

Vulnerability Scanning Lab

To reinforce my understanding of the vulnerability assessment process, I set up a lab environment using some virtual machines (VMs) and the Tenable Nessus vulnerability scanner, running out of a Docker container.

Nessus Docker Image:

[tenable/nessus:latest-oracle](https://tenable.com/docker/nessus/latest-oracle) (Ver. 10.8.4-oracle)

Target OS	IP
Ubuntu	192.168.254.132
Kali	192.168.254.133
Windows Server 2019	192.168.254.134
Metasploitable 2	192.168.254.135

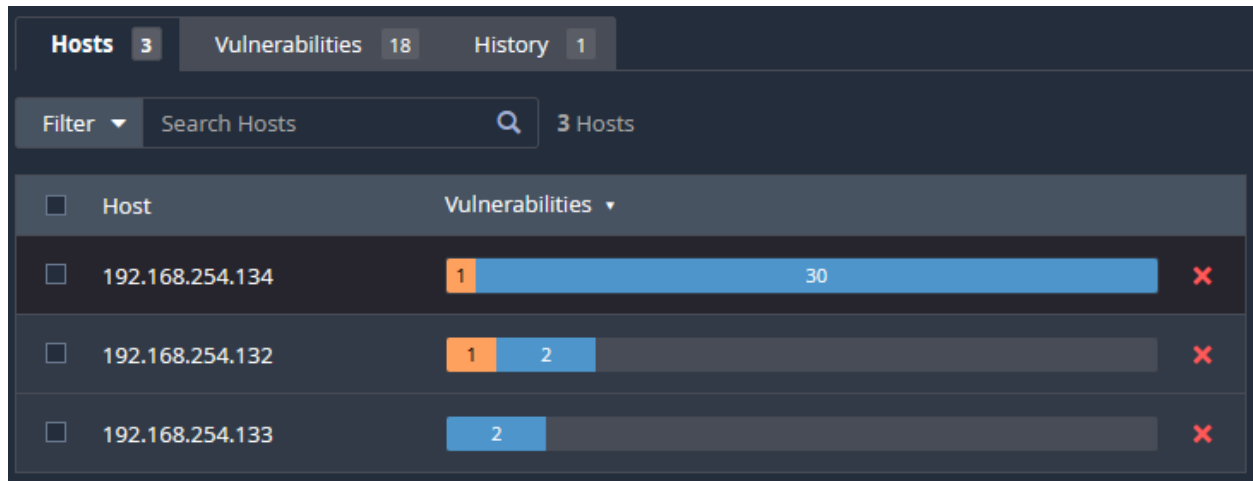
Understanding CVSS and Vulnerability Prioritization

The **Common Vulnerability Scoring System (CVSS)** assigns scores from 0 to 10 to vulnerabilities based on exploitability and impact. Nessus uses these scores to rank vulnerabilities, helping teams prioritize high and critical issues first. While a CVSS score near 10 indicates severe risk, prioritization also depends on the asset's importance. For example, dozens of medium-severity (around 7) vulnerabilities on critical employee laptops may require more urgent attention than a single critical flaw on a less important system. Balancing severity with asset context ensures effective vulnerability management and resource allocation.

Rating	Score
Critical	9.0 - 10.0
High	7.0 - 8.9
Medium	4.0 - 6.9
Low	0.1 - 3.9
None	0

Vulnerability Scanning Lab

Uncredentialed Vulnerability Reports



The screenshot shows the Nessus interface with the 'Hosts' tab selected. It displays a table of three hosts with their vulnerability counts and scores. Host 192.168.254.134 has 1 vulnerability with a score of 30. Host 192.168.254.132 has 1 vulnerability with a score of 2. Host 192.168.254.133 has 2 vulnerabilities with a score of 2. Each row has a checkbox, a host IP, a bar chart showing the count and score, and a red 'X' icon.

