ACME Corp Security Report

Group 6

Anmol Singh
System Administrator

Abbas Hussain Syed
System Administrator

Solomon Mithra
System Administrator

1. Executive Summary

Recent vulnerability scans and security assessments have identified significant risks in ACME Corp's current infrastructure and processes. As the systems administrator, we are proposing a comprehensive security plan to mitigate these threats and protect our valuable data and systems.

2. Risks Identified

1. Critical vulnerabilities in Metasploitable 2011 operating system (Nessus findings attached)
2. Lack of network segmentation and access controls
3. Outdated/unpatched software and operating system
4. No formal incident response or disaster recovery plan
5. Absence of documented security policies/standards

3. Potential Impacts

1. Data breaches/theft of sensitive information
2. System outages and downtime
3. Regulatory compliance violations
4. Damaged reputation and customer trust
5. Significant financial losses

4. Proposed Security Enhancements

4.1 Network Security:

1. Implement firewalls and intrusion detection/prevention systems
2. Segment network with VLANs and DMZs to restrict access
3. Enable encrypted communications with VPNs and SSL/TLS
4. Filter/restrict inbound/outbound traffic as per policies

4.2 System Hardening:

1. Patch management program for timely software updates
2. Secure system configurations and hardening guides
3. Endpoint protection (anti-virus, anti-malware)
4. Vulnerability scanning and penetration testing

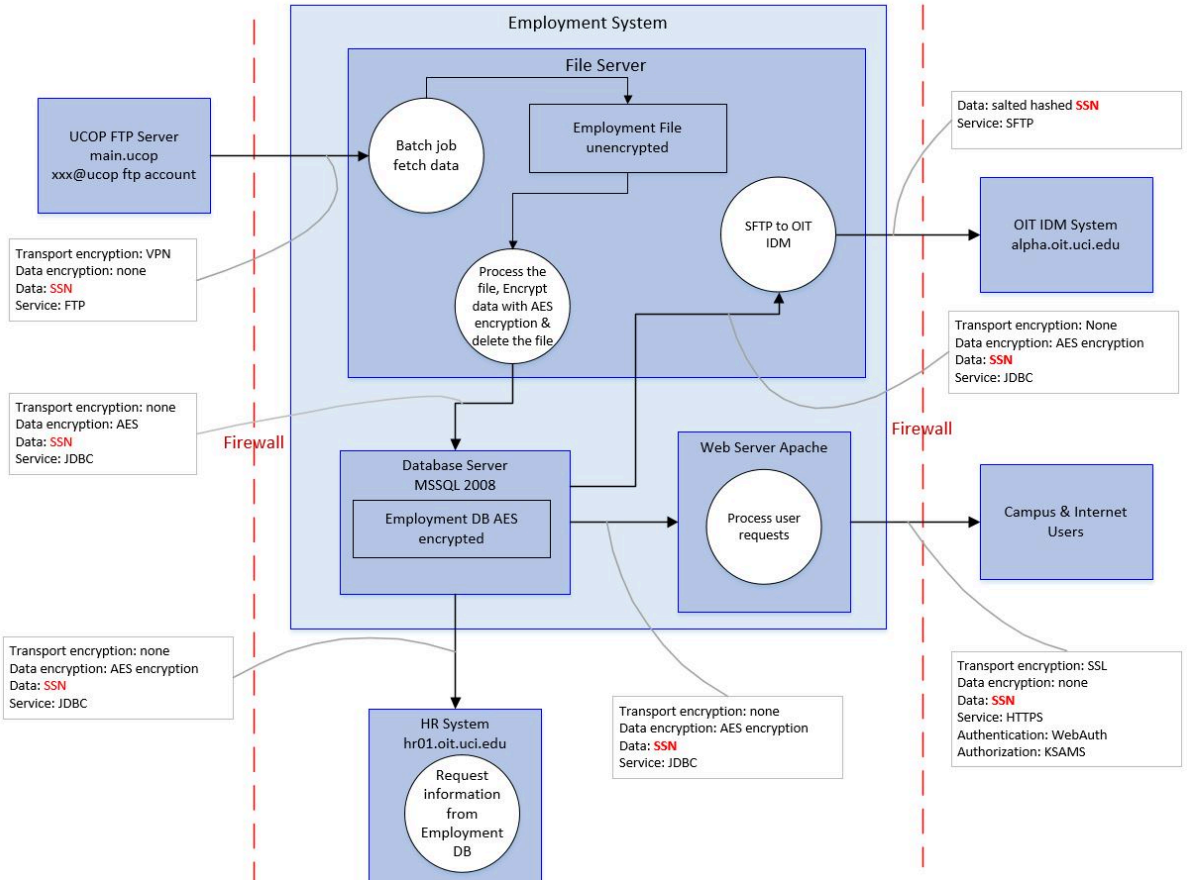
4.3 Access Controls:

1. Enforce least-privilege principles and role-based access
2. Multi-factor authentication for critical systems/data
3. Auditing and logging of all privileged activities
4. Managing and revoking access for terminated employees

4.4 Policy & Procedures:

1. Documented security policies for employees to follow
2. Incident response plan for data breach/cyber attacks
3. Business continuity and disaster recovery planning
4. Security awareness training for all employees

5. Data flow diagram for an employment system



Security implementation inspired from source:
https://www.security.uci.edu/program/risk-assessment/data-flow-diagram/

As part of our efforts to secure ACME Corp's critical systems and data flows, we have conducted a thorough analysis of the employment system's architecture and data handling processes. The attached data flow diagram illustrates the various components involved, the flow of sensitive employee data, and the security controls in place.

The employment data originates from the UCOP FTP server, where employee Social Security Numbers (SSNs) are transmitted in salted and hashed format. This data is then processed through a batch job and stored temporarily in an unencrypted file on the file server. While the transport from UCOP is secured via VPN, the data itself is not encrypted at this stage, posing a potential risk.

The unencrypted file is then processed, and the data is encrypted using AES encryption before being stored in the Employment DB on the MSSQL 2008 database server. The database is securely located within our internal network, behind a firewall, ensuring that direct external access is restricted.

The Web Server Apache component handles user requests and retrieves data from the Employment DB as needed. This data flow between the web server and the database is secured using AES encryption.

External users, such as campus and internet users, interact with the system through the web interface. Their access is secured using SSL encryption for transport, and authentication mechanisms like WebAuth and KSAMS are in place.

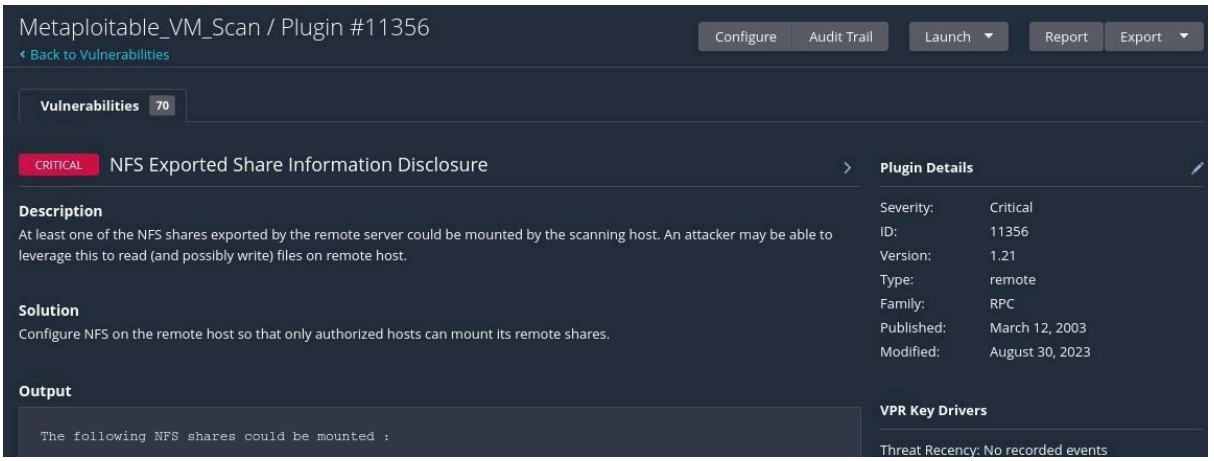
The HR System (hr01.oit.uci.edu) also interacts with the Employment DB to request and retrieve employee information. This data flow is secured using AES encryption.

