

Nessus Vulnerability Management Lab

By Michael Ambeguia

Purpose:

To provide myself with a beginner friendly, structured, and hands-on learning experience applying the vulnerability management process using Nessus Essentials. I would also like to practice effectively identifying, analyzing, and remediating security vulnerabilities within Linux-based environments.

Skills Applied:

1. Applying the vulnerability management process
2. Using the Nessus Essentials vulnerability scanner
3. Configuring/ Maintaining vulnerability Scanners
4. Interpreting scan results
5. Remediating vulnerabilities
6. Reporting Findings

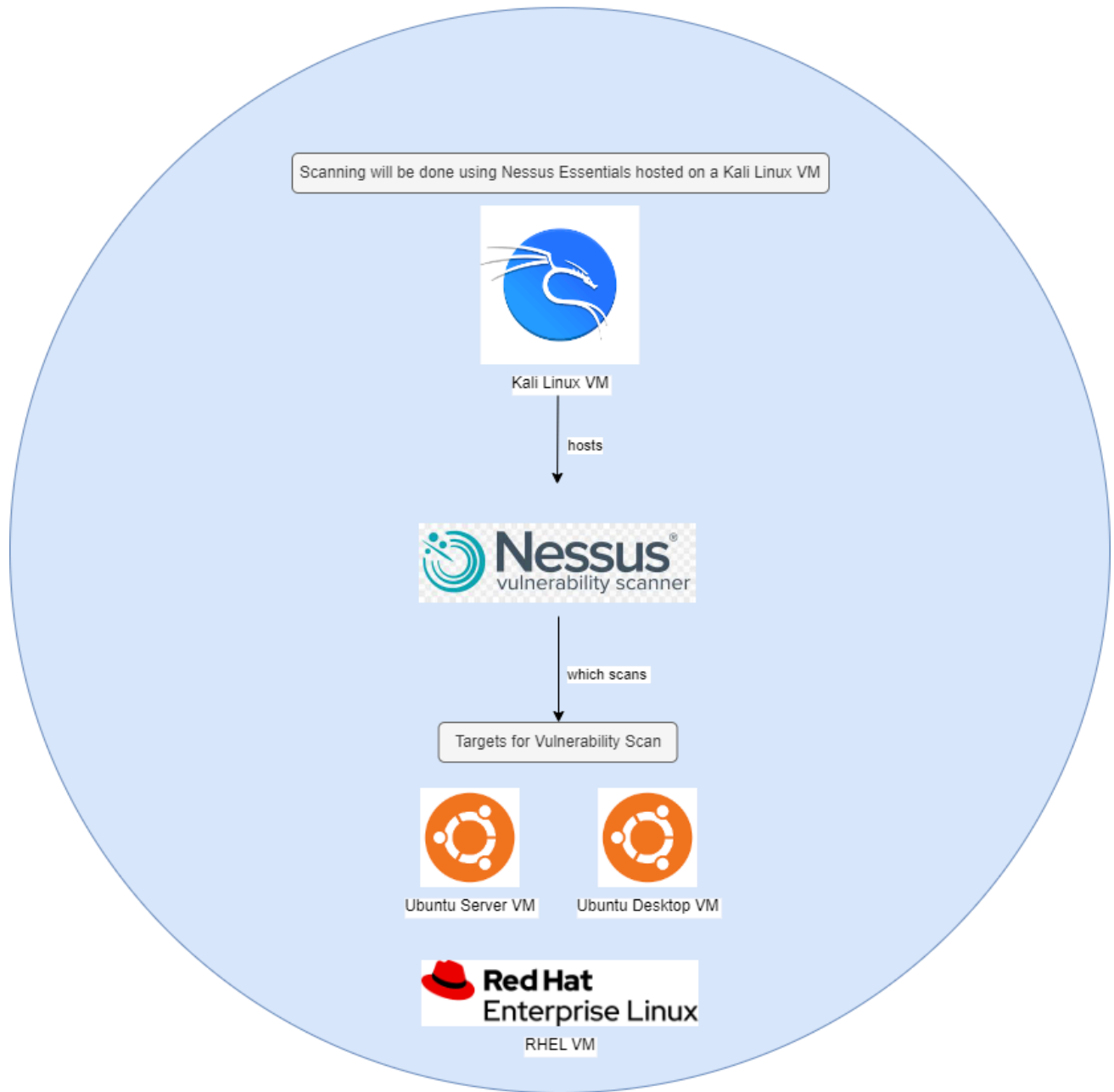
Sections:

1. Lab Introduction
2. Nessus Installation
3. Configuration and Initialization
4. Basic Scanning
5. Advanced Scanning
6. Interpreting Scan Results
7. Remediation Strategies
8. Reporting and Documentation

Section #1 Lab Introduction:

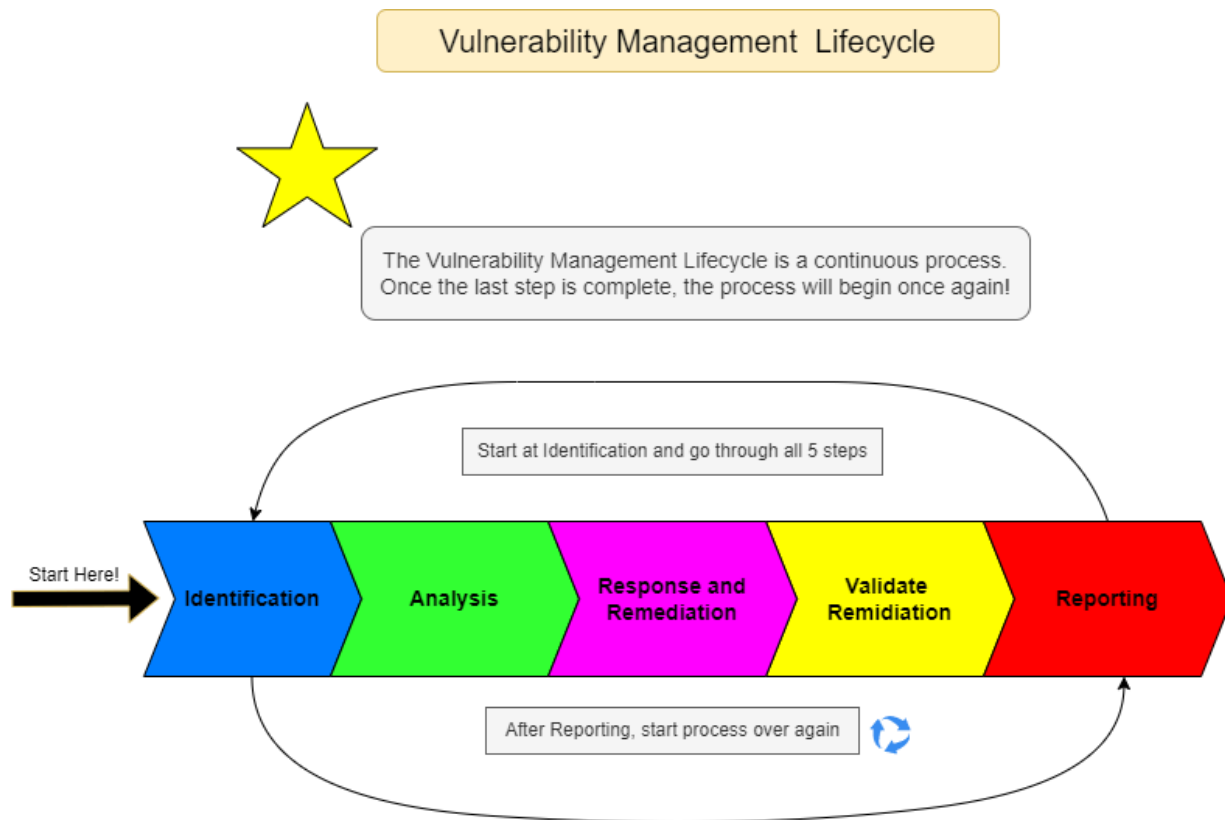
1.1. Lab Topology:

Nessus Vulnerability Management Lab Toplogy



All devices for this lab are on the same subnet, hence the circle containing the topology of the lab. The devices are on the 192.168.1.0/24 subnet.

1.2 Vulnerability Management Lifecycle:



The vulnerability management lifecycle has 5 steps.

Step 1. Identification: Identification is when you use vulnerability scanners to scan devices and applications for vulnerabilities. At this step your goal is to use the tools at your disposal to find **possible** vulnerabilities. Not all vulnerabilities are valid. Some might be false positives which means that the scanner thinks it found one when it is not present. Some might be false negatives which means that the scanner thinks a vulnerability is not present when it is. That is why step #3 is so important.

Step 2. Analysis: Analysis is the step after the scanning occurs. During this step your goal is to analyze the results of the scan. At this step you will analyze the vulnerabilities found and prioritize which ones need to be remediated. This is done based off of the CVSS scores and the ease of the remediation.

Step 3. Response and Remediation: During this step you will use a remediation strategy to resolve the vulnerabilities. The remediation strategy I will be using is to first verify that the vulnerability is present, patch the vulnerability, then rescan the device to verify that the vulnerability has been remediated. If the vulnerability is still present you will have to try patching it again.

Step #4: Validate Remediation: This step is when the rescanning to validate the vulnerability is gone. In this lab I included it as part of step #3 since it is closely tied to step 3.