Host	Vulnerabilities
192.168.254.134	1 (Score: 30)
192.168.254.132	1 (Score: 2)
192.168.254.133	2 (Score: 2)

Uncredentialed scans give fewer results than credentialed scans as Nessus does not have access to the target machine. However, there are still some merits to conducting uncredentialed scans:

1. **Simulate External Attackers**

Uncredentialed scans mimic what a threat actor with no internal access can detect. This gives you visibility into publicly exposed services, banners, and versions that could be leveraged in an attack.

2. **Network Perimeter Testing**

Useful for assessing internet-facing systems like web servers, firewalls, and VPNs where credentials aren't available by default.

3. **Misconfigurations and Default Settings**

Many systems expose sensitive information (e.g. open ports, outdated services, default pages) without requiring credentials. Uncredentialed scans catch these.

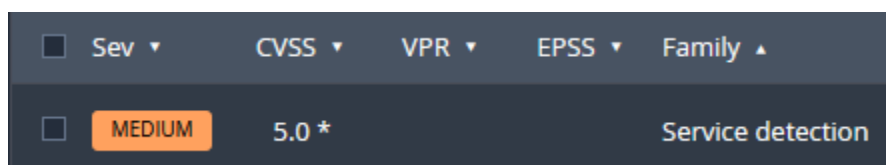
4. **Compliance Requirements**

Some standards (e.g. PCI DSS) require external vulnerability scans, which are typically uncredentialed.

5. **Lower Risk**

Uncredentialed scans don't log into systems, so they carry less operational risk in production environments.

Ubuntu (192.168.254.132)



The screenshot shows the details of a vulnerability for the host 192.168.254.132. It displays the severity as 'MEDIUM', the CVSS score as '5.0 *', and the family as 'Service detection'.

Sev	CVSS	VPR	EPSS	Family
MEDIUM	5.0 *			Service detection

Description

Vulnerability Scanning Lab

The remote service understands the Bonjour (also known as ZeroConf or mDNS) protocol, which allows anyone to uncover information from the remote host such as its operating system type and exact version, its hostname, and the list of services it is running.

This plugin attempts to discover mDNS used by hosts that are not on the network segment on which Nessus resides.

Solution

Filter incoming traffic to UDP port 5353, if desired.

Kali Linux (192.168.254.133)

<input type="checkbox"/>	Sev ▼	CVSS ▼	VPR ▼	EPSS ▼	Name ▲
<input type="checkbox"/>	INFO				Nessus Scan Information
<input type="checkbox"/>	INFO				Traceroute Information

Kali is intentionally left minimally secured to avoid interfering with penetration testing tools and workflows. As a result, vulnerability scanners like Nessus typically find very little on a standard Kali installation.

Windows Server 2019 (192.168.254.134)

<input type="checkbox"/>	Sev ▼	CVSS ▼	VPR ▼	EPSS ▼	Name ▲
<input type="checkbox"/>	MEDIUM	5.3			SMB Signing not required

Description

Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

Solution

Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.

