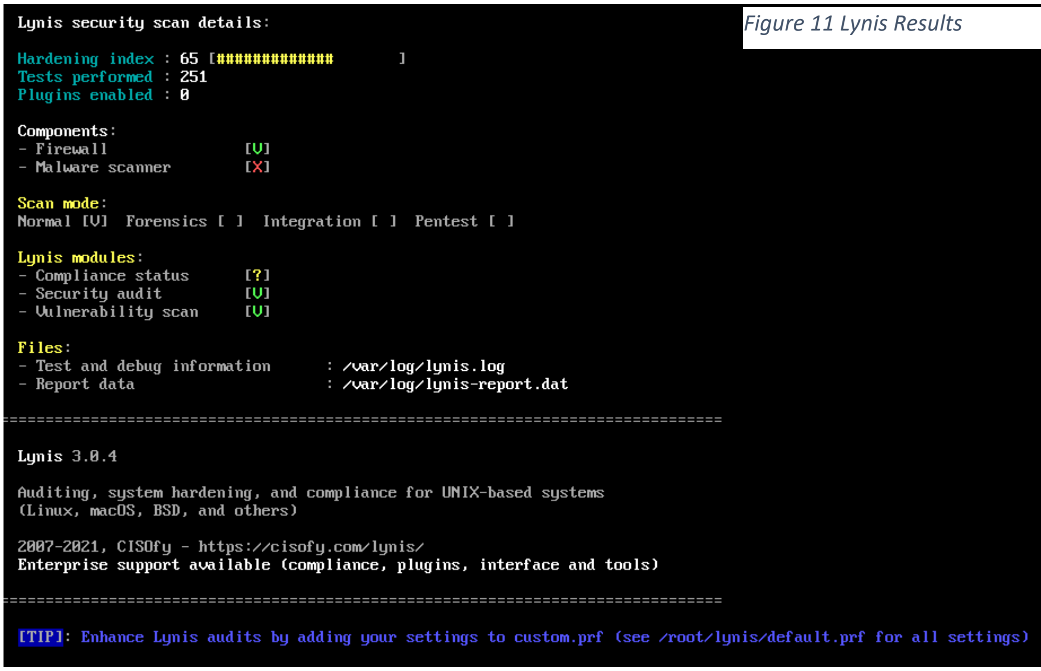
On my Nginx Rocky Linux server. Lynis was installed using wget.



```
[root@AS-nginx ~]# wget https://cisofy.com/files/lynis-3.0.4.tar.gz
--2023-02-06 20:24:14--  https://cisofy.com/files/lynis-3.0.4.tar.gz
Resolving cisofy.com (cisofy.com)... 2a01:7c8:aac4:309::1, 37.97.224.115
Connecting to cisofy.com (cisofy.com)|2a01:7c8:aac4:309::1|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://downloads.cisofy.com/lynis/lynis-3.0.4.tar.gz [following]
--2023-02-06 20:24:15--  https://downloads.cisofy.com/lynis/lynis-3.0.4.tar.gz
Resolving downloads.cisofy.com (downloads.cisofy.com)... 2a01:7c8:aac2:37b::1, 37.97.194.171
Connecting to downloads.cisofy.com (downloads.cisofy.com)|2a01:7c8:aac2:37b::1|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 332004 (324K) [application/octet-stream]
Saving to: 'lynis-3.0.4.tar.gz'

lynis-3.0.4.tar.gz              100%[==========================================================================>] 324.22K   234KB/s    in 1.4s

2023-02-06 20:24:18 (234 KB/s) - 'lynis-3.0.4.tar.gz' saved [332004/332004]
```

Figure 10 Wget Lynis Download

After complete install and Lynis started. A local audit was run on the server. This command was used "./lynis audit system | tee lynis-audit-new.log" this wrote the results to a log file. By default, the log file is located in the current directory where the audit was run but the lynis application logs are located in /var/log/lynis.log. After the audit was completed the log file was viewed.