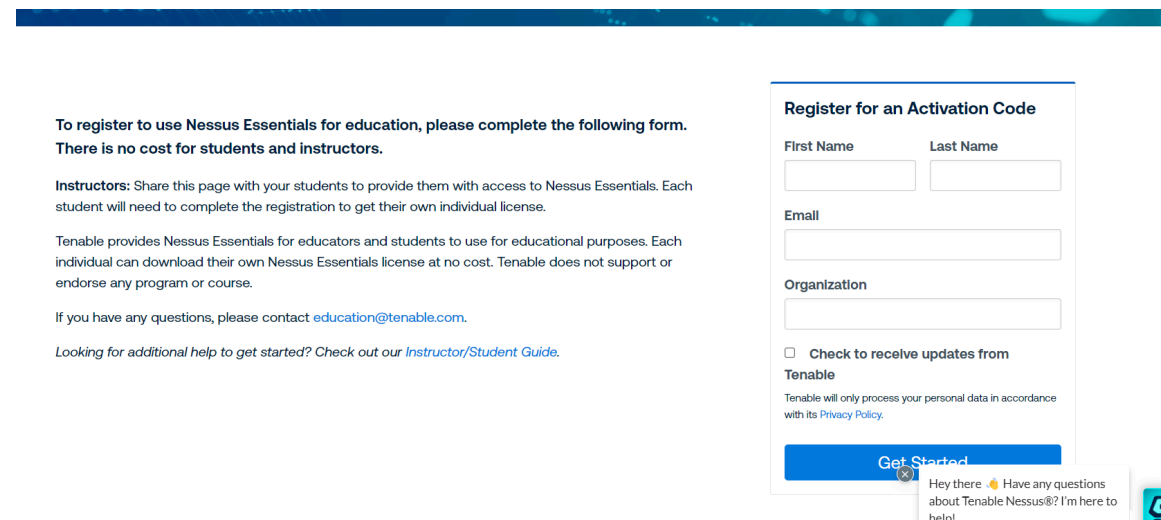
Step #5: Reporting: The last step is to report your findings. This step also serves as a point of reflection. You can go over the steps you took during the scan and think about what went wrong and what you might do next time to improve the process.

Section #2 Nessus Installation:

In this lab I will be using Nessus Essentials, the free version of Nessus Professional, to perform the vulnerability scanning.

2.1 Sign up for a Nessus For Education account on the Nessus website:

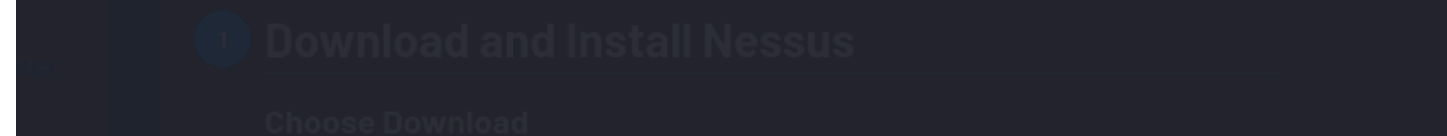
<https://www.tenable.com/tenable-for-education/nessus-essentials?edu=true> so that you can get a license for Nessus Essentials.



The screenshot shows the registration page for Nessus Essentials for education. On the left, there is instructional text: "To register to use Nessus Essentials for education, please complete the following form. There is no cost for students and instructors." It also includes instructions for instructors and a link to contact education@tenable.com. On the right, there is a registration form titled "Register for an Activation Code". The form has fields for "First Name", "Last Name", "Email", and "Organization". Below these fields is a checkbox labeled "Check to receive updates from Tenable". At the bottom of the form is a blue "Get Started" button. A small chat bubble from Tenable is visible in the bottom right corner of the form area.

2.2 Download Nessus

```
(kali㉿kali)-[~]
$ curl --request GET \nable Nessus
--url 'https://www.tenable.com/downloads/api/v2/pages/nessus/files/Nessus-10.8.2-debian10_amd64.deb' \
--output 'Nessus-10.8.2-debian10_amd64.deb'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 5403k    0 5403k    0     0  861k      0 --:--:--  0:00:06 --:--:-- 965k
```



The screenshot shows the Nessus download page. It features a large blue circular icon with a white 'D' and the text "Download and Install Nessus". Below this, there is a section titled "Choose Download" with a list of download links for different operating systems and architectures.

I used dpkg -i to install nessus. Dpkg -i is different from apt install since apt install is used to install packages from a remote repository over the internet while dpkg -i is used to download packages you download manually off of the internet onto your Debian based Linux device.

```
(kali@kali)-[~/Downloads]
$ ls -l
total 66944
-rw-rw-r-- 1 kali kali 68547492 Sep  2 03:47 Nessus-10.8.2-debian10_amd64.deb

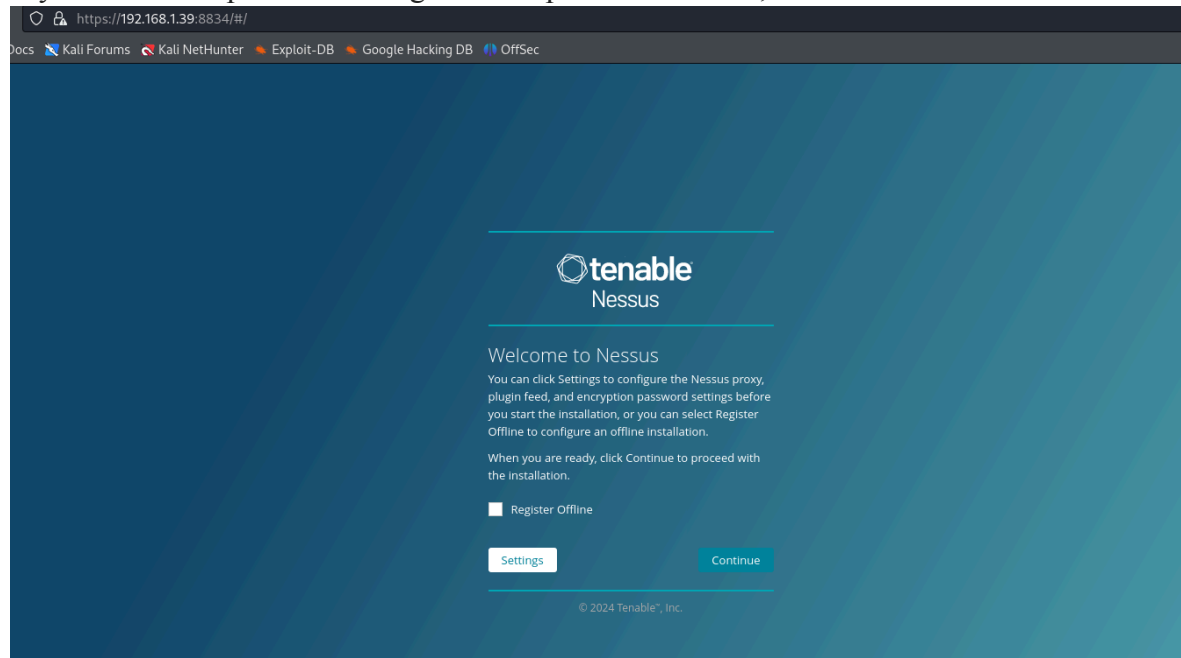
(kali@kali)-[~/Downloads]
$ sudo dpkg -i Nessus-10.8.2-debian10_amd64.deb
Selecting previously unselected package nessus.
(Reading database ... 403519 files and directories currently installed.)
Preparing to unpack Nessus-10.8.2-debian10_amd64.deb ...
Unpacking nessus (10.8.2) ...
Setting up nessus (10.8.2) ...
HMAC : (Module_Integrity) : Pass
SHA1 : (KAT_Digest) : Pass
SHA2 : (KAT_Digest) : Pass
SHA3 : (KAT_Digest) : Pass
TDES : (KAT_Cipher) : Pass
AES_GCM : (KAT_Cipher) : Pass
AES_ECB_Decrypt : (KAT_Cipher) : Pass
RSA : (KAT_Signature) : RNG : (Continuous_RNG_Test) : Pass
Pass
ECDSA : (PCT_Signature) : Pass
ECDSA : (PCT_Signature) : Pass
DSA : (PCT_Signature) : Pass
TLS13_KDF_EXTRACT : (KAT_KDF) : Pass
TLS13_KDF_EXPAND : (KAT_KDF) : Pass
TLS12_PRF : (KAT_KDF) : Pass
PBKDF2 : (KAT_KDF) : Pass
SSHKDF : (KAT_KDF) : Pass
KDKDF : (KAT_KDF) : Pass
HKDF : (KAT_KDF) : Pass
SSKDF : (KAT_KDF) : Pass
X963KDF : (KAT_KDF) : Pass
X942KDF : (KAT_KDF) : Pass
HASH : (DRBG) : Pass
CTR : (DRBG) : Pass
HMAC : (DRBG) : Pass
DH : (KAT_KA) : Pass
ECDH : (KAT_KA) : Pass
RSA_Encrypt : (KAT_AsymmetricCipher) : Pass
RSA_Decrypt : (KAT_AsymmetricCipher) : Pass
```