While several security controls are in place, such as encryption, firewalls, and authentication mechanisms, we have identified potential areas for improvement. These include implementing end-to-end encryption for data in transit, enhancing access controls, and implementing robust logging and monitoring mechanisms.

By addressing these concerns and adhering to industry best practices, we can significantly reduce the risk of **data breaches**, **unauthorized access**, and other **security incidents**, ensuring our sensitive employment data's confidentiality, integrity, and availability.

6. Vulnerabilities Resolution

1. NFS Exported Share Information Disclosure



To fix the issue of unauthorized access to NFS shares on the Metasploitable:
Modify the NFS server configuration on the Metasploitable to only allow specific hosts to mount its shares.

```
sudo nano /etc/exports
```

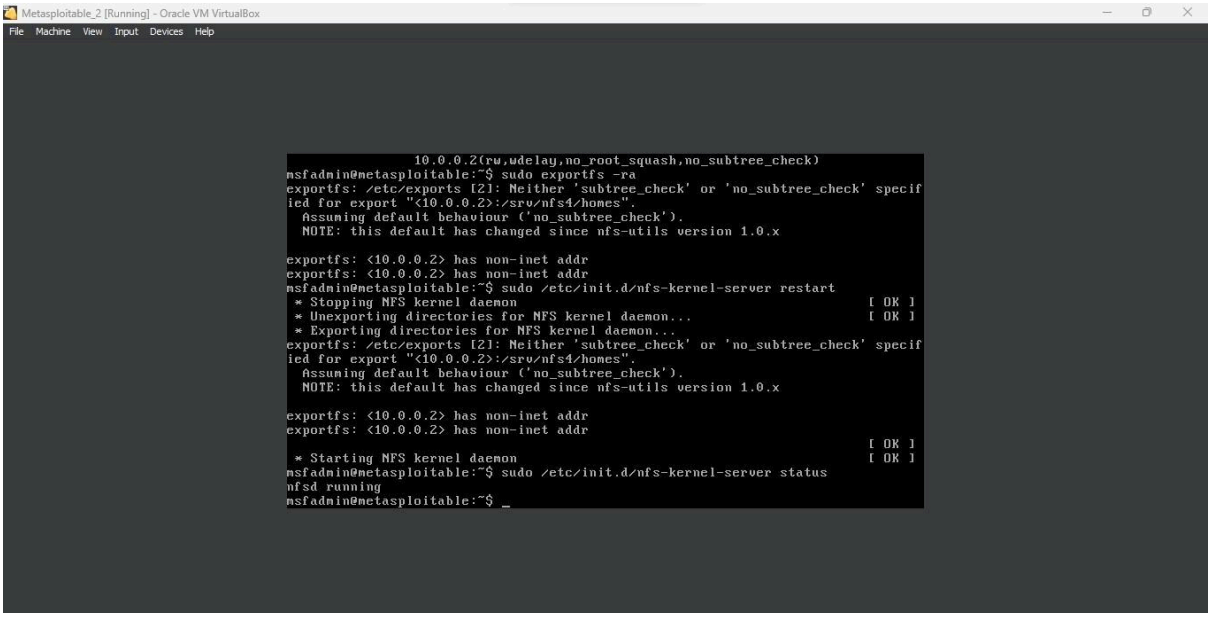
Added the entry 10.0.0.2 ip for NFS share, specifying the IP address or range of authorized host and the permission

```
/etc/exports  / 10.0.0.2(rw,sync,no_root_squash)
```

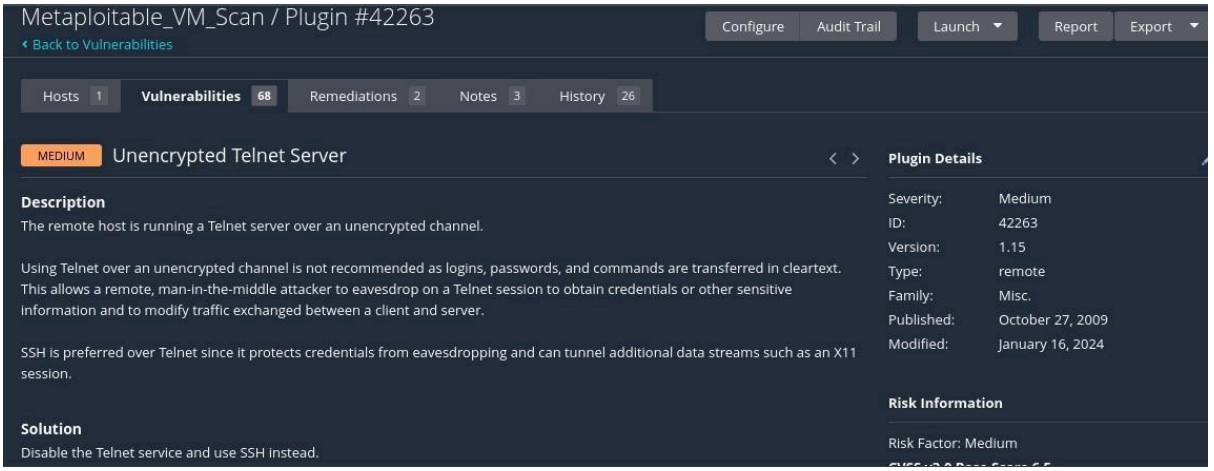
After updating the `/etc/exports` file, apply the changes by running the following command:

```
sudo exportfs -ra
```

`sudo /etc/init.d/nfs-kernel-server restart` to restart the service



2. Unencrypted Telnet Server



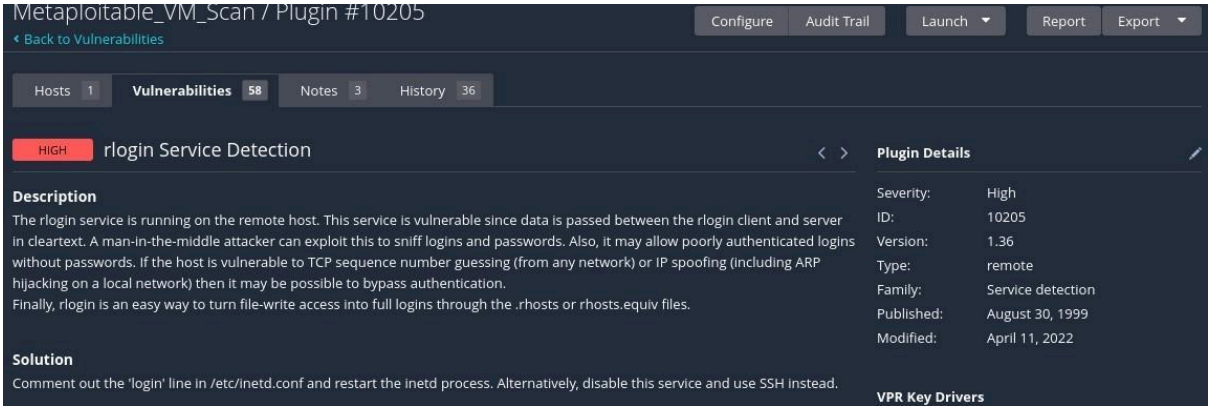
To fix the issue of having an unencrypted Telnet server on Metasploitable:

First run the command in the terminal and get into the SSH config file
`sudo nano /etc/ssh/sshd_config`

So made below service yes to no
`PermitRootLogin no`

`sudo reboot` to restart the system.