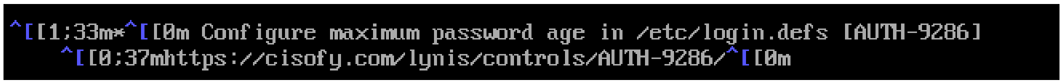


```
Lynis security scan details:

Hardening index : 65 [#############       ]
Tests performed : 251
Plugins enabled : 0

Components:
- Firewall               [V]
- Malware scanner        [X]

Scan mode:
Normal [V]  Forensics [ ]  Integration [ ]  Pentest [ ]

Lynis modules:
- Compliance status      [?]
- Security audit         [V]
- Vulnerability scan     [V]

Files:
- Test and debug information      : /var/log/lynis.log
- Report data                     : /var/log/lynis-report.dat

================================================================================

Lynis 3.0.4

Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2007-2021, CISOfy - https://cisofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)

================================================================================

[TIP]: Enhance Lynis audits by adding your settings to custom.prf (see /root/lynis/default.prf for all settings)
```

Figure 11 Lynis Results

As shown in figure 6. Lynis gave me a hardening index of 65 and gave a tip at the bottom.

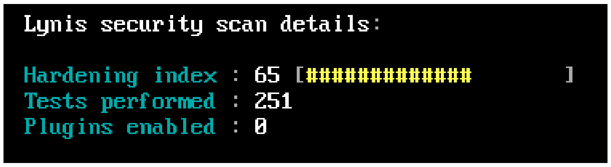
Some hardening suggestions were completed.



```
^[[1;33m*^[[0m Configure maximum password age in /etc/login.defs [AUTH-9286]
    ^[[0;37mhttps://cisofy.com/lynis/controls/AUTH-9286/^[[0m
```

Figure 12 Hardening

After adding these recommendations, the audit was run again.



```
Lynis security scan details:

Hardening index : 65 [#############       ]
Tests performed : 251
Plugins enabled : 0
```

Figure 13 Audit after Hardening

Sadly the hardening I did didn't adjust my index score but I imagine after changing many that this score would increase.

What is Lynis:

Lynis is an open-source security auditing tool used to perform security assessments on Unix-based systems such as Linux, macOS, and BSD. It is important because it helps system administrators and security professionals identify potential security issues and implement best practices for hardening their systems, thereby reducing the risk of successful attacks.

Part 3 - msfvenom payload attack

This was completed in Lab 2. The link for that document is below. See part 5.

https://docs.google.com/document/d/199-CrwRQl4ZfhNYtVOCsxi53jU2Yxb4l/edit?usp=sharing&ouid=106005233586037779518&rtpof=true&sd=true

Part 4 - rootkits and SELINUX

What are rootkits and what do they do?

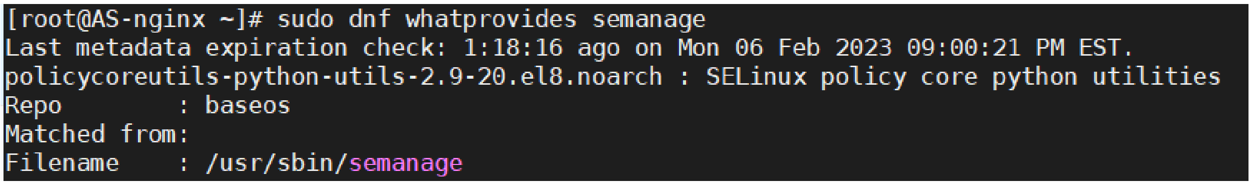
Rootkits are malicious software that hide their presence and modify system behavior to evade detection. They work by replacing or modifying the system components and hiding their files, processes, and registry keys. In order to detect rootkits, various tools such as rkhunter and AIDE can be used, but a deeper understanding of SELinux is considered one of the best ways to detect rootkits effectively.

Oracle Lab:

An Oracle lab was followed, major commands used in the lab will be documented below.

Determine which package provides the utility:

“sudo dnf whatprovides semanage”



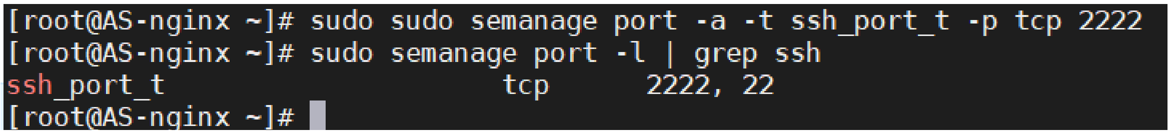
```
[root@AS-nginx ~]# sudo dnf whatprovides semanage
Last metadata expiration check: 1:18:16 ago on Mon 06 Feb 2023 09:00:21 PM EST.
policycoreutils-python-utils-2.9-20.el8.noarch : SELinux policy core python utilities
Repo        : baseos
Matched from:
Filename    : /usr/sbin/semanage
```

Figure 14 What Package Provides the Utilty

Add a new port to the SELinux policy:

“sudo sudo semanage port -a -t ssh_port_t -p tcp 2222”

Then check what ports it allows now.