Once it is installed,, you need to start the nessusd service so that Nessus is up and running on your system.

```
(kali@kali)-[~/Downloads]
$ sudo systemctl start nessusd

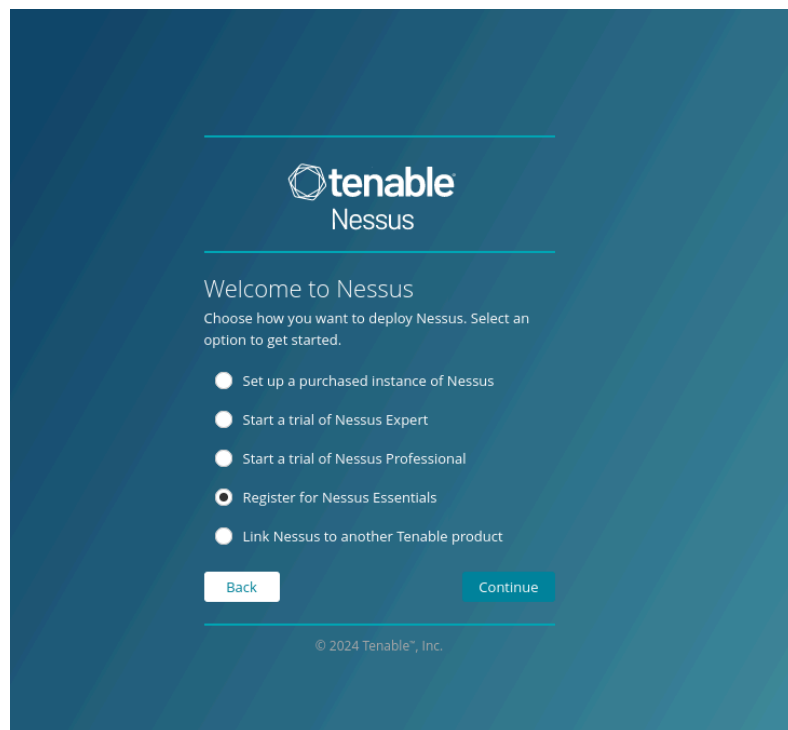
(kali@kali)-[~/Downloads]
$
```

Nessus can then be accessed through your browser on the device you installed it on. Simply type in your device's ip address along with the port Nessus uses, 8834.




Section #3 Configuring and Initializing Nessus

3.1 Choose register for Nessus Essentials:



3.2. Skip logging in if you already have a license key.



Nessus

Login

Login with your Tenable Community credentials to view your available products.

New Users: Please select the 'Create or reset your password' link to complete account setup.

Email

Password

[Create or reset your password](#)

Already have activation code? Skip this step to enter it manually.

BackSkipContinue

© 2024 Tenable™, Inc.

After skipping, enter in the license key, then create your login info for the Nessus Essentials scanner.



Nessus

Register Nessus

Enter your activation code.

Activation Code

BackContinue

© 2024 Tenable™, Inc.



Nessus

Create a user account

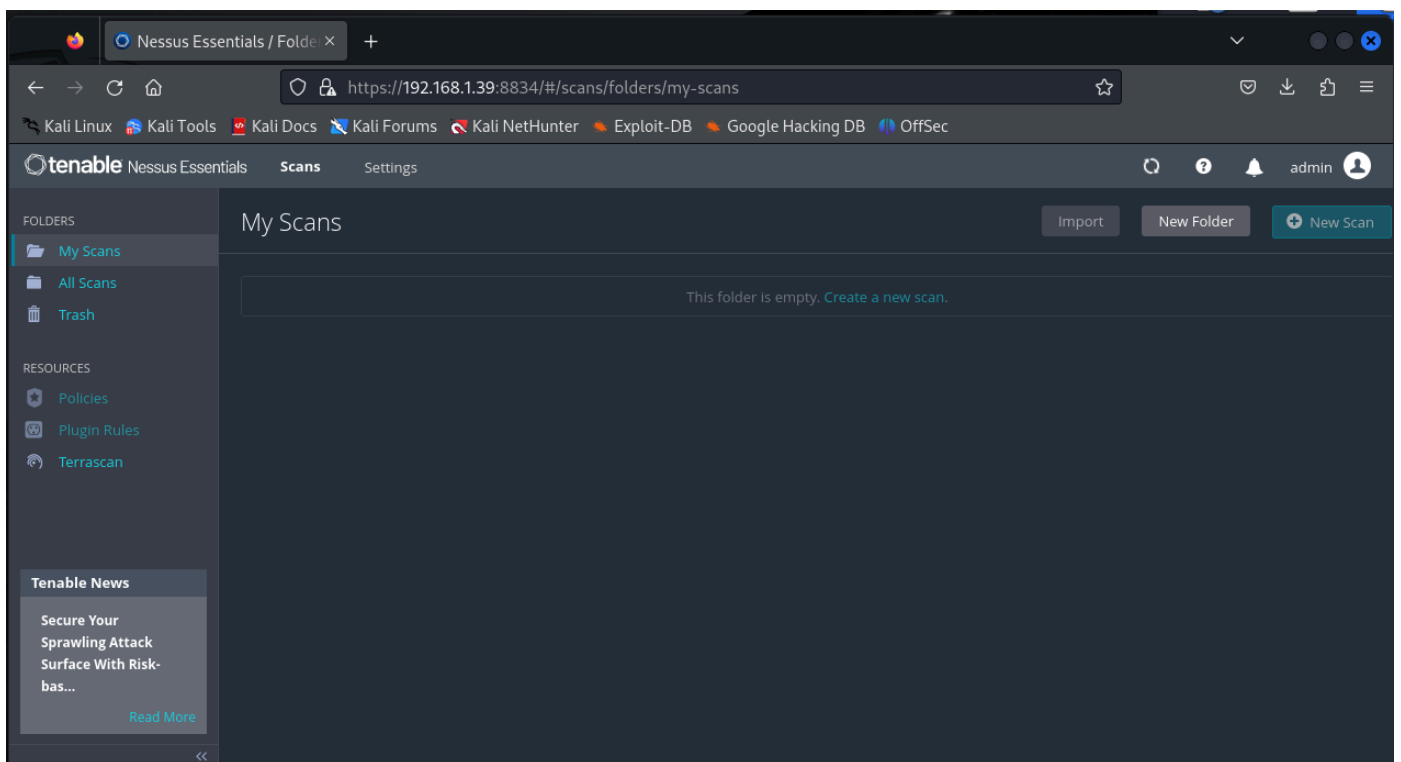
Create a Nessus administrator user account. Use this username and password to log in to Nessus.

Username *

Password *

BackSubmit

© 2024 Tenable™, Inc.



Section #4 Basic Scanning:

A. The first type of scan that I will perform is basic, non credentialed scans. Non-credentialed scans are non-invasive and simulate what an external attacker will see if they were trying to find vulnerabilities on the network devices. Unlike credentialed scans, these scans do not get a deep view of the system configuration of the devices and merely identify open ports and services. Since these scans only look at the devices from the outside they might not find all vulnerabilities, and if vulnerabilities are found they might be false-positives.

B. Scanning Ubuntu Desktop VM #1 (IPv4 Address 192.168.1.36):

For this first scan I will demonstrate the steps in detail, subsequent scans will omit the details since I will use the same parameters for them except for the name and the ip address.

1. Create the scan name and specify the target ip address:

The screenshot shows the Nessus 'Settings' window with the 'Credentials' tab selected. On the left, a sidebar lists categories: BASIC (with sub-items General, Schedule, and Notifications), DISCOVERY, ASSESSMENT, REPORT, and ADVANCED. The main area is for configuring a scan. The 'Name' field contains 'Ubuntu VM #1 Scan 1'. The 'Description' field is empty. The 'Folder' dropdown menu is set to 'Lab Scans'. The 'Targets' text area contains the IP address '192.168.1.36'. At the bottom, there is an 'Upload Targets' section with an 'Add File' link. At the very bottom of the window are 'Save' and 'Cancel' buttons.

2. Keep all the defaults for the other parameters:

Settings

Credentials

Plugins

BASIC

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Scan Type

Port scan (common ports)

General Settings:

Always test the local Nessus host

Use fast network discovery

Port Scanner Settings:

Scan common ports

Use netstat if credentials are provided

Use SYN scanner if necessary

Ping hosts using:

TCP

ARP

ICMP (2 retries)

Save

Cancel

New Scan / Basic Network Scan

[Back to Scan Templates](#)

Settings

Credentials

Plugins

BASIC

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Scan Type

Default

General Settings:

Avoid potential false alarms

Disable CGI scanning

Web Applications:

Disable web application scanning

Save

Cancel

New Scan / Basic Network Scan

[Back to Scan Templates](#)

Settings

Credentials

Plugins

BASIC

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Processing

☐ Override normal verbosity

☒ I have limited disk space. Report as little information as possible

☐ Report as much information as possible

☒ Show missing patches that have been superseded

☒ Hide results from plugins initiated as a dependency

Output

☒ Allow users to edit scan results

☐ Designate hosts by their DNS name

☐ Display hosts that respond to ping

☐ Display unreachable hosts

3. I won't include credentials this time around:

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

Settings

Credentials

Plugins

CATEGORIES

Host

▼

Filter Credentials

Q

SSH

∞

Windows

∞

Save

▼

Cancel

4. I also don't have to do anything for the plugins:

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

| Settings | Credentials | Plugins |
|------------------------------------|-------------|---------|
| PLUGIN FAMILY ▲ | | TOTAL |
| AIX Local Security Checks | | 11563 |
| Alma Linux Local Security Checks | | 1433 |
| Amazon Linux Local Security Checks | | 4791 |
| Artificial Intelligence | | 31 |
| Backdoors | | 123 |
| Brute force attacks | | 26 |
| CentOS Local Security Checks | | 4947 |
| CGI abuses | | 6123 |
| CGI abuses : XSS | | 707 |
| CISCO | | 2437 |
| Databases | | 1011 |
| Debian Local Security Checks | | 9480 |
| Default Unix Accounts | | 172 |
| Denial of Service | | 110 |

5. Perform the scan:

To perform the scan you have to go to click on the scan you created, then click launch:

Configure

Launch


Scan Details

Status: Empty

Scanner: Local Scanner


Now the scan is running!