3. rlogin Service Detection



Connected to the Metasploitable using SSH

Edit the `/etc/inetd.conf` file

```
sudo nano /etc/inetd.conf
```

Commented below line in the file:

```
#login stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rlogind
```

Save the changes and exit the text editor.

Restart the inetd process to apply the changes

```
sudo service inetutils-inetd restart

sudo update-inetd --disable rlogin
```

4. OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)

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.

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.

Regenerate SSH Keys:

Remove existing SSH keys

```
sudo rm /etc/ssh/ssh_host_*
```

Regenerate SSH keys

```
sudo dpkg-reconfigure openssh-server
```

Regenerate SSL Certificates:

Remove existing SSL certificates

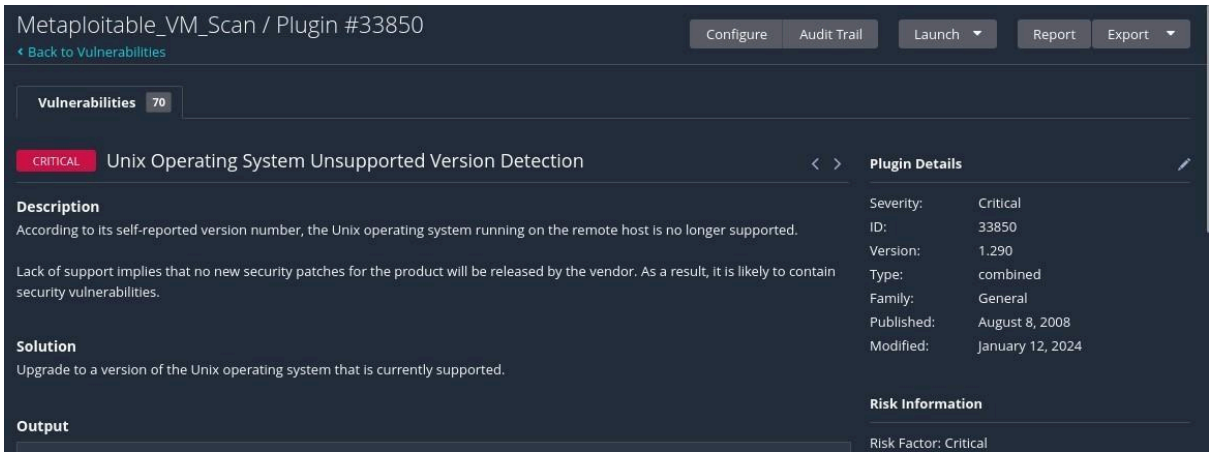
```
sudo rm /etc/ssl/private/ssl-cert-snakeoil.key /etc/ssl/certs/ssl-cert-snakeoil.pem
```

Regenerate SSL certificates

```
sudo make-ssl-cert generate-default-snakeoil
--force-overwrite
```

`sudo reboot` to restart the system.

5. Unix operating system unsupported version detected



For this vulnerability the version is Ubuntu 8.04, with the codename "Hardy Heron," is a quite old version of Ubuntu and is no longer supported by Canonical (the company behind Ubuntu). It reached its end of life on May 12, 2011, meaning it no longer receives security updates or support.
Given the age and lack of support for Ubuntu 8.04, updating the package metadata and upgrading packages through the package manager won't be possible using the usual methods.
If we are still using Ubuntu 8.04, it's strongly recommended to upgrade to a newer,supported version of Ubuntu. You can do this by performing a fresh installation of a newer Ubuntu release, such as the latest LTS (Long Term Support) version, which provides support for several years.
However, if for some reason you must continue using Ubuntu 8.04, you won't be able to update packages through the package manager. In such cases, you'll need to manually download and install updated packages or consider alternative methods for managing your software dependencies and security updates.

Metaploitable VM 2011 might be built on outdated technology or dependencies that are no longer supported or compatible with newer versions. Upgrading it to the latest version would require significant re-architecture and redevelopment, which might not be feasible or cost-effective.

Older versions of software often have known security vulnerabilities that have been addressed in newer versions. Continuing to use an outdated version could expose the system to potential security breaches and compromises. Upgrading to the latest version would ensure that the system is equipped with the latest security patches and features.

The infrastructure or software stack upon which Metaploitable VM 2011 is built might not be compatible with the latest versions of other essential software or platforms. Attempting to upgrade without addressing these compatibility issues could result in system instability or functionality issues.

Upgrading a system requires resources such as time, manpower, and potentially financial investment. If there are constraints on these resources, it may not be feasible to undertake the upgrade process, especially if the benefits of upgrading are not deemed significant enough to justify the effort.

The software or platform on which Metaploitable VM 2011 is based may have reached its end-of-life status, meaning that it is no longer supported or maintained by its developers. In such cases, upgrading to the latest version might not be possible without significant custom development or migration efforts.

Considering these factors, it may be justifiable to conclude that Metaploitable VM 2011 is not upgradable to the latest version, and alternative strategies such as migration to a newer platform or implementing additional security measures may need to be explored to ensure the continued reliability and security of the system.

6. Debian OpenSSH/OpenSSL Package Random Number Generator:

CRITICAL Debian OpenSSH/OpenSSL Package Random Number Generator W...

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.

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.

Regenerate SSH keys:

Remove the existing SSH host keys:

```
sudo rm /etc/ssh/ssh_host_*
```

Regenerate SSH host keys:

```
sudo dpkg-reconfigure openssh-server
```

Regenerate SSL certificates:
If you're using SSL certificates for any services, such as Apache, Nginx, etc., generate new SSL certificates using a secure method. You may need to refer to the documentation of the specific service you're using.

Restart affected services: After regenerating cryptographic material, restart any affected services to apply the changes. For example, for SSH, you can restart it with:

sudo service ssh restart

Verify the changes: After regenerating keys and certificates, make sure to verify that the new cryptographic material is in use and functioning correctly. Test SSH connections, SSL connections, or any other services that use cryptographic material to ensure they are working as expected.

```
msfadmin@metasploitable:~$ sudo rm /etc/ssh/ssh_host_*
msfadmin@metasploitable:~$ sudo dpkg-reconfigure openssh-server
Creating SSH2 RSA key; this may take some time ...
Creating SSH2 DSA key; this may take some time ...
 * Restarting OpenBSD Secure Shell server sshd                          [ OK
msfadmin@metasploitable:~$
```

```
4834:error:0200100D:system library:fopen:Permission denied:bss_file.c:352:fope
'/etc/ssl/private/newkey.pem','w')
4834:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:354:
msfadmin@metasploitable:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa
048 -keyout /etc/ssl/private/newkey.pem -out /etc/ssl/certs/newcert.pem
Generating a 2048 bit RSA private key
.............................................+++
.......................................................................+++
writing new private key to '/etc/ssl/private/newkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CALIFORNIA
Locality Name (eg, city) []:STOCKTON
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GROUP6
Organizational Unit Name (eg, section) []:GROUP6
Common Name (eg, YOUR name) []:GROUP6
Email Address []:comp279@u.pacific.edu
msfadmin@metasploitable:~$
```

7. VNC server ‘password’ password

CRITICAL VNC Server 'password' Password

Description

The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

Solution

Secure the VNC service with a strong password.