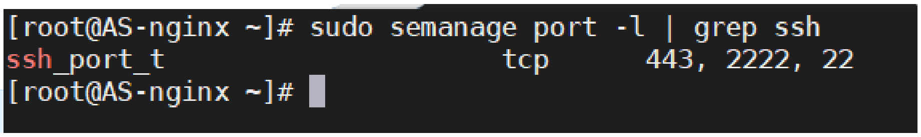


```
[root@AS-nginx ~]# sudo sudo semanage port -a -t ssh_port_t -p tcp 2222
[root@AS-nginx ~]# sudo semanage port -l | grep ssh
ssh_port_t                     tcp      2222, 22
[root@AS-nginx ~]# 
```

Figure 15 New Allowed Port

Modify a port in the SELinux policy:

“sudo semanage port -m -t ssh_port_t -p tcp 443”



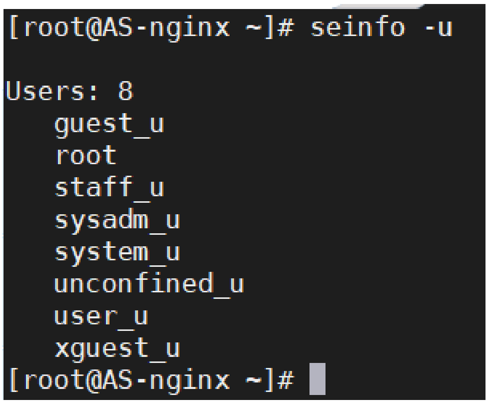
```
[root@AS-nginx ~]# sudo semanage port -l | grep ssh
ssh_port_t                     tcp      443, 2222, 22
[root@AS-nginx ~]#
```

Figure 16 Port 443 Added

This added port 443 as showing using the command showed in figure 16.

Get a listing of the SELinux users:

“seinfo -u”

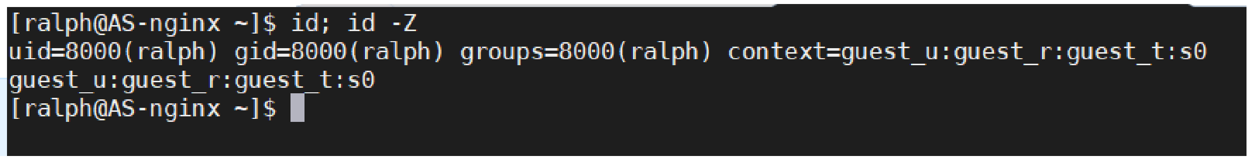


```
[root@AS-nginx ~]# seinfo -u

Users: 8
   guest_u
   root
   staff_u
   sysadm_u
   system_u
   unconfined_u
   user_u
   xguest_u
[root@AS-nginx ~]#
```

Figure 17 Seinfo -u

Check the users context:

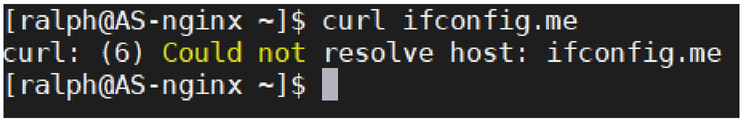
“id: id -Z”



```
[ralph@AS-nginx ~]$ id; id -Z
uid=8000(ralph) gid=8000(ralph) groups=8000(ralph) context=guest_u:guest_r:guest_t:s0
guest_u:guest_r:guest_t:s0
[ralph@AS-nginx ~]$
```

Figure 18 User Context

Ralph user not having access to using curl command:

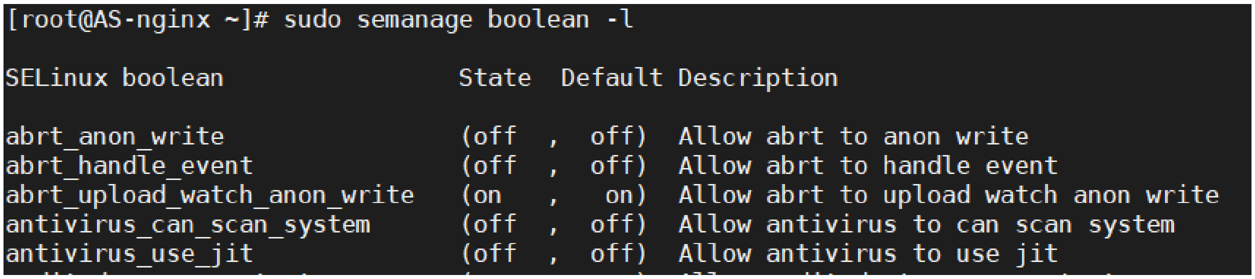


```
[ralph@AS-nginx ~]$ curl ifconfig.me
curl: (6) Could not resolve host: ifconfig.me
[ralph@AS-nginx ~]$
```