Configure

| Status |
|---|
|  Running |

Scan Details

Policy: Basic Network Scan

Status: Running 

Severity Base: CVSS v3.0

Scanner: Local Scanner

Start: Today at 2:12 PM

6. Look at the results:

I won't analyze the results in detail now, but here are the results of the uncredentialed basic scan:

Hosts1

Vulnerabilities20

History1

Filter

Search Vulnerabilities

20 Vulnerabilities

| <input type="checkbox"/> | Sev | CVSS | VPR | EPSS | Name | Family | Count | | |
|--------------------------|------|--------------------------|-----|--------|---|-------------------|-------|--|--|
| <input type="checkbox"/> | LOW | 2.1 * | 4.2 | 0.8808 | ICMP Timestamp Request Remote Date Disclosure | General | 1 | | |
| <input type="checkbox"/> | INFO | ... | ... | ... | 2 RPC (Multiple Issues) | RPC | 2 | | |
| <input type="checkbox"/> | INFO | ... | ... | ... | 2 SSH (Multiple Issues) | General | 2 | | |
| <input type="checkbox"/> | INFO | ... | ... | ... | 2 SSH (Multiple Issues) | Misc. | 2 | | |
| <input type="checkbox"/> | INFO | ... | ... | ... | 2 SSH (Multiple Issues) | Service detection | 2 | | |
| <input type="checkbox"/> | INFO | Nessus SYN scanner | | | | Port scanners | 2 | | |
| <input type="checkbox"/> | INFO | RPC Services Enumeration | | | | Service detection | 2 | | |

C. Scanning Ubuntu Server VM:

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

Settings

Credentials

Plugins

BASIC

General

Schedule

Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name

Ubuntu Server Scan 1

Description

Folder

Lab Scans

Targets

192.168.1.22

Upload Targets

Add File

Save

Cancel

| Ubuntu Server Scan 1 | | | | | | |
|--|-------|--------|-------|--------|---|-------------------|
| ← Back to Lab Scans | | | | | | |
| <div> <div>Hosts 1</div> <div>Vulnerabilities 17</div> <div>History 1</div> </div> | | | | | | |
| <div> <div>Filter</div> <div>Search Vulnerabilities</div> <div>17 Vulnerabilities</div> </div> | | | | | | |
| <input type="checkbox"/> | Sev ▼ | CVSS ▼ | VPR ▼ | EPSS ▼ | Name ▲ | Family ▲ |
| <input type="checkbox"/> | LOW | 2.1 * | 4.2 | 0.8808 | ICMP Timestamp Request Remote Date Disclosure | General |
| <input type="checkbox"/> | INFO | ... | ... | ... | SSH (Multiple Issues) | General |
| <input type="checkbox"/> | INFO | ... | ... | ... | SSH (Multiple Issues) | Misc. |
| <input type="checkbox"/> | INFO | ... | ... | ... | SSH (Multiple Issues) | Service detection |
| <input type="checkbox"/> | INFO | | | | Common Platform Enumeration (CPE) | General |
| <input type="checkbox"/> | INFO | | | | Device Type | General |
| <input type="checkbox"/> | INFO | | | | Ethernet Card Manufacturer Detection | Misc. |
| <input type="checkbox"/> | INFO | | | | Ethernet MAC Addresses | General |

Section #5 Advanced Scanning:

A. The second type of scan I will perform is the credentialed scans. Like I mentioned before these scans can perform a deeper scan of the devices and identify system configuration vulnerabilities unlike non credentialed scans. It is good practice to perform both non-credentialed scans and credentialed scans since they can complement each other. The credentialed scans can find vulnerabilities that the non credentialed scans missed, assuring that all vulnerabilities are identified on the devices. Credentialed scans are also great to utilize in an environment where network security protections like firewalls and intrusion prevention systems are prevalent. Firewalls and IPSs will interfere with regular external, non credentialed scans. With credentialed scans the vulnerability scanner can subvert the network defenses and perform the scan unimpeded.

B. Scanning Ubuntu Desktop VM #1 (IPv4 Address 192.168.1.36):

1. I will now create the credentialed scan for the Ubuntu Desktop VM #1. I will use the same parameters as the non credentialed scan, except this time I will provide the credentials for SSH login so the scanner goes deep into the VM.

Settings

Credentials

Plugins

BASIC

General

Schedule

Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name

Ubuntu VM #1 Deep Scan 1

Description

Folder

Lab Scans

Targets

192.168.1.36

Upload Targets

Add File

Save

Cancel

CATEGORIES

Host

Filter Credentials

SSH

Windows

SSH

User: spy2, Auth method: password

Authentication method

password

Username

spy2

Password (unsafe!)

This password could be compromised if Nessus connects to

Elevate privileges with

su

su login

root

Account to escalate to

Escalation password

Location of su (directory)

/etc/sudoers

Custom password prompt

password:

Some devices are configured to prompt for a password with standard password prompts.

Targets to prioritize credentials

192.168.1.36

Any hostnames or IPs or CIDR blocks (in a comma or space separated list)

Location of su (directory)

/etc/sudoers

Custom password prompt

password:

Some devices are configured to prompt for a password with a non-standard string such as 'secret-p' standard password prompts.

Targets to prioritize credentials

192.168.1.36

Any hostnames or IPs or CIDR blocks (in a comma or space separated list in this field) that match a credentials

Global Credential Settings

known_hosts file

Add File

If an SSH known_hosts file is available and provided as part of the Global Credential Settings of the this file. This can ensure that the same username and password you are using to audit your known control.

Preferred port

2500

This option can be set to direct Nessus to connect to SSH if it is running on a port other than 22.

Client version

OpenSSH_5.0

Specifies which type of SSH client Nessus will impersonate while scanning.

Attempt least privilege

☐

Enables or disables dynamic privilege escalation. When enabled, Nessus attempts to run the scan v enabled. If a command fails, Nessus will escalate privileges. Plugins 102095 and 102094 report whi increase scan run time by up to 30%.