Steps taken:

1. Locate the configuration directory:

In many cases, VNC server configuration files are stored in a directory such as /root/.vnc/passwd. used below command to find the .vnc file which has the password details

```
ls -l ~/.vnc/passwd
```

2. Edit the .vnc file:

Use a text editor to open the .vnc file. You can use nano, vim, or any other text editor of your choice.

```
sudo nano /root/.vnc/passwd
```

3. Locate the line specifying the VNC password:

Search for a line in the vnc.conf file that specifies the VNC password. It might look something like this:

```
password=COMP@279@group6
```

4. Restart the system:

Restart the system to ensure that all services, including the VNC server, start up correctly after making changes.

```
sudo reboot
```

```
GNU nano 2.0.7          File: /root/.vnc/passwd

password=COMP@279@group6
```

Hosts 1 | Vulnerabilities 64 | Remediations 2 | Notes 3 | History 3

Filter | Search Hosts | 1 Host

Host	Vulnerabilities
10.0.0.3	6 6 21 10 135

Result:

Successfully we fixed the vnc service vulnerability by defining strong password in `/root/.vnc/passwd`

8. Samba Badlock Vulnerability

HIGH **Samba Badlock Vulnerability**

Description

The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

Solution

Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

We used below command to download latest samba service

```
wget http://download.samba.org/pub/samba/samba-latest.tar.gz
```

After online research it is not possible to update system libraries on an older system like Metasploitable 2011. Above commands have failed on the Metasploitable. Check below VM results for the command.

```
          => `samba-latest.tar.gz'
Resolving download.samba.org... 144.76.82.148
Connecting to download.samba.org|144.76.82.148|:443... connected.
Unable to establish SSL connection.
msfadmin@metasploitable:~$ wget --no-check-certificate https://download.samb
/pub/samba/samba-latest.tar.gz
--01:59:19--  https://download.samba.org/pub/samba/samba-latest.tar.gz
          => `samba-latest.tar.gz'
Resolving download.samba.org... 144.76.82.148
Connecting to download.samba.org|144.76.82.148|:443... connected.
Unable to establish SSL connection.
msfadmin@metasploitable:~$ curl -O https://download.samba.org/pub/samba/samb
test.tar.gz
curl: (77) error setting certificate verify locations:
  CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: none

msfadmin@metasploitable:~$ sudo apt update
[sudo] password for msfadmin:
sudo: apt: command not found
msfadmin@metasploitable:~$ sudo apt upgrade
sudo: apt: command not found
msfadmin@metasploitable:~$ sudo update
sudo: update: command not found
msfadmin@metasploitable:~$ _
```

Result: We tried above steps and commands but as this metasploitable version is old due to which the system requires SSL Libraries to get updated which is not possible so we cannot fix this vulnerability.

9. SSL Version 2 and 3 Protocol Detection:

CRITICAL SSL Version 2 and 3 Protocol Detection

Description

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.

- Insecure session renegotiation and resumption schemes.

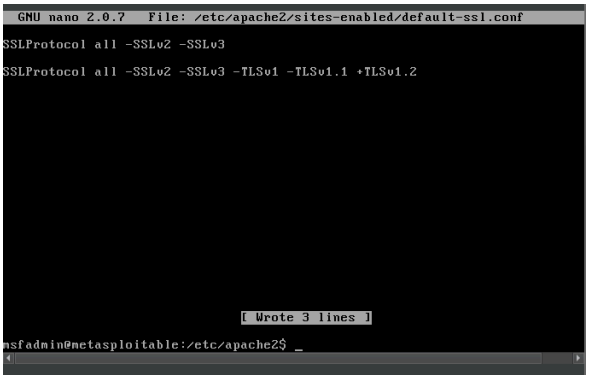
An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

Solution

Consult the application's documentation to disable SSL 2.0 and 3.0.
Use TLS 1.2 (with approved cipher suites) or higher instead.



Conclusion:

We are still unable to generate the certificate files, and the above process creates a self signed certificate and the self signed certificate is not considered safe. Instead we have to take the public key and go to the trusted site like go daddy and they issue the certificate. Then we have to execute the above commands to fix the vulnerability.

10. Msfadmin Password Change (Critical)

It is very critical to change password from default password so that it is not accessible for attackers to login using the default credentials.

Steps taken:

Log In to the VM: Access the Metasploitable using the current username and password.

Open a Terminal or Command Prompt: Depending on the operating system of the VM, you'll need to open a terminal or command prompt.

Change Password: Once you have the terminal or command prompt open, you'll use the passwd command to change the password. Type the following command and press Enter:

```
sudo passwd msfadminb
```

Enter Current Password: You'll be prompted to enter the current password. Type in the current password and press Enter.

Enter New Password: After entering the current password, you'll be prompted to enter the new password. Type in the new password and press Enter.

Confirm New Password: You'll be asked to confirm the new password by typing it again. Type in the new password once more and press Enter.

Password Changed: If everything was successful, you should see a message indicating that the password has been changed.

Test the New Password: Log out of the Metasploitable and log back in using the new password to ensure it was changed successfully.

```
Last login: Wed Apr 10 23:00:25 EDT 2024 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ passwrd msfadmin
-bash: passwrd: command not found
msfadmin@metasploitable:~$ sudo passwrd msfadmin
[sudo] password for msfadmin:
sudo: passwrd: command not found
msfadmin@metasploitable:~$ sudo passwrd msfadmin
sudo: passwrd: command not found
msfadmin@metasploitable:~$ sudo passwd msfadmin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
msfadmin@metasploitable:~$ _
```

11. Apache Tomcat AJP Connector Request Injection (Ghostcat)

CRITICAL Apache Tomcat AJP Connector Request Injection (Ghostcat)

Description

A file read/inclusion vulnerability was found in AJP connector. A remote, unauthenticated attacker could exploit this vulnerability to read web application files from a vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could upload malicious JavaServer Pages (JSP) code within a variety of file types and gain remote code execution (RCE).

Solution

Update the AJP configuration to require authorization and/or upgrade the Tomcat server to 7.0.100, 8.5.51, 9.0.31 or later.

Open Terminal: Access the terminal or command prompt in the Metasploitable 2011.

Navigate to Desired Directory: Use the cd command to navigate to the directory where you want to download the Apache Tomcat archive. For example, to download it to the /tmp directory, you can use:

```
cd /tmp
```

Download the Latest Version: Use the wget command to download the latest version of Apache Tomcat. You need to provide the URL of the download link. You can find the URL on the Apache Tomcat website.

```
wget
https://downloads.apache.org/tomcat/tomcat-8/v8.5.100/bin/apac
he-tomcat-8.5.100.tar.gz
```

```
erse/source/Sources.gz  404 Not Found [IP: 185.125.190.36 80]

E: Some index files failed to download, they have been ignored, or old ones used
 instead.
msfadmin@metasploitable:/$ sudo apt-get install --reinstall ca-certificates
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ca-certificates
0 upgraded, 1 newly installed, 0 to remove and 139 not upgraded.
Need to get 98.4kB of archives.
After this operation, 569kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
  ca-certificates
Install these packages without verification [y/N]? y
Err http://us.archive.ubuntu.com hardy/main ca-certificates 20070303-0ubuntu3
  404 Not Found [IP: 91.189.91.81 80]
Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/c/ca-certificates/
ca-certificates_20070303-0ubuntu3_all.deb  404 Not Found [IP: 91.189.91.81 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-mis
sing?
msfadmin@metasploitable:/$ sudo update-ca-certificates --fresh
sudo: update-ca-certificates: command not found
msfadmin@metasploitable:/$ _
```

Result/Conclusion: We are not able to download the latest Apache Tomcat version due to old SSL certificates. Also, we tried to download the latest SSL certificates but it failed due to VM being obsolete.