Credentialed Windows Scan Report

<https://docs.tenable.com/nessus/Content/CredentialedChecksOnWindows.htm>

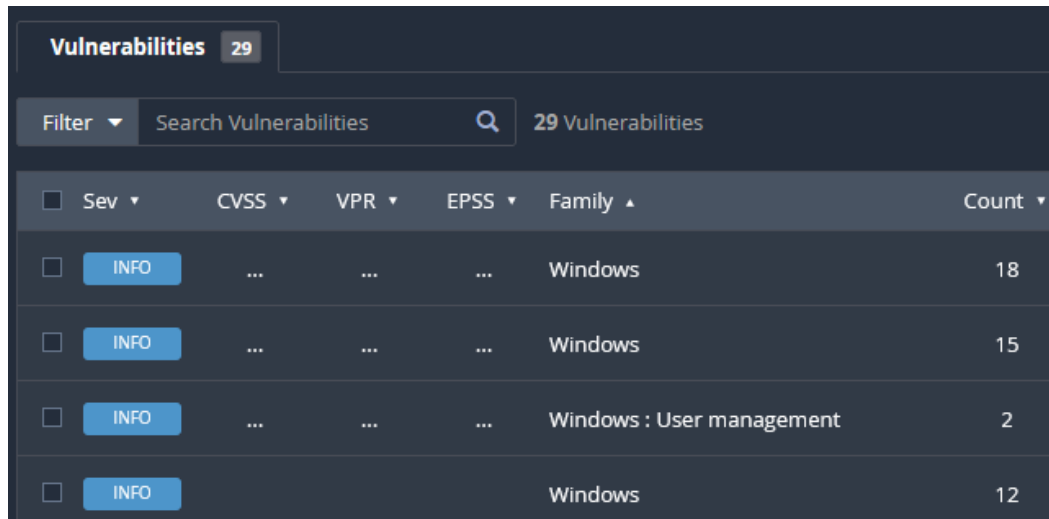
Vulnerability Scanning Lab

Followed the above Tenable guide to allow for credentialed scan on Windows Server by creating AD groups and GPOs to allow for remote registry and configure an account for authenticated scanning.

<https://docs.tenable.com/nessus/Content/ConfigureNessusForWindowsLogins.htm>

Created credentialed scan with this guide. Gave Nessus the admin username/password to run the scan with.

Results:



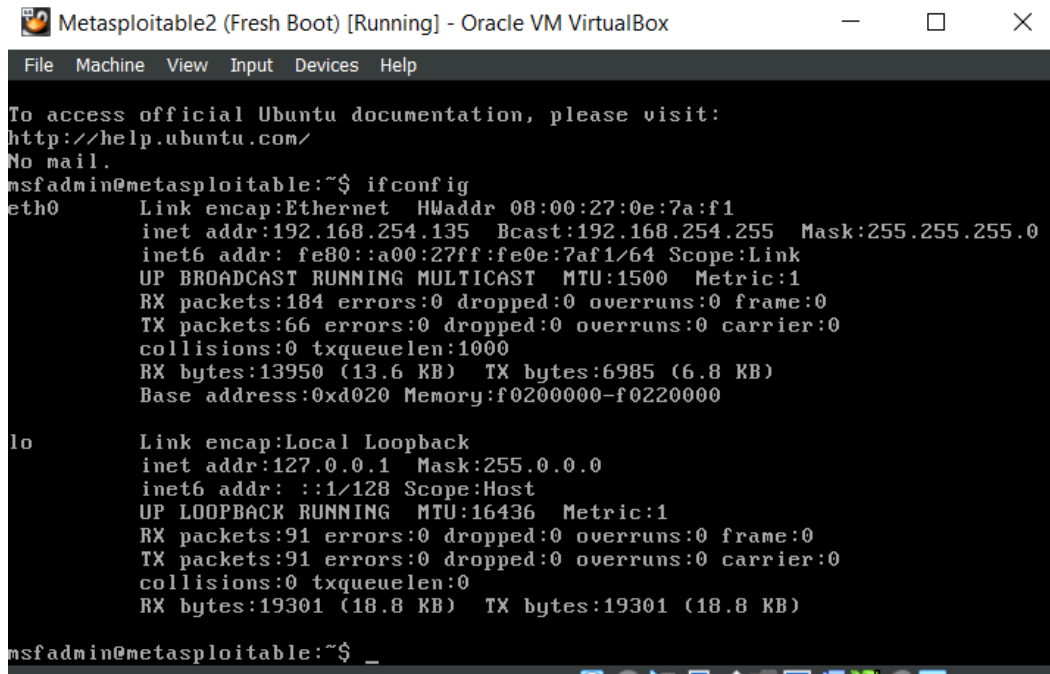
Vulnerabilities 29						
Filter	Search Vulnerabilities			29 Vulnerabilities		
<input type="checkbox"/> Sev	CVSS	VPR	EPSS	Family	Count	
<input type="checkbox"/> INFO	Windows	18	
<input type="checkbox"/> INFO	Windows	15	
<input type="checkbox"/> INFO	Windows : User management	2	
<input type="checkbox"/> INFO				Windows	12	

No vulnerabilities were discovered, but info plugins show that the credentialed scan did work. It seems Windows Server 2019 is more secure out of the box than I thought.