2. Look at results of the credentialed scan:

Hosts1

Vulnerabilities47

Remediations2

History1

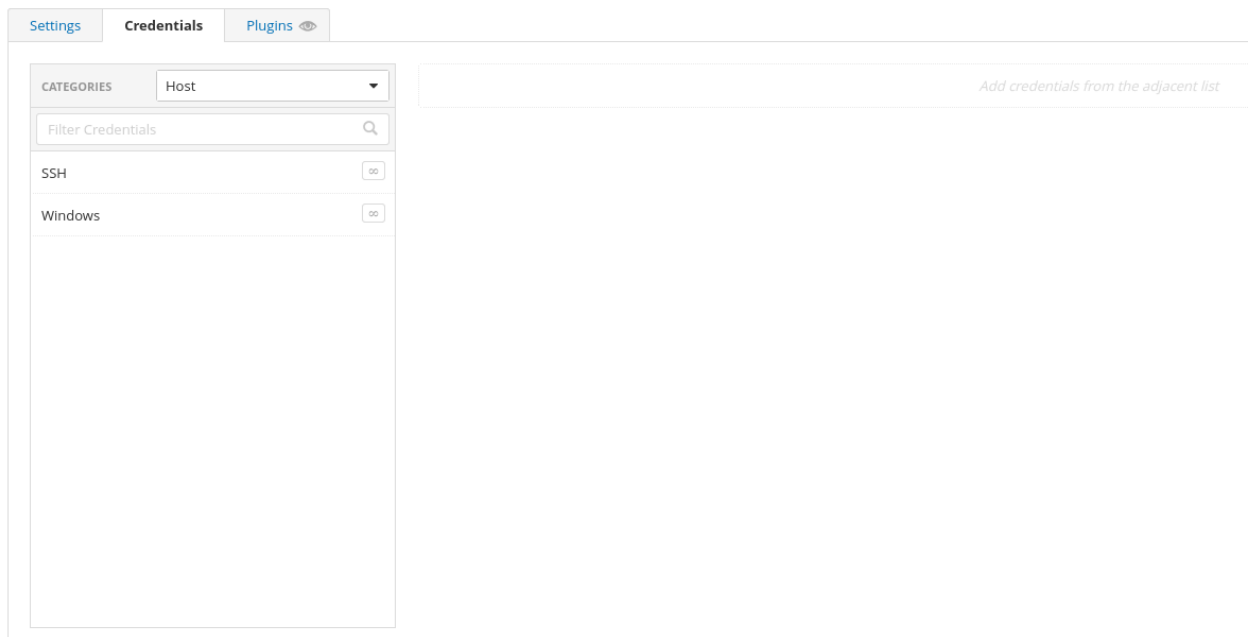
Filter

Search Vulnerabilities

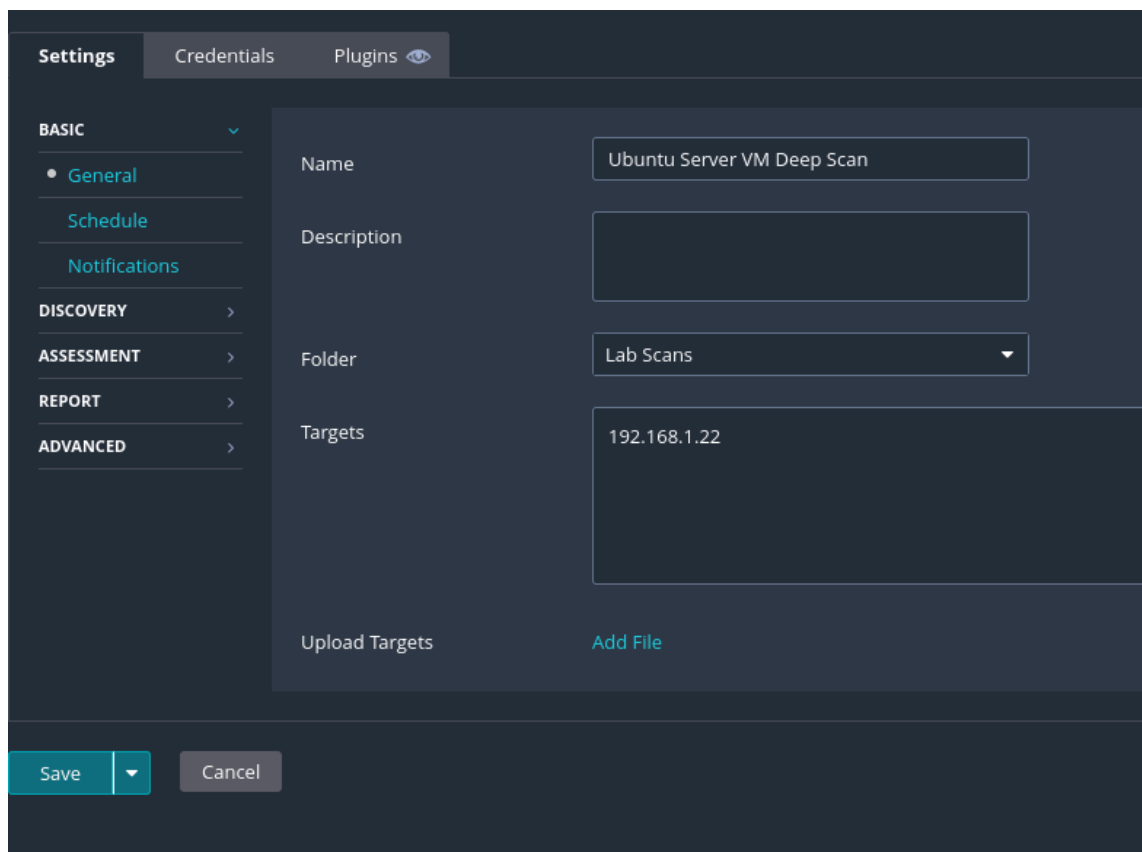
47 Vulnerabilities

| <input type="checkbox"/> | Sev | CVSS | VPR | EPSS | Name | Family |
|--------------------------|-------|-------|-----|--------|--|------------------------------|
| <input type="checkbox"/> | HIGH | 7.8 | 5.9 | | GNOME Shell <= 45.7 Code Execution in Portal Helper (CVE-2024-36472) | Misc. |
| <input type="checkbox"/> | LOW | 2.1 * | 4.2 | 0.8808 | ICMP Timestamp Request Remote Date Disclosure | General |
| <input type="checkbox"/> | MIXED | ... | ... | ... | 2 Canonical Ubuntu Linux (Multiple Issues) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | INFO | ... | ... | ... | 9 SSH (Multiple Issues) | General |
| <input type="checkbox"/> | INFO | ... | ... | ... | 4 SSH (Multiple Issues) | Misc. |
| <input type="checkbox"/> | INFO | ... | ... | ... | 2 RPC (Multiple Issues) | RPC |
| <input type="checkbox"/> | INFO | ... | ... | ... | 2 SSH (Multiple Issues) | Service detection |

3. The last thing that you should do after the scan is delete the credentials for Nessus:



C. Scanning Ubuntu Server VM #2 (IPv4 Address):



Settings

Credentials

Plugins

CATEGORIES

Host

Filter Credentials

SSH

Windows

SSH

Authentication method

password

Username

spy3

Password (unsafe!)

This password could be compromised if Nessus connects to the target

Elevate privileges with

su

su login

root

Account to escalate to

Escalation password

Location of su (directory)

/usr/bin

Custom password prompt

password:

Some devices are configured to prompt for a password with standard password prompts.

Targets to prioritize credentials

192.168.1.22

Any hostnames or IPs or CIDR blocks (in a comma or space separated list)

Ubuntu Server VM Deep Scan / Canonical Ubuntu Linux (Multiple Issues)

[Back to Vulnerabilities](#)

Hosts 1

Vulnerabilities 41

Remediations 21

History 1

Search Vulnerabilities

22 Vulnerabilities

| <input type="checkbox"/> | Sev ▼ | CVSS ▼ | VPR ▼ | EPSS ▼ | Name ▲ | Family ▲ |
|--------------------------|----------|--------|-------|--------|--|------------------------------|
| <input type="checkbox"/> | CRITICAL | 9.8 | 7.1 | 0.0384 | Ubuntu 14.04 LTS / 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 23.10 : Python vulnerabilities (...) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | CRITICAL | 9.8 | 6.7 | 0.0016 | Ubuntu 20.04 ESM / 22.04 ESM : SciPy vulnerabilities (USN-6226-1) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | CRITICAL | 9.8 | 5.9 | 0.0021 | Ubuntu 20.04 LTS / 22.04 LTS / 24.04 LTS : BusyBox vulnerabilities (USN-6961-1) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | CRITICAL | 9.1 | 6.0 | 0.0009 | Ubuntu 14.04 LTS / 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 24.04 LTS : Kerberos vulnerabil... | Ubuntu Local Security Checks |
| <input type="checkbox"/> | CRITICAL | 9.1 | 6.0 | 0.0008 | Ubuntu 20.04 LTS / 22.04 LTS / 23.10 / 24.04 LTS : Wget vulnerability (USN-6852-1) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | CRITICAL | 9.1 | 6.0 | 0.0004 | Ubuntu 20.04 LTS / 22.04 LTS / 24.04 LTS : OpenSSL vulnerabilities (USN-6937-1) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | HIGH | 8.2 | 8.1 | 0.0005 | Ubuntu 20.04 LTS / 22.04 LTS / 24.04 LTS : snapd vulnerabilities (USN-6940-1) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | HIGH | 8.1 | 9.5 | 0.7147 | Ubuntu 22.04 LTS / 23.10 / 24.04 LTS : OpenSSH vulnerability (USN-6859-1) | Ubuntu Local Security Checks |
| <input type="checkbox"/> | HIGH | 7.8 | 8.1 | 0.0004 | Ubuntu 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 24.04 LTS : Intel Microcode vulnerabilities (...) | Ubuntu Local Security Checks |

D. Scanning RHEL server VM (IPv4 Address 192.168.1.28):

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

Settings

Credentials

Plugins 

BASIC

• General

[Schedule](#)

[Notifications](#)

DISCOVERY >

ASSESSMENT >

REPORT >

ADVANCED >

Name

RHEL VM Deep Scan #1

Description

Folder

Lab Scans

Targets

192.168.1.28

Upload Targets

[Add File](#)

Save

Cancel

SSH

Authentication method

password

Username

administrator

Password (unsafe!)

●●●●●●●●●●

This password could be compromised if Nessus connects to a rogue SSH server.

Elevate privileges with

su

su login

root

Account to escalate to

Escalation password

●●●●●●●●●●

Location of su (directory)

/usr/bin

Custom password prompt

password:

Some devices are configured to prompt for a password with a non-standard password prompts.

Targets to prioritize credentials

192.168.1.28

Any hostnames or IPs or CIDR blocks (in a comma or space separated list) to prioritize credentials.