12. Bind Shell Backdoor Detection

CRITICAL Bind Shell Backdoor Detection

Description

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

Solution

Verify if the remote host has been compromised, and reinstall the system if necessary.

Verify System Integrity: Before taking any actions, it's crucial to verify the integrity of the system to ensure it has not been compromised. You can use various tools and techniques for this purpose, such as checking system logs, verifying file integrity using checksums, and scanning for malware.

Reinstall the System: If you have confirmed that the system has been compromised or suspect that it may be compromised, the safest course of action is to reinstall the operating system from a trusted installation source. This will remove any existing backdoors and malicious software and provide a clean slate for configuring the system securely.

Apply Security Updates: After reinstalling the system, make sure to apply all available security updates to patch known vulnerabilities. Depending on the operating system used in the Metasploitable 2011, you can use the appropriate package management commands to update the system:

For Debian-based systems (such as Metasploitable), you can use apt-get:

```
sudo apt-get update
sudo apt-get upgrade
```

Harden System Configuration: Implement security best practices to harden the system configuration and minimize the risk of future vulnerabilities. This may include:

1. **Disabling unnecessary services and ports.**
2. **Enabling firewall rules to restrict incoming and outgoing traffic.**
3. **Configuring strong authentication mechanisms, such as enforcing password policies and using multi-factor authentication where possible.**
4. **Regularly monitoring system logs and implementing intrusion detection/prevention systems to detect and respond to suspicious activity.**
5. **Regular Security Audits: Conduct regular security audits and vulnerability assessments to identify and address any new security issues that may arise. This will help ensure that the system remains secure over time.**

13. SSL Certificate Signed Using Weak Hashing Algorithm

HIGH SSL Certificate Signed Using Weak Hashing Algorithm

Description

The remote service uses an SSL certificate chain that has been signed using a cryptographically weak hashing algorithm (e.g. MD2, MD4, MD5, or SHA1). These signature algorithms are known to be vulnerable to collision attacks. An attacker can exploit this to generate another certificate with the same digital signature, allowing an attacker to masquerade as the affected service.

Note that this plugin reports all SSL certificate chains signed with SHA-1 that expire after January 1, 2017 as vulnerable. This is in accordance with Google's gradual sunsetting of the SHA-1 cryptographic hash algorithm.

Note that certificates in the chain that are contained in the Nessus CA database (known_CA.inc) have been ignored.

Solution

Contact the Certificate Authority to have the SSL certificate reissued.

```
 -set_serial    serial number to use for a certificate generated by -x509.
 -newhdr        output "NEW" in the header lines
 -asn1-kludge   Output the 'request' in a format that is wrong but some CA's
                have been reported as requiring
 -extensions .. specify certificate extension section (override value in config
file)
 -reqexts ..    specify request extension section (override value in config file
)
 -utf8          input characters are UTF8 (default ASCII)
 -nameopt arg    - various certificate name options
 -reqopt arg    - various request text options

msfadmin@metasploitable:~$ sudo ls /etc/apache2/ssl
msfadmin@metasploitable:~$ ls /etc/apache2/ssl
msfadmin@metasploitable:~$ ls /etc/apache2/ssl/
msfadmin@metasploitable:~$ ls -l /etc/apache2/ssl/
total 0
msfadmin@metasploitable:~$ sudo openssl genrsa -out /etc/apache2/ssl/key.pem 204
8
Generating RSA private key, 2048 bit long modulus
.....................................................................+++
........+++
e is 65537 (0x10001)
msfadmin@metasploitable:~$ sudo openssl req -new -key /etc/apache2/ssl/key.pem_
```

```
msfadmin@metasploitable:~$ ls /etc/apache2/ssl
msfadmin@metasploitable:~$ ls /etc/apache2/ssl/
msfadmin@metasploitable:~$ ls -l /etc/apache2/ssl/
total 0
msfadmin@metasploitable:~$ sudo openssl genrsa -out /etc/apache2/ssl/key.pem 204
8
Generating RSA private key, 2048 bit long modulus
.....................................................................+++
........+++
e is 65537 (0x10001)
msfadmin@metasploitable:~$ sudo openssl req -new -key /etc/apache2/ssl/key.pem -
out /etc/apache2/ssl/cert.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:california
Locality Name (eg, city) []:stockton
Organization Name (eg, company) [Internet Widgits Pty Ltd]:comp279
Organizational Unit Name (eg, section) []:comp279
Common Name (eg, YOUR name) []:_
```

```
msfadmin@metasploitable:~$ sudo openssl x509 -req -days 365 -in /etc/apache2/ssl
/cert.csr -signkey /etc/apache2/ssl/key.pem -out /etc/apache2/ssl/cert.pem
Signature ok
subject=/C=US/ST=california/L=stockton/O=comp279/OU=comp279/CN=comp@279/emailAdd
ress=comp279@gmail.com
Getting Private key
msfadmin@metasploitable:~$
```

If the SSL directory (`/etc/apache2/ssl/`) exists but there are no `key.pem` and `cert.pem` files in it after attempting to generate them with OpenSSL, there might have been an issue during the generation process. Let's troubleshoot the situation:

1. Check for Errors: After running the OpenSSL command, it should output information about the certificate generation process. Check if there were any error messages or warnings that might indicate why the files were not generated.

2. Verify Permissions: Ensure that you have the necessary permissions to write files to the `/etc/apache2/ssl/` directory. You can use the `ls -l` command to check the permissions:

```
ls -l /etc/apache2/ssl/
```

Make sure that your user has write permissions (`w`) for the directory.

3. Retry the Command: If there were no obvious errors and you have the correct permissions, try running the OpenSSL command again:

```
sudo openssl req -new -newkey rsa:2048 -keyout /etc/apache2/ssl/key.pem -out /etc/apache2/ssl/cert.pem -days 365 -nodes -x509 -md md5
```

Double-check that you're running the command with `sudo` to ensure elevated privileges.

4. Check Disk Space: Verify that there is enough disk space available on the system. If the disk is full, the certificate files may not be generated successfully.

5. Manual Generation: If OpenSSL continues to fail to generate the certificate files, you can try generating them manually using the following commands:

```
sudo openssl genrsa -out /etc/apache2/ssl/key.pem 2048
```

```
sudo openssl req -new -key /etc/apache2/ssl/key.pem -out /etc/apache2/ssl/cert.csr
```

```
sudo openssl x509 -req -days 365 -in /etc/apache2/ssl/cert.csr -signkey /etc/apache2/ssl/key.pem -out /etc/apache2/ssl/cert.pem
```

These commands generate a private key (`key.pem`), a certificate signing request (`cert.csr`), and a self-signed certificate (`cert.pem`) using the key. Ensure you replace `/etc/apache2/ssl/` with the correct directory path if it differs.

6. Check System Logs: Review system logs (`/var/log/syslog`, `/var/log/messages`, or Apache error logs) for any relevant error messages that might provide clues as to why the certificate files were not generated.