Figure 19 Ralph Blocked from Curl

Get a list of booleans along with their meaning:

“sudo semanage boolean -l”

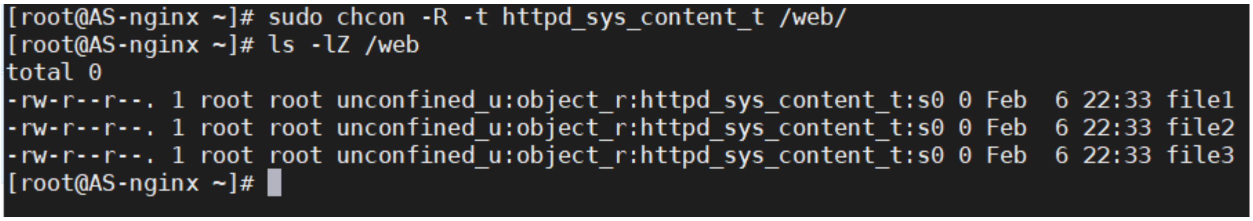


```
[root@AS-nginx ~]# sudo semanage boolean -l

SELinux boolean                State  Default Description

abrt_anon_write                (off  ,  off)  Allow abrt to anon write
abrt_handle_event              (off  ,  off)  Allow abrt to handle event
abrt_upload_watch_anon_write   (on   ,   on)  Allow abrt to upload watch anon write
antivirus_can_scan_system      (off  ,  off)  Allow antivirus to can scan system
antivirus_use_jit              (off  ,  off)  Allow antivirus to use jit
```

Figure 20 List of Booleans with Semanage

Make temporary label changes:

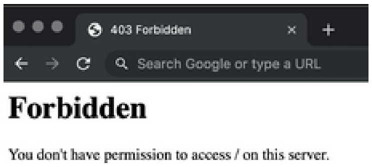
“sudo chcon -R -t httpd_sys_content_t /web/”



```
[root@AS-nginx ~]# sudo chcon -R -t httpd_sys_content_t /web/
[root@AS-nginx ~]# ls -lZ /web
total 0
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 0 Feb  6 22:33 file1
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 0 Feb  6 22:33 file2
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 0 Feb  6 22:33 file3
[root@AS-nginx ~]#
```

Figure 21 Temporary Label Changes

This command was then done to my site 1 located in /var/www/site1



Forbidden

You don't have permission to access / on this server.

Figure 22 Nginx Site Forbidden

This caused my site to give a 403 Forbidden error message.

Brief description of how they work for each:

"sudo dnf whatprovides semanage":

This command determines which package provides the SELinux utility called "semanage".

"sudo semanage port -a -t ssh_port_t -p tcp 2222":

This command adds a new port (2222) to the SELinux policy and assigns it the type "ssh_port_t".

"sudo semanage port -m -t ssh_port_t -p tcp 443":

This command modifies an existing port (443) in the SELinux policy and changes its type to "ssh_port_t".

"seinfo -u":

This command gets a listing of the SELinux users.

"id -Z":

This command checks the context of the current user.

"sudo semanage boolean -l":

This command gets a list of SELinux booleans and their meanings.

"sudo chcon -R -t httpd_sys_content_t /var/www/site1”:

This command makes temporary changes to the SELinux label of files and directories under "/var/www/site1 to "httpd_sys_content_t".

Part 5 - System hardening

Using CIS hardening guides on both a Windows and Linux Server VM. 10 recommendations will be found and applied to each of the aforementioned VMs.

Windows Server 2016:

1. Disable unnecessary services and protocols such as Telnet, FTP, and TFTP. This can be done through the services management console or the command line.
2. Remove unnecessary software, such as games and development tools. This can be done through the add/remove programs feature in Control Panel.
3. Enable logging and audit trails to track security-related events. This can be configured in the Event Viewer.
4. Limit user privileges and use the principle of least privilege. This can be done through User Account Control settings and Group Policy.
5. Apply patches and updates regularly to address known vulnerabilities. This can be done through Windows Update.
6. Use strong passwords and implement account lockout policies. This can be configured in the Local Security Policy.
7. Enable and configure firewall to restrict incoming and outgoing network traffic. This can be done through the Windows Firewall with Advanced Security.
8. Enable and configure antivirus and anti-malware software. This can be done through the Windows Defender Security Center.
9. Encrypt sensitive data, such as files and system images. This can be done through BitLocker Drive Encryption.
10. Implement network segmentation and access control to limit exposure of sensitive data. This can be done through Virtual LANs (VLANs) and access control lists (ACLs).