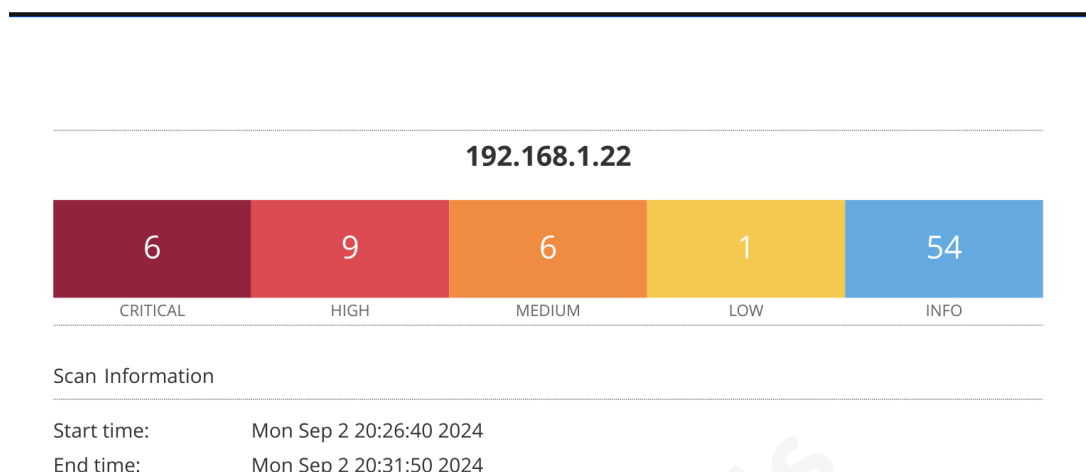
| RHEL VM Deep Scan #1 | | | | | | |
|-------------------------------------|------------------------|-----------------|--------------------|---|-------------------|--|
| ← Back to Lab Scans | | | | | | |
| Hosts | 1 | Vulnerabilities | 16 | History | 1 | |
| Filter | Search Vulnerabilities | | 16 Vulnerabilities | | | |
| Sev | CVSS | VPR | EPSS | Name | Family | |
| <input type="checkbox"/> MIXED | ... | ... | ... | HTTP (Multiple Issues) | Web Servers | |
| <input type="checkbox"/> LOW | 2.1 * | 4.2 | 0.8808 | ICMP Timestamp Request Remote Date Disclosure | General | |
| <input type="checkbox"/> INFO | ... | ... | ... | RPC (Multiple Issues) | RPC | |
| <input type="checkbox"/> INFO | | | | RPC Services Enumeration | Service detection | |
| <input type="checkbox"/> INFO | | | | Nessus SYN scanner | Port scanners | |
| <input type="checkbox"/> INFO | | | | Apache HTTP Server Version | Web Servers | |
| <input type="checkbox"/> INFO | | | | Backported Security Patch Detection (WWW) | General | |

Section #6 Interpreting Scan Results:

For the purposes of this lab I will only be interpreting the results based on the credentialed scans because these scans provide much more information than the non credentialed scans. I will only interpret the results for the Ubuntu Server VM, and focus on critical vulnerabilities only to save time as well. Also the scans used the CVSS 3.0 scoring system even though it has been replaced with CVSS 4.0.

A. Interpret the Scan Report for Ubuntu Server VM :

1. Create a PDF report of the scan:



I created a PDF report for the Ubuntu Server VM scan. The report contains details for all the vulnerabilities found such as the cvss score, how the vulnerability works, and some remediation steps to fix it. Nessus is helpful since the remediation information is highly detailed and straightforward to implement.

2. Interpret the results for Ubuntu Server VM:

The Ubuntu Server VM has 6 critical vulnerabilities and 9 high.

1. The first critical vulnerability Nessus found is actually multiple CVEs related to Python.

| CVSS 3.0 Score | Attack Vector | Attack Complexity | Privileges Required | User Interaction | Scope | Confidentiality | Integrity | Integ rity |
|----------------|---------------|-------------------|---------------------|------------------|-----------|-----------------|-----------|------------|
| Score: 9.8 | Network | Low | None | None | Unchanged | High | High | High |

2. The second critical vulnerability is related to Kerberos which is a network authentication protocol used to control access to network resources. It should be noted that I do not use Kerberos in my lab environment

| CVSS 3.0 Score | Attack Vector | Attack Complexity | Privileges Required | User Interaction | Scope | Confidentiality | Integrity | Ava lability |
|----------------|---------------|-------------------|---------------------|------------------|-----------|-----------------|-----------|--------------|
| Score: 9.1 | Network | Low | None | None | Unchanged | High | None | Hig h |

3. The third critical vulnerability is related to the SciPy package.

| CVSS 3.0 Score | Attack Vector | Attack Complexity | Privileges Required | User Interaction | Scope | Confidentiality | Integrity | Availability |
|----------------|---------------|-------------------|---------------------|------------------|-----------|-----------------|-----------|--------------|
| Score: 9.8 | Network | Low | None | None | Unchanged | High | High | High |

4. Critical vulnerability four is related to the wget package.

| CVSS 3.0 Score | Attack Vector | Attack Complexity | Privileges Required | User Interaction | Scope | Confidentiality | Integrity | Availability |
|----------------|---------------|-------------------|---------------------|------------------|-----------|-----------------|-----------|--------------|
| Score: 9.1 | Network | Low | None | None | Unchanged | High | High | None |

5. Critical vulnerability 5 is related to the BusyBox package.

| CVSS 3.0 Score | Attack Vector | Attack Complexity | Privileges Required | User Interaction | Scope | Confidentiality | Integrity | Availability |
|----------------|---------------|-------------------|---------------------|------------------|-----------|-----------------|-----------|--------------|
| Score: 9.8 | Network | Low | None | None | Unchanged | High | High | High |

6. Critical Vulnerability 6 is related to the OpenSSL package.

| CVSS 3.0 Score | Attack Vector | Attack Complexity | Privileges Required | User Interaction | Scope | Confidentiality | Integrity | Availability |
|----------------|---------------|-------------------|---------------------|------------------|-----------|-----------------|-----------|--------------|
| Score: 9.1 | Network | Low | None | None | Unchanged | High | None | High |

7. Choose the most urgent vulnerabilities to remediate for the next section:

I will make my decisions regarding the order for remediation based on a few factors: 1. The CVSS base score, 2. Is the full CIA triad compromised by the vulnerability?, and lastly 3. Is the remediation easy?

| Vulnerability | CVSS Base Score | Full CIA compromised? | Remediation easy? |
|-----------------|-----------------|-----------------------|------------------------|
| Python Packages | 9.8 | Yes | Yes (update packages) |

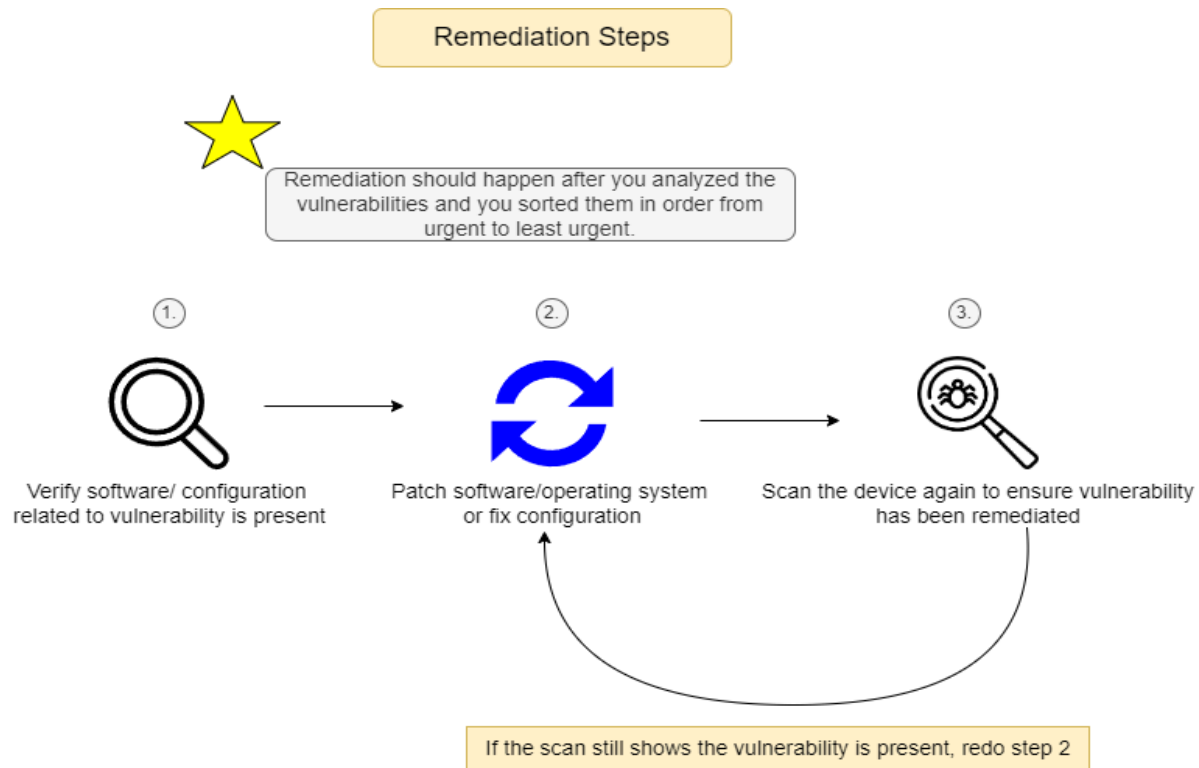
| | | | |
|----------|-----|-----|---------------------------|
| Kerberos | 9.1 | No | Yes (update packages) |
| SciPy | 9.8 | Yes | No (Requires Ubuntu Pro) |
| Wget | 9.1 | No | Yes (update packages) |
| BusyBox | 9.8 | Yes | Yes (update packages) |
| OpenSSL | 9.1 | No | Yes (update packages) |

So, I will do the Python vulnerability first, then BusyBox. I will skip SciPy since I do not have Ubuntu Pro, and I do not use it. Next I will do the Kerberos, Wget, and OpenSSL patches.