Credentialed Metasploitable 2 Scan Report

Since my regular VMs didn't expose enough vulnerabilities for meaningful testing, I set up Metasploitable 2 and performed a credentialed scan. While Metasploitable is highly outdated and doesn't fully reflect the types of vulnerabilities present in modern systems, it remains an excellent resource for practicing vulnerability assessment and penetration testing. It was intentionally designed to be insecure, making it ideal for learning how to identify and exploit weaknesses in a controlled environment.

Vulnerability Scanning Lab



The screenshot shows a terminal window titled "Metasploitable2 (Fresh Boot) [Running] - Oracle VM VirtualBox". The terminal output displays the results of the `ifconfig` command for the `eth0` and `lo` interfaces. The `eth0` interface is configured with IP address 192.168.254.135 and netmask 255.255.255.0. The `lo` interface is the standard loopback interface with IP 127.0.0.1.

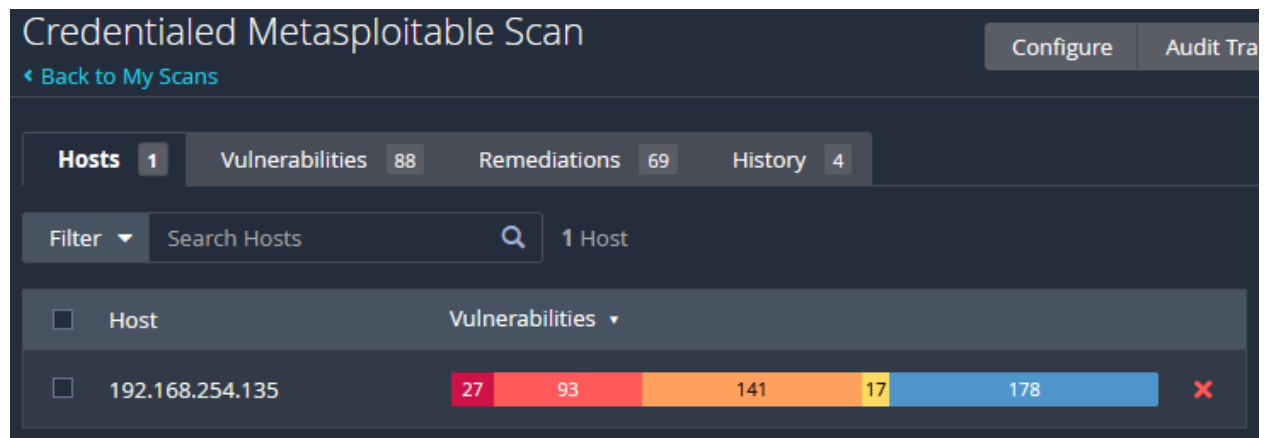
```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:0e:7a:f1
          inet addr:192.168.254.135  Bcast:192.168.254.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe0e:7af1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:184 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13950 (13.6 KB)  TX bytes:6985 (6.8 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$ _
```

Creating the VM was quick and easy as Rapid7 provides a VirtualBox image on their SourceForge page: [Metasploitable download | SourceForge.net](#)

Results:



Vulnerability Scanning Lab

<input type="checkbox"/>	CRITICAL	10.0 *	7.4	0.7216	UnrealIRCd Backdoor Detection
<input type="checkbox"/>	CRITICAL	10.0 *			VNC Server 'password' Password
<input type="checkbox"/>	CRITICAL	9.8	9.5	0.9422	Bash Remote Code Execution (Shellshock)
<input type="checkbox"/>	CRITICAL	9.8	8.9	0.9447	Apache Tomcat AJP Connector Request Injection (Ghostcat)
<input type="checkbox"/>	CRITICAL	9.8	5.1	0.0165	Weak Debian OpenSSH Keys in ~/.ssh/authorized_keys
<input type="checkbox"/>	CRITICAL	9.8			SSL Version 2 and 3 Protocol Detection
<input type="checkbox"/>	MIXED	Canonical Ubuntu Linux (Multiple Issues)
<input type="checkbox"/>	CRITICAL	SSL (Multiple Issues)
<input type="checkbox"/>	HIGH	7.5	5.9	0.7992	Samba Badlock Vulnerability
<input type="checkbox"/>	HIGH	7.5			NFS Shares World Readable
<input type="checkbox"/>	MIXED	SSL (Multiple Issues)
<input type="checkbox"/>	MIXED	ISC Bind (Multiple Issues)

The scan returned numerous High and Critical Vulnerabilities.

Shellshock

As an example, lets look at **Bash Remote Code Execution (Shellshock)**. Shellshock (CVE-2014-6271), also known as Bashdoor, is a remote code execution (RCE) vulnerability in Bash shell versions 1.0.3 to 4.3 discovered in 2014.

It can be exploited by setting an environmental variable like the following:

```
() { :;; }; /bin/bash -c "echo pwned"
```

- **() { :;; };** is the malformed function definition that triggers the bug.
- **/bin/bash -c "echo pwned"** is the payload to execute on the remote system.

Vulnerability Scanning Lab

Output

```
Nessus was able to set the TERM environment variable used in an SSH
connection to :

() { :}; /usr/bin/id > /tmp/nessus.1748660619

and read the output from the file :

uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)

Note: Nessus has attempted to remove the file /tmp/nessus.1748660619
```

Above is Nessus's output, which shows how it was able to exploit the vulnerability and verify that the system is vulnerable. Nessus then removes the temp file it made to avoid leaving behind artifacts.

This exploit can also be executed through visiting a webpage that sends the following request to a vulnerable victim:

```
GET http://example.com HTTP/1.1
User-Agent: Firefox
Host: shellshock.example.com
Referer: () { :}; echo "pwned"
```

This exploit can be remediated by simply updating the version of Bash to >4.3.

Default/Weak Credentials

<input type="checkbox"/>	CRITICAL	10.0 *	7.4	0.7216	UnrealIRCd Backdoor Detection
<input type="checkbox"/>	CRITICAL	10.0 *			VNC Server 'password' Password

IRC and VNC servers were detected with weak credentials. While this is a MAJOR security risk, it can be easily fixed by changing those passwords to stronger ones.

Ghostcat

Ghostcat (CVE-2020-1745) is a vulnerability in the AJP connector that allows unauthenticated remote attackers to access files from the web application directory. If the server also permits file uploads, an attacker could upload a file containing malicious JSP code and potentially achieve remote code execution.

Nessus checked this vulnerability by sending the following request which sees if the vulnerability can be exploited, which in this case, it was.

Vulnerability Scanning Lab

```
HTTP/1.1 /asdf/xxxxx.jsp
localhost
localhost
keep-alive
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, sdch
Cache-Control: max-age=0
Mozilla
Upgrade-Insecure-Requests: 1
Accept: text/html
localhost
javax.servlet.include.request_uri: /WEB-INF/web.xml
javax.servlet.include.path_info: /WEB-INF/web.xml
javax.servlet.include.servlet_path
```

This vulnerability can easily be fixed by restricting access to the AJP connector by enforcing authentication and updating Tomcat to version 7.0.100, 8.5.51, 9.0.31, or newer.

Conclusion

While the underlying vulnerabilities do not need to be completely understood in order to remediate them, they can be interesting to explore to gain a better understanding of how exploitation works, how attackers think, and how to better defend systems in the future.

This exercise also highlights the value of **credentialed scanning**, which allows for deeper visibility into vulnerabilities that might otherwise go undetected.

Using a deliberately vulnerable system like **Metasploitable 2** creates a safe and effective environment for learning how real-world attacks work, how scanning tools like Nessus operate, and how to prioritize and remediate threats.