Conclusion:

We are still unable to generate the certificate files, and the above process creates a self signed certificate and the self signed certificate is not considered safe. Instead we have to take the public key and go to the trusted site like go daddy and they issue the certificate.

14. (Info) FTP Server Detection

Justification: Isolate the port of FTP to one network

15. (Info) TFTP Daemon Detection

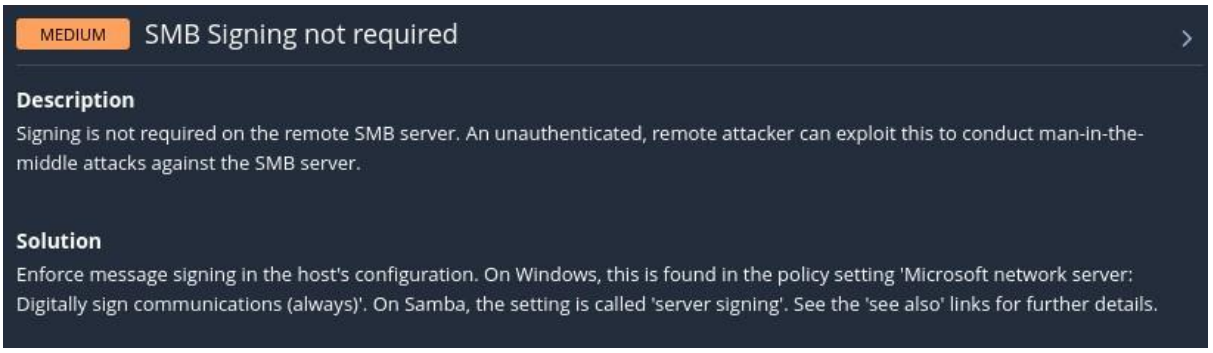
Description

The remote host is running a TFTP (Trivial File Transfer Protocol) daemon. TFTP is often used by routers and diskless hosts to retrieve their configuration. It can also be used by worms to propagate.

Solution

Disable this service if you do not use it.

16. SMB Signing not required



Edit the Samba configuration file: Use a text editor to modify the Samba configuration file, typically located at /etc/samba/smb.conf. You may need root privileges to edit this file.

```
sudo nano /etc/samba/smb.conf
```

Locate the global section: Within the smb.conf file, find the [global] section where global configuration options are defined.

Add or modify the server signing parameter: If the server signing parameter already exists, ensure it is set to "mandatory". If it does not exist, add it under the [global] section.

```
server signing = mandatory
```

Save and exit the editor: After making the changes, save the smb.conf file and exit the text editor.

Restart the Samba service: To apply the changes, restart the Samba service using the following command:

```
sudo service smbd restart
```

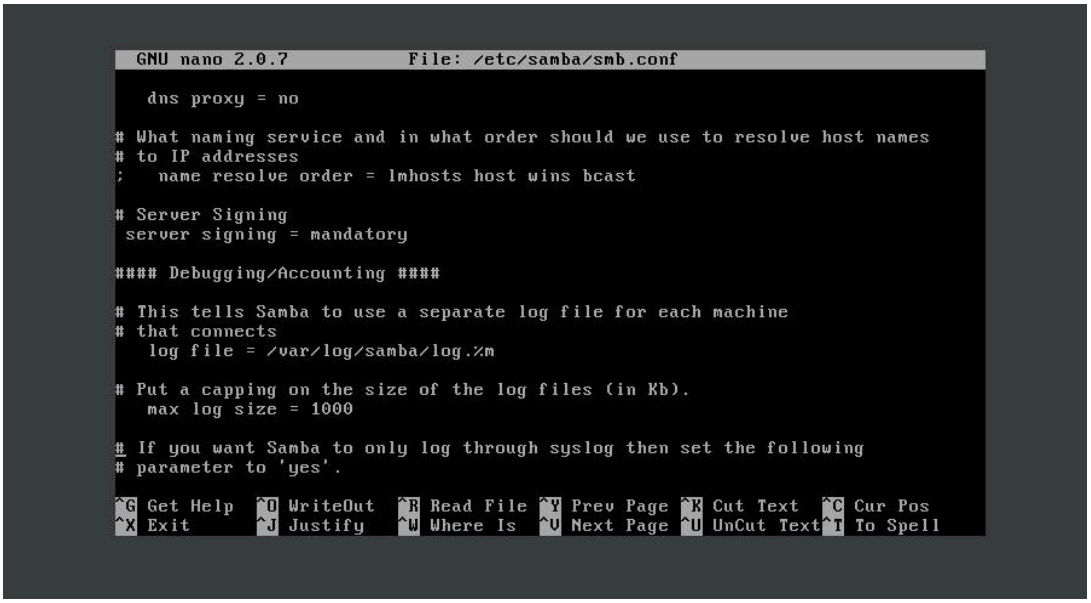
Verify SMB signing: Once the service has restarted, verify that SMB signing is enforced by connecting to the SMB server and ensuring that signing is required for all communications.

By following these steps, you can enforce SMB signing on your Metasploitable 2011, thereby mitigating the risk of man-in-the-middle attacks against the SMB server.

Restart Samba using the appropriate method:

If init is being used, you can restart the Samba service using the /etc/init.d/ scripts. You can try:

```
sudo /etc/init.d/samba restart
```



17. HTTP TRACE/TRACK Methods Allowed

MEDIUM HTTP TRACE / TRACK Methods Allowed

Description

The remote web server supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods that are used to debug web server connections.

Solution

Disable these HTTP methods. Refer to the plugin output for more information.

To resolve the issue of allowing HTTP TRACE and TRACK methods on your Metasploitable 2011, you can follow these steps:

1. Edit the Apache configuration file: Modify the Apache configuration file to disable the TRACE and TRACK methods. The Apache configuration file is typically located at `/etc/apache2/apache2.conf`.

```
sudo nano /etc/apache2/apache2.conf
```

2. Locate the `<Directory>` directive for your web directory: Within the Apache configuration file, find the `<Directory>` directive that corresponds to the directory where your web files are located. This is often `/var/www` or similar.

3. Add directives to disable TRACE and TRACK methods: Add the following directives within the `<Directory>` section to explicitly disable the TRACE and TRACK methods:

```
<Directory /var/www>
      # Disable TRACE and TRACK methods
      RewriteEngine On
      RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
      RewriteRule .* - [F]
  </Directory>
```

These directives use mod_rewrite to block any requests using the TRACE or TRACK methods and respond with a "Forbidden" (HTTP 403) status code.

4. Save and exit the editor: After making the changes, save the `apache2.conf` file and exit the text editor.

5. Restart Apache: To apply the changes, restart the Apache service using the following command:

```
sudo service apache2 restart
```

6. Verify the configuration: Test whether TRACE and TRACK methods are now disabled by attempting to send requests using these methods. You should receive a "403 Forbidden" response.