Section #6 Remediation Strategies:

Note: I will only demonstrate the remediation strategies I implemented for the Ubuntu Server VM to cut the size of this lab writeup down. Also, I will only demonstrate the steps I took to remediate the critical vulnerabilities since they are the most urgent ones that must be remediated.

Remediation Strategy:



6a. Remediate Ubuntu Server Vulnerabilities:

1. Verify the vulnerabilities are present and patch if present:

To verify the vulnerabilities are present I will create a bash script that will automate the process. The bash script will take the package name, the version needed, and if the package version needed to remediate the vulnerability is not present, the script will update the package.

| Package | Version needed for remediation |
|---------|--------------------------------|
|---------|--------------------------------|

| | |
|----------------|---|
| <p>Python</p> | <div> <div>Solution</div> <div>Update the affected packages.</div> </div> <div> <div>See Also</div> <div> https://ubuntu.com/security/notices/USN-6891-1 </div> </div> <div> <div>Output</div> <div> <pre> - Installed package : libpython3.10_3.10.12-1~22.04.3 - Fixed package : libpython3.10_3.10.12-1~22.04.4 - Installed package : libpython3.10-dev_3.10.12-1~22.04.3 - Fixed package : libpython3.10-dev_3.10.12-1~22.04.4 - Installed package : libpython3.10-minimal_3.10.12-1~22.04.3 - Fixed package : libpython3.10-minimal_3.10.12-1~22.04.4 - Installed package : libpython3.10-stdlib_3.10.12-1~22.04.3 - Fixed package : libpython3.10-stdlib_3.10.12-1~22.04.4 - Installed package : python3.10_3.10.12-1~22.04.3 - Fixed package : python3.10_3.10.12-1~22.04.4 - Installed package : python3.10-dev_3.10.12-1~22.04.3 - Fixed package : python3.10-dev_3.10.12-1~22.04.4 - Installed package : python3.10-minimal_3.10.12-1~22.04.3 - Fixed package : python3.10-minimal_3.10.12-1~22.04.4 </pre> </div> </div> |
| <p>SciPy</p> | <div> <div>Solution</div> <div>Update the affected python3-scipy package.</div> </div> <div> <div>See Also</div> <div> https://ubuntu.com/security/notices/USN-6226-1 </div> </div> <div> <div>Output</div> <div> <p>NOTE: This vulnerability check contains fixes that apply to packages only available in Ubuntu ESM repositories. Access to these package security updates require an Ubuntu Pro subscription.</p> <pre> - Installed package : python3-scipy_1.8.0-1exp2ubuntu1 - Fixed package : python3-scipy_1.8.0-1exp2ubuntu1+esm1 </pre> </div> </div> |
| <p>BusyBox</p> | <div> <div>Solution</div> <div>Update the affected packages.</div> </div> <div> <div>See Also</div> <div> https://ubuntu.com/security/notices/USN-6961-1 </div> </div> <div> <div>Output</div> <div> <pre> - Installed package : busybox-initramfs_1:1.30.1-7ubuntu3 - Fixed package : busybox-initramfs_1:1.30.1-7ubuntu3.1 - Installed package : busybox-static_1:1.30.1-7ubuntu3 - Fixed package : busybox-static_1:1.30.1-7ubuntu3.1 </pre> </div> </div> |

| | |
|----------|--|
| Kerberos | <p>Solution Update the affected packages.</p> <p>See Also https://ubuntu.com/security/notices/USN-6947-1</p> <p>Output</p> <pre> - Installed package : libgssapi-krb5-2_1.19.2-2ubuntu0.3 - Fixed package : libgssapi-krb5-2_1.19.2-2ubuntu0.4 - Installed package : libk5crypto3_1.19.2-2ubuntu0.3 - Fixed package : libk5crypto3_1.19.2-2ubuntu0.4 - Installed package : libkrb5-3_1.19.2-2ubuntu0.3 - Fixed package : libkrb5-3_1.19.2-2ubuntu0.4 - Installed package : libkrb5support0_1.19.2-2ubuntu0.3 - Fixed package : libkrb5support0_1.19.2-2ubuntu0.4 </pre> |
| Wget | <p>Solution Update the affected wget package.</p> <p>See Also https://ubuntu.com/security/notices/USN-6852-1</p> <p>Output</p> <pre> - Installed package : wget_1.21.2-2ubuntu1 - Fixed package : wget_1.21.2-2ubuntu1.1 </pre> |
| OpenSSL | <p>Solution Update the affected packages.</p> <p>See Also https://ubuntu.com/security/notices/USN-6937-1</p> <p>Output</p> <pre> - Installed package : libssl3_3.0.2-0ubuntu1.15 - Fixed package : libssl3_3.0.2-0ubuntu1.17 - Installed package : openssl_3.0.2-0ubuntu1.15 - Fixed package : openssl_3.0.2-0ubuntu1.17 </pre> |

Script:

While you can simply use apt upgrade to patch the packages, using my script is a better option for me. Using apt upgrade will update all packages that can be updated, but you

don't have control over the version installed when doing so. I created the bash script since I want to have control over the package version being installed. I also want the ability to check that either the currently installed package version is the one with the patch or not.

How does the script work?

1. The script prompts users for the number of packages they want to search for, and stores that value in the variable `num_packages`.
2. Next, using a for loop structure, the script iterates through the prompts based on the `num_packages` value. For each package the user is prompted to enter the string value for the package name and the version required for remediation.
3. The `package_version` value is used to create a new variable called `installed_version` that stores the output of the `dpkg -s` command used on `package_name`. The output is also filtered so that only the string value of the version installed is displayed.

```
GNU nano 6.2          vuln_remediation.sh *
#!/bin/bash

# This script can be used to automate the vulnerability remediation process
# on Ubuntu or other Debian based Linux systems.

# Allow user to choose how many vulnerabilities they want to search for.

echo "Enter the number of packages you want to search for:"
read num_packages

for ((i=1; i<=num_packages; i++))
do
# User will enter the package name and the version needed for remediation
echo "Enter the name for package $i:"
read package_name

echo "Enter the version for package $i needed for remediation:"
read package_version

installed_version=$(dpkg -s $package_name | awk '/^Version:/ {print $2}')

if dpkg --compare-versions "$installed_version" "lt" "$package_version"; then

echo "The installed version is $installed_version when remediation requires $package_version!"
echo "Would you like to update the package to remediate it? (yes/no)"
read response
if [ "$response" = "yes" ]; then
sudo apt-get update > /dev/null
sudo apt-get install $package_name > /dev/null
echo "The new version is now $installed_version !"
else
echo "Thank You. Bye!"
fi
fi
done
```

Fix The Vulnerabilities using my script:

1.BusyBox

Something interesting occurred! The currently installed version is the version needed to fix the vulnerability discovered by Nessus. This could possibly be a sign that the vulnerability was a false positive. A false positive means that the vulnerability scanner, in this case Nessus, thought that it found a vulnerability when in fact none was present. This also demonstrates why my script is useful for the remediation process. If I simply used apt upgrade and assumed that all the packages were fixed I would have skipped an important part of the remediation process, verifying the vulnerability is present! I would have updated a package for no reason!

```
spy3@spyserver1:~/Scripts$ sudo ./vuln_remediation.sh
[sudo] password for spy3:
Enter the number of packages you want to search for:
2
Enter the name for package 1:
busybox-initramfs
Enter the version for package 1 needed for remediation:
1:1.30.1-7ubuntu3.1
The installed version is okay since it is version 1:1.30.1-7ubuntu3.1.
Enter the name for package 2:
busybox-static
Enter the version for package 2 needed for remediation:
1:1.30.1-7ubuntu3.1
The installed version is okay since it is version 1:1.30.1-7ubuntu3.1.
spy3@spyserver1:~/Scripts$
```

2. Python Vulnerabilities

False Positive!

```
spy3@spyserver1:~/Scripts$ sudo ./vuln_remediation.sh
Enter the number of packages you want to search for:
7
Enter the name for package 1:
libpython3.10
Enter the version for package 1 needed for remediation:
3.10.12-1~22.04.4
The installed version is okay since it is version 3.10.12-1~22.04.5.
Enter the name for package 2:
libpython3.10-dev
Enter the version for package 2 needed for remediation:
3.10.12-1~22.04.4
The installed version is okay since it is version 3.10.12-1~22.04.5.
Enter the name for package 3:
libpython3.10-minimal
Enter the version for package 3 needed for remediation:
3.10.12-1~22.04.4
The installed version is okay since it is version 3.10.12-1~22.04.5.
Enter the name for package 4:
libpython3.10-stdlib
Enter the version for package 4 needed for remediation:
3.10.12-1~22.04.4
The installed version is okay since it is version 3.10.12-1~22.04.5.
```

3. Kerberos

False Positive!

```
spy3@spyserver1:~/Scripts$ sudo ./vuln_remediation.sh
Enter the number of packages you want to search for:
4
Enter the name for package 1:
libgssapi-krb5-2
Enter the version for package 1 needed for remediation:
1.19.2-2ubuntu0.4
The installed version is okay since it is version 1.19.2-2ubuntu0.4.
Enter the name for package 2:
libk5crypto3
Enter the version for package 2 needed for remediation:
1.19.2-2ubuntu0.4
The installed version is okay since it is version 1.19.2-2ubuntu0.4.
Enter the name for package 3:
libkrb5-3
Enter the version for package 3 needed for remediation:
1.19.2-2ubuntu0.4
The installed version is okay since it is version 1.19.2-2ubuntu0.4.
Enter the name for package 4:
libkrb5support0
Enter the version for package 4 needed for remediation:
1.19.2-2ubuntu0.4
The installed version is okay since it is version 1.19.2-2ubuntu0.4.
```