Rocky Linux 8:

1. Disable unnecessary services and protocols such as Telnet, FTP, and TFTP. This can be done through the system's init scripts or system control commands.
2. Remove unnecessary software, such as games and development tools. This can be done through the package manager, such as yum or dnf.
3. Enable logging and audit trails to track security-related events. This can be configured in the system's syslog or rsyslog facilities.
4. Limit user privileges and use the principle of least privilege. This can be done through user account settings and access control mechanisms, such as sudo.
5. Apply patches and updates regularly to address known vulnerabilities. This can be done through the package manager, such as yum or dnf.
6. Use strong passwords and implement account lockout policies. This can be configured in the system's PAM or shadow files.
7. Enable and configure firewall to restrict incoming and outgoing network traffic. This can be done through the iptables firewall.
8. Enable and configure antivirus and anti-malware software. This can be done through the ClamAV antivirus software.
9. Encrypt sensitive data, such as files and system images. This can be done through dm-crypt or LUKS encryption.
10. Implement network segmentation and access control to limit exposure of sensitive data. This can be done through virtual private networks (VPNs) and access control lists (ACLs).

Masking a Service: Masking a service can be a security measure as it reduces the attack surface of the system by disabling unnecessary services. This is also helpful when you know a service is vulnerable but a patch doesn’t exist yet and you need it running. In this case, you can mask it and wait for it to be patched. Then patch the software and unmask it.

NCP Site for nist.gov

Are there any for Cisco IOS? What about on CIS?

Yes, there are security hardening guidelines for Cisco IOS available on NIST.gov and the Center for Internet Security (CIS).

5 things that we do to follow the guidelines for securing Cisco routers and switches:

1. Configure secure management protocols such as SSH and disable Telnet.
2. Enable access control lists (ACLs) to restrict network access.
3. Implement traffic filtering and limit exposure of sensitive information.
4. Enable logging and monitoring to detect and respond to security events.
5. Enable password encryption services.

Defense Techniques

This lab didn't have many attacks so the only one that will be covered is the msfvenom payload attack.

Here is a list of ways you can protect your computer or devices from a msfvenom attack.

- Keep your operating system and software up to date with the latest security patches.
- Use antivirus software and keep it updated. (Defender seemed to do a good job with this)
- Enable a firewall to block unauthorized access to your computer. (Not always going to work but can mitigate attacks)
- Be cautious when opening email attachments or clicking on links, especially if they come from unknown sources.
- Avoid downloading and installing software from untrusted websites.
- Use strong passwords and enable two-factor authentication where possible.
- Backup important data regularly and store it securely.
- Regularly educate yourself about common attack methods and stay informed about the latest threats.

References

[1] A. Linuxshelltips, "Verify SHA256 Checksum of File in Linux," Linuxshelltips, [Online]. Available: https://www.linuxshelltips.com/verify-sha256-checksum-of-file-in-linux/. [Accessed: 07-Feb-2023].

[2] Cisofy, "Lynis," [Online]. Available: https://cisofy.com/lynis/. [Accessed: 07-Feb-2023].

[3] Red Hat, Inc., "SELinux Troubleshooting Tips and Techniques," [Online]. Available: https://access.redhat.com/articles/2639581. [Accessed: 07-Feb-2023].

[4] The SELinux Project, "Category: Notebook," [Online]. Available: https://selinuxproject.org/page/Category:Notebook. [Accessed: 07-Feb-2023].

[5] Ask Ubuntu, "Protect Ubuntu System from MSFvenom Attacks," [Online]. Available: https://askubuntu.com/questions/715343/protect-ubuntu-system-from-msfvenom-attacks. [Accessed: 07-Feb-2023].

[6] National Institute of Standards and Technology, "National Checklist Program," [Online]. Available: https://ncp.nist.gov/. [Accessed: 07-Feb-2023].

[7] T. Hernandez, "Masked Services in Linux: How to Manage," TechRepublic, January 8, 2021, [Online]. Available: https://www.techrepublic.com/article/masked-services-linux-how-manage/. [Accessed: February 7, 2023].