By following these steps, you can disable the HTTP TRACE and TRACK methods on your Metasploitable 2011, thereby enhancing the security of your Apache web server.

```
  GNU nano 2.0.7          File: /etc/apache2/apache2.conf

# even on a per-VirtualHost basis.  The default include files will display
# your Apache version number and your ServerAdmin email address regardless
# of the setting of ServerSignature.
#
# The internationalized error documents require mod_alias, mod_include
# and mod_negotiation.  To activate them, uncomment the following 30 lines.
     <Directory "/var/www/">
        RewriteEngine On
        RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
        RewriteRule .* - [F]
     </Directory>
     TraceEnable off_
#    Alias /error/ "/usr/share/apache2/error/"
#
#    <Directory "/usr/share/apache2/error">
#        AllowOverride None
#        Options IncludesNoExec
#        AddOutputFilter Includes html
#        AddHandler type-map var
#        Order allow,deny

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

18. TLS Version 1.0 Protocol Detection

MEDIUM TLS Version 1.0 Protocol Detection

Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

Commands used: `sudo nano /etc/apache2/apache2.conf`

We used the below command to open the apache2/apache2.conf file to add below lines

`SSLProtocol TLSv1.2`

```
  GNU nano 2.0.7          File: /etc/apache2/apache2.conf                Modified

#     ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
#     ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
#     ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
#     ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
#     ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var



# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
Include /etc/apache2/conf.d/

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/

SSLProtocol TLSv1.2



^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Conclusion:

The above fix needs a new SSL certificate which should be downloaded from the server and installed in the Metasploitable. The above process does not accept machine generated self signed certificate.

19. SSH Weak Algorithms Supported

MEDIUM SSH Weak Algorithms Supported

Description

Nessus has detected that the remote SSH server is configured to use the Arcfour stream cipher or no cipher at all. RFC 4253 advises against using Arcfour due to an issue with weak keys.

Solution

Contact the vendor or consult product documentation to remove the weak ciphers.

Access SSH server configuration:

Log in to the Metasploitable using SSH or any other method you prefer.

Navigate to the SSH server configuration file. Typically, it's located at /etc/ssh/sshd_config.

Edit the SSH configuration file:

Open the SSH configuration file using a text editor such as nano or vi. For example:

```
sudo nano /etc/ssh/sshd_config
```

Disable weak algorithms:

Search for any lines that reference the Arcfour cipher or any other weak cipher suites. These lines might look like:

```
Ciphers arcfour
```

Comment out or remove any lines that include Arcfour or other weak ciphers. You can use stronger ciphers such as AES.

```
# Ciphers arcfour
Ciphers "aes128-ctr,aes192-ctr,aes256-ctr"
```

This way, the arcfour cipher is commented out, and the stronger AES ciphers are listed correctly in the sshd_config file.

Restart SSH service:

After making changes, save the configuration file and exit the text editor.

Restart the SSH service to apply the changes:

```
sudo service ssh restart
```

Verify changes:

Test SSH access to ensure that the changes haven't disrupted SSH connectivity.

You can also use tools like Nessus or similar vulnerability scanners to re-scan the system and verify that the weak algorithms are no longer supported.

Monitor for any issues:

Keep an eye on SSH access logs and system behavior after making the changes to ensure there are no unexpected issues.

```
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

Subsystem sftp /usr/lib/openssh/sftp-server

UsePAM yes

Ciphers aes128-ctr,aes192-ctr,aes256-ctr

                                [ Wrote 79 lines ]

msfadmin@metasploitable:~$ sudo /etc/init.d/ssh restart
 * Restarting OpenBSD Secure Shell server sshd                          [ OK ]
msfadmin@metasploitable:~$ sudo service ssh restart
sudo: service: command not found
msfadmin@metasploitable:~$
```

20. SSH Weak Key Exchange Algorithms Enabled

LOW SSH Weak Key Exchange Algorithms Enabled

Description

The remote SSH server is configured to allow key exchange algorithms which are considered weak.

This is based on the IETF draft document Key Exchange (KEX) Method Updates and Recommendations for Secure Shell (SSH) RFC9142. Section 4 lists guidance on key exchange algorithms that SHOULD NOT and MUST NOT be enabled. This includes:

diffie-hellman-group-exchange-sha1

diffie-hellman-group1-sha1

gss-gex-sha1-*

gss-group1-sha1-*

gss-group14-sha1-*

rsa1024-sha1

Note that this plugin only checks for the options of the SSH server, and it does not check for vulnerable software versions.

Solution

Contact the vendor or consult product documentation to disable the weak algorithms.

Adding the KexDHMin and KexDHMax directives to the SSHD configuration file, you can further enhance the security of the key exchange process by specifying stronger key exchange algorithms using the KexAlgorithms directive.

Access SSH server configuration:

Log in to the Metasploitable via SSH or any other method you prefer.

Navigate to the SSH server configuration file /etc/ssh/sshd_config.

Edit the SSH configuration file:

Open the SSH configuration file using a text editor like nano or vi. For example:

```
sudo nano /etc/ssh/sshd_config
```

Add the stronger key exchange algorithms:

Add the KexAlgorithms directive and specify the stronger key exchange algorithms such as diffie-hellman-group14-sha256 and diffie-hellman-group16-sha512, separated by commas.

```
KexAlgorithms
diffie-hellman-group14-sha256,diffie-hellman-group16-sha512
```

Save and exit:

Save the changes to the sshd_config file and exit the text editor.

Restart SSH service:

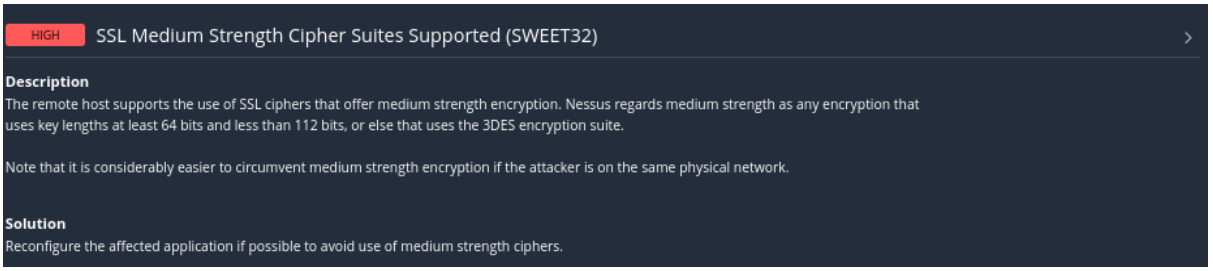
Restart the SSH service to apply the changes:

```
sudo service ssh restart
```

By adding the KexAlgorithms directive with stronger key exchange algorithms, you ensure that the SSH server negotiates key exchanges using more secure algorithms, further enhancing the security of your SSH connections.

Conclusion:
we are unable to use KexDHMin, KexDHMax, and KexAlgorithms, you can still improve the security of your SSH server by updating to a **newer version of OpenSSH** if possible. Newer versions often include security enhancements and support for more advanced configuration options.

21. SSL Medium Strength Cipher Suites Supported (SWEET32)



Identify the SSL/TLS Configuration File: Depending on the application you're using (e.g., Apache HTTP Server, nginx, etc.), locate the SSL/TLS configuration file. Common locations include /etc/apache2/apache2.conf, /etc/nginx/nginx.conf, or specific virtual host configuration files.

Backup the Configuration File: Before making any changes, it's a good practice to create a backup of the configuration file to revert to in case of any issues.

Edit the Configuration File: Open the SSL/TLS configuration file using a text editor like nano or vi. For example:

```
sudo nano /etc/apache2/apache2.conf
```

Locate SSLCipherSuite Directive: Look for the SSLCipherSuite directive within the configuration file. This directive specifies the list of ciphers that the server will accept during SSL/TLS negotiation.

Modify the Cipher Suite: Update the SSLCipherSuite directive to exclude medium strength ciphers. You can specify a list of strong ciphers that you want to allow. For example:

```
SSLCipherSuite HIGH:!aNULL:!MD5
```

This configuration allows only high-strength ciphers and excludes ciphers with NULL authentication or MD5 hashing.