4. Wget

False Positive!

```
spy3@spyserver1:~/Scripts$ sudo ./vuln_remediation.sh
Enter the number of packages you want to search for:
1
Enter the name for package 1:
wget
Enter the version for package 1 needed for remediation:
1.21.2-2ubuntu1.1
The installed version is okay since it is version 1.21.2-2ubuntu1.1.
```

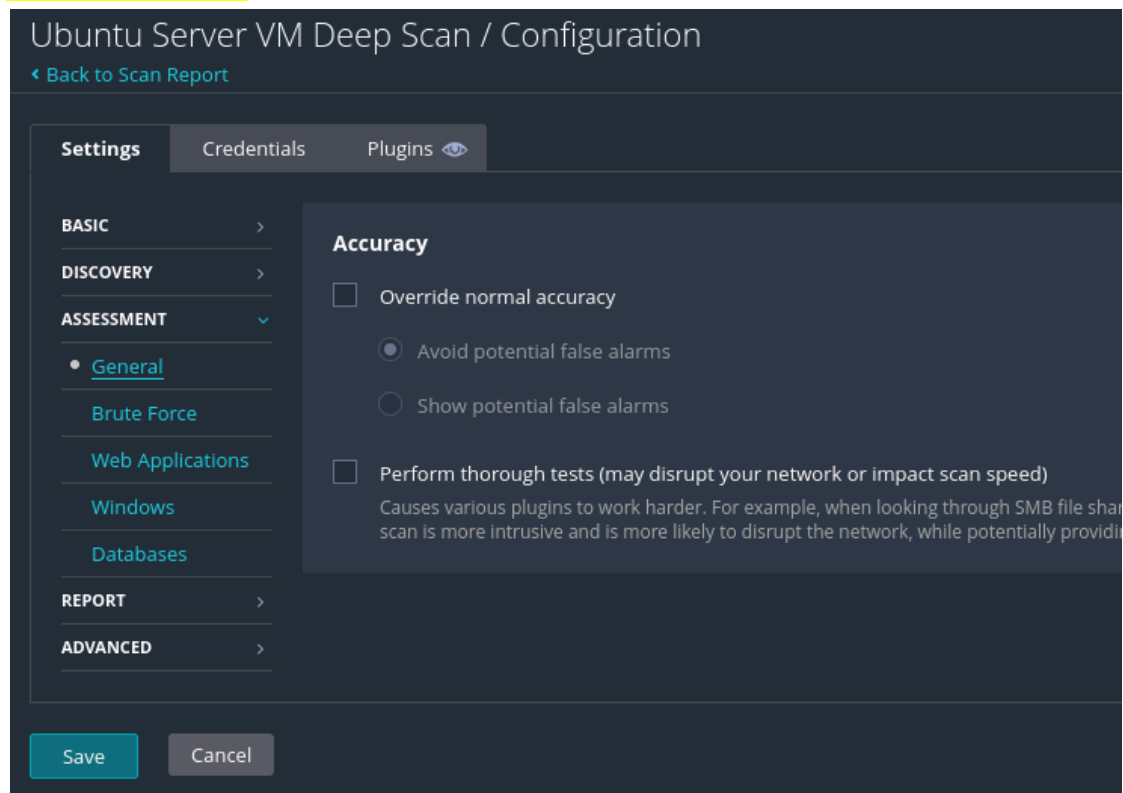
5. OpenSSH

False Positive!

```
spy3@spyserver1:~/Scripts$ sudo ./vuln_remediation.sh
Enter the number of packages you want to search for:
2
Enter the name for package 1:
libssl1
Enter the version for package 1 needed for remediation:
3.0.2-0ubuntu1.17
The installed version is okay since it is version 3.0.2-0ubuntu1.18.
Enter the name for package 2:
openssl
Enter the version for package 2 needed for remediation:
3.0.2-0ubuntu1.17
The installed version is okay since it is version 3.0.2-0ubuntu1.18.
```

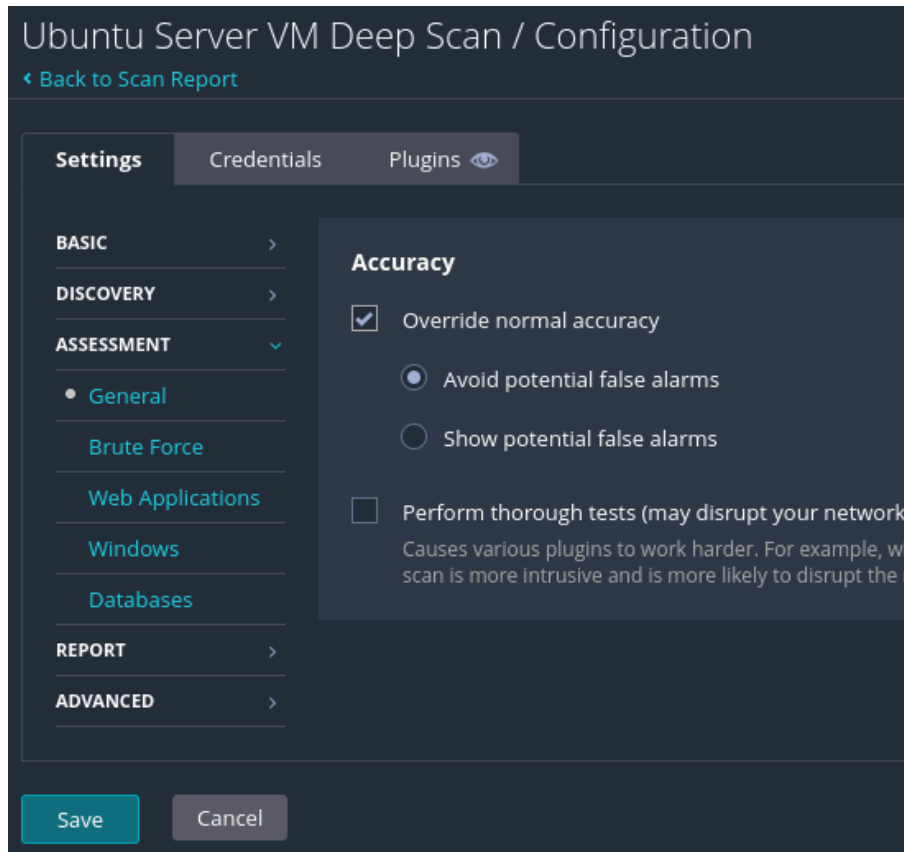
Why are all my critical vulnerabilities false positives?

I used a basic network scan and did not configure it whatsoever, I just used the default options. Apparently in order to avoid false positives you need to configure custom settings on the scan template!



2. Scan the VM again to verify the vulnerabilities are gone:

Since the critical vulnerabilities are possibly false positives, I will rescan the Ubuntu Server but I will tweak the scan settings so that false positives can be avoided.



The scan will also be credentialed again as well.

SSH User: spy3, Auth method: password

Authentication method: password

Username: spy3

Password (unsafe!):

This password could be compromised if Nessus connects to a rogue

Elevate privileges with: su

su login: root

Account to escalate to

Escalation password:

Location of su (directory): /usr/bin

Custom password prompt: password:

Some devices are configured to prompt for a password with a non-standard password prompts.

Targets to prioritize credentials: 192.168.1.22

I will now run the new scan to show that the critical vulnerabilities were simply false positives:
The new scan has no critical vulnerabilities anymore! Only one high, a few medium and one low.

Ubuntu Server VM Deep Scan

[Back to All Scans](#)

| Hosts | 1 | Vulnerabilities | 45 | History | 2 |
|--------------|-----------------|-----------------|--------|---------|----|
| Filter | Search Hosts | Q | 1 Host | | |
| Host | Vulnerabilities | | | | |
| 192.168.1.22 | 3 | | | | 55 |

Well, I learned something very important from this. You should never just assume that the vulnerabilities are actually present and you should also try to configure scans to reduce false positives so that your time can be spent on real vulnerabilities.

Section #8 Reporting and Documentation:

The last step in the vulnerability management process is to perform reporting and documentation. During reporting and documentation you should explain what steps you took during the round of vulnerability management, what you discovered, and what you should do next time to improve the efficiency and effectiveness of the vulnerability management process. For this lab I will just discuss some things that I would need to do during the next scan to improve the process.

Improvements for my next scanning process:

1. I should use a custom basic scan template and configure the scan to limit the number of false positives it finds.
 - During this round many false positives were detected on the Ubuntu Server VM and RHEL VM.
 - A lot of time was spent tracking down false positives that could have been spent on remediating actual vulnerabilities.
2. Export data from Nessus to Excel for quicker analysis/ drilldown.
 - I could have used Excel to analyze the vulnerability data more efficiently.
3. I can use CVSS 4.0 scoring for the vulnerabilities next time. I used CVSS 3.0 scoring for this lab but CVSS 3.0 is deprecated.