Save and Exit: After making the changes, save the file and exit the text editor.

Restart the Application: Restart the application to apply the new SSL/TLS configuration. For Apache HTTP Server, you can use the following command:

```
sudo service apache2 restart
```

Verify the Configuration: Test the SSL/TLS configuration to ensure that medium strength ciphers are disabled. You can use SSL/TLS testing tools like OpenSSL or online SSL testing services to verify the configuration.

Conclusion:
The above fix needs a new SSL certificate which should be downloaded from the server and installed in the Metasploitable. After getting the latest certificate for the VM above commands and procedure can be executed.

22. SSL Certificate Cannot Be Trusted

MEDIUM SSL Certificate Cannot Be Trusted

Description
The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

- Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

Solution
Purchase or generate a proper SSL certificate for this service.

1. Generate a Private Key: Use OpenSSL to generate a private key. You can generate a key using RSA with a specific bit size. For example, to generate a key with 2048 bits, run:

```
openssl genpkey -algorithm RSA -out private.key -pkeyopt rsa_keygen_bits:2048
```

2. Generate a Certificate Signing Request (CSR): Create a CSR using the private key generated in the previous step. Provide the required information when prompted.

```
openssl req -new -key private.key -out server.csr
```

3. Generate a Self-Signed Certificate: Use the private key and CSR to generate a self-signed certificate.

```
openssl x509 -req -days 365 -in server.csr -signkey private.key -out server.crt
```

Adjust the number of days (`-days`) according to your requirements. This command creates a certificate valid for 365 days.

4. Configure the Application: Depending on the application (e.g., Apache HTTP Server, nginx), configure it to use the generated SSL certificate (`server.crt`) and private key (`private.key`). Update the SSL/TLS configuration file of your application to point to these files.

5. Restart the Application: Restart the application to apply the changes. For example, for Apache HTTP Server:

```
sudo service apache2 restart
```

6. Verify the Configuration: Test the SSL/TLS configuration to ensure that the self-signed certificate is being used and is trusted.

By following these steps, you can generate a self-signed SSL certificate and configure your application to use it, thereby fixing the SSL Certificate Cannot Be Trusted vulnerability in Metasploitable 2011. Keep in mind that while self-signed certificates provide encryption, they may not be trusted by all clients, especially in a production environment. For production use, consider purchasing a certificate from a trusted certificate authority.

Conclusion:
The above fix needs a new SSL certificate which should be downloaded from the server and installed in the Metasploitable. After getting the latest certificate for the VM above commands and procedure can be executed.

23. SSL Self-Signed Certificate

MEDIUM SSL Self-Signed Certificate

Description
The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.

Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.

Solution
Purchase or generate a proper SSL certificate for this service.

Generate a Private Key: Use OpenSSL to generate a private key. You can generate a key using RSA with a specific bit size. For example, to generate a key with 2048 bits, run:

```
openssl genpkey -algorithm RSA -out private.key -pkeyopt
rsa_keygen_bits:2048
```

Generate a Certificate Signing Request (CSR): Create a CSR using the private key generated in the previous step. Provide the required information when prompted.

```
openssl req -new -key private.key -out server.csr
```

Generate a Self-Signed Certificate: Use the private key and CSR to generate a self-signed certificate.

```
openssl x509 -req -days 365 -in server.csr -signkey
private.key -out server.crt
```

Adjust the number of days (-days) according to your requirements. This command creates a certificate valid for 365 days.

Configure the Application: Depending on the application (e.g., Apache HTTP Server, nginx), configure it to use the generated SSL certificate (server.crt) and private key (private.key). Update the SSL/TLS configuration file of your application to point to these files.

Restart the Application: Restart the application to apply the changes. For example, for Apache HTTP Server:

```
sudo service apache2 restart
```

Verify the Configuration: Test the SSL/TLS configuration to ensure that the self-signed certificate is being used and is trusted.

Conclusion:
The above fix needs a new SSL certificate which should be downloaded from the server and installed in the Metasploitable or above commands and procedure can be executed.

24. SSL RC4 Cipher Suites Supported (Bar Mitzvah)

MEDIUM SSL RC4 Cipher Suites Supported (Bar Mitzvah)

Description
The remote host supports the use of RC4 in one or more cipher suites.
The RC4 cipher is flawed in its generation of a pseudo-random stream of bytes so that a wide variety of small biases are introduced into the stream, decreasing its randomness.

If plaintext is repeatedly encrypted (e.g., HTTP cookies), and an attacker is able to obtain many (i.e., tens of millions) ciphertexts, the attacker may be able to derive the plaintext.

Solution
Reconfigure the affected application, if possible, to avoid use of RC4 ciphers. Consider using TLS 1.2 with AES-GCM suites subject to browser and web server support.

Identify the SSL/TLS Configuration File: Depending on the application you're using (e.g., Apache HTTP Server, nginx), locate the SSL/TLS configuration file. Common locations include /etc/apache2/apache2.conf, /etc/nginx/nginx.conf, or specific virtual host configuration files.

Edit the Configuration File: Open the SSL/TLS configuration file using a text editor like nano or vi. For example:

```
sudo nano /etc/apache2/apache2.conf
```

Locate SSLCipherSuite Directive: Look for the SSLCipherSuite directive within the configuration file. This directive specifies the list of ciphers that the server will accept during SSL/TLS negotiation.

Modify the Cipher Suite: Update the SSLCipherSuite directive to exclude RC4 ciphers. You can specify a list of strong ciphers that you want to allow. For example:

```
SSLCipherSuite HIGH:!RC4
```

This configuration allows only high-strength ciphers and excludes RC4 ciphers.

Save and Exit: After making the changes, save the file and exit the text editor.

Restart the Application: Restart the application to apply the new SSL/TLS configuration. For Apache HTTP Server, you can use the following command:

```
sudo service apache2 restart
```

Verify the Configuration: Test the SSL/TLS configuration to ensure that RC4 ciphers are disabled. You can use SSL/TLS testing tools like OpenSSL or online SSL testing services to verify the configuration.

Conclusion:
The above fix needs a new SSL certificate which should be downloaded from the server and installed in the Metasploitable. After getting the latest certificate for the VM above commands and procedure can be executed.

25. SSL Certificate Expiry

MEDIUM SSL Certificate Expiry

Description
This plugin checks expiry dates of certificates associated with SSL- enabled services on the target and reports whether any have already expired.

Solution
Purchase or generate a new SSL certificate to replace the existing one.

Test the SSL/TLS configuration to ensure that the new certificate is properly installed and not expired. You can use SSL testing tools like OpenSSL or online SSL testing services to verify the configuration.

Conclusion:
The above fix needs a new SSL certificate which should be downloaded from the server and installed in the Metasploitable.

26. SSL Certificate with Wrong Hostname

MEDIUM SSL Certificate with Wrong Hostname

Description
The 'commonName' (CN) attribute of the SSL certificate presented for this service is for a different machine.

Solution
Purchase or generate a proper SSL certificate for this service.

Test the SSL/TLS configuration to ensure that the new certificate with the correct hostname is properly installed and configured. You can use SSL testing tools like OpenSSL or online SSL testing services to verify the configuration.

Solution:
The above fix needs a new SSL certificate which should be downloaded from the server and installed in the Metasploitable.

7. Conclusion

By implementing these security measures, ACME Corp will enhance its resilience against cyber threats, protect sensitive data, and demonstrate a commitment to security excellence. I am confident that our efforts will fortify our infrastructure and contribute to the company's success.