Infrastructure PenTest Series : Part 1 - Intelligence Gathering

This post (always Work in Progress) lists technical steps which one can follow while gathering information about an organization.

Suppose, we are tasked with an external/ internal penetration test of a big organization with DMZ, Data centers, Telecom network etc. Moreover, the only information that we know at this moment is the company name and/or it's domain name such as example.com

Shout-outs (thanks) to the Vulnhub-ctf team, bonsaiviking, recrudesce, Rajesh and Tanoy

Scenarios

Mostly, there are only two scenarios either we are outside/ inside the organization.

Outside - External

If we are outside or doing an external pentest. We need to figure out the attack surface area first.

The could be achieved by answering the following questions:

What are the

- Domain/ subdomains present? (like example.com – domain; ftp.example.com – subdomain)
- IP Addresses/ Network ranges/ ASN Number(s) assigned?
- Different Services (open ports) running on those IP Addresses?
- Email addresses or People working for the organization?
- Different Operating Systems/ Software used in the organization?

Additionally it is also interesting to know if there have been any security breaches in the past.

Table Of Contents

Infrastructure PenTest Series : Part 1 - Intelligence Gathering
- Scenarios
  - Outside - External
  - Inside - Internal
    - Wired LAN
    - Wireless LAN
    - Responder/ Inveigh
    - NTLM/ NTLMv1/v2 / Net-NTLMv1/v2
- Fingerprinting
- Passive Fingerprinting:
  - Whois
  - ASN Number
    - Recon-ng
    - The Harvester
  - Enumeration with Domain Name (e.g. example.com) using external websites
    - DNS Dumpster API
    - Google Dorks (search operators)
    - Other Tools
    - Publicly available scans of IP Addresses
  - Reverse DNS Lookup using External Websites
    - DomainTools Reverse IP Lookup
    - PassiveTotal

We might be able to compromise user credential(s) or running vulnerable service(s) and get inside the internal network of the organization.

Inside - Internal

When we are inside the organization (let's say physically) there are two common situations:

- Potentially, posing as an employee (already having access to the internal network).
- External consultant (with no internal network access as of now).

Let's first explore what options we have as external consultant sitting in a conference room.

Wired LAN

If there's a LAN cable laying around and we (obviously) plug it in our computer, the following situations can occur:

- DHCP (Dynamic Host Configuration Protocol) is enabled and your machine is provided with an IP Address.
- DHCP is disabled; however the LAN cable is working. In this case, we might be able to sniff the network and figure out the near-by IP Address, netmask and default gateway and configure our device to use a static IP.
- Network Access Control is enabled, then probably we would need to search for
    - A device (such as printers) attached to network, clone it's MAC address and try again) or
    - IP Phones or any Hub or
    - Connect USB to LAN device to any already connected machine.
- LAN port is disabled (We can't do much here! Can we?).

Todo

explain how to clone a mac

        - Server-Sniff
        - Robtex
- Active Fingerprinting
    - Finding DNS, MX, AAAA, A using
        - host
        - nslookup
    - DNS Zone Transfer
        - host
        - Dig
        - dnsrecon
        - DNSEnum
        - SRV Records
- Internal Infrastructure Mapping
    - Internal Network Range Identification
        - Ping Gateway IP Addresses
        - DNS Enumeration
        - Internal Portal Links
        - Reverse DNS Lookup
    - Identifying Alive IP Addresses
    - Port Scanning
        - Identifying service versions
        - Performance
        - Nmap Scripts
        - Output Options
    - Exploring the Network Further
        - Gathering Screenshots for http* services
        - Information Gathering for http*

Wireless LAN

- Check for Open/ Guest Wi-Fi - If you are connected somehow try to access the internal network range(s). Most probably, the organization would have segregated the network properly. However, sometimes DNS Names can be resolved.
- Check if any WEP/ WPA2 networks are present. If so, we can try to crack them.

Once you are inside, we need to find answers to the above questions (Outside - External section).

Note

Fingerprinting can be done from both Internal/ External of the organization

Todo

explain how to crack WEP/WPA2

Responder/ Inveigh

Once, you are inside, probably the first thing would be to utilize **Responder** or **Inveigh** in **Analyze mode**.

- Responder : Responder is a LLMNR, NBT-NS and MDNS poisoner, with built-in HTTP/ SMB/ MSSQL/ FTP/ LDAP rogue authentication server supporting NTLMv1/ NTLMv2/ LMv2, Extended Security NTLMSSP and Basic HTTP authentication. It is very important to understand LLMNR, NBT-NS. Understand the basics by reading the Local Network Attacks: LLMNR and NBT-NS Poisoning, What is LLMNR & WPAD and How to Abuse Them During Pentest ? and How to get Windows to give you credentials through LLMNR

  Basically it’s like this

  - A user wants to access a file server named “NAS001” by \NAS001, however, mistakenly types \NAS01.

- Services
- NetBIOS Service
- NBTSCAN
- enum4linux
- SNMP Enumeration
- Attack Surface Area - Reconnaissance Tools
  - Aquatone: A tool for domain flyovers
  - DataSploit
  - Spiderfoot
  - Intrigue.io
- Appendix-I : Interesting Stories
  - Initial Compromise
- Changelog

- The query goes to the DNS server to resolve the IP address of NAS01. However, as it’s not a valid hostname, DNS Server responds to the user saying that it doesn’t know that host.
- The user broadcasts on the local network and asks if anyone knows who is \NAS01
- The attacker (if on the same network) seizing the opportunity says “I am NAS01 here is my IP Address”
- The user believes the attacker and sends its own username and NTMLv2 hash to the attacker.
- The attacker gathers all the hashes and cracks them (offline) to gain password.

Recently, Responder also got the functionality to act as a Multi-relay, which allows you to relay the NTLMv1/2 authentication to a specific target and possibly execute code (during a successful attack) on the target node. NotSoSecure has written a detailed blog on this technique: Pwning with Responder – A Pentester’s Guide

Similar to Python Responder, there is Inveigh

- Inveigh is a PowerShell LLMNR/ mDNS/ NBNS spoofer and man-in-the-middle tool designed to assist penetration testers and red teamers that find themselves limited to a Windows system.

Todo

eloborate on Inveigh?

NTLM/ NTLMv1/v2 / Net-NTLMv1/v2

The Responder/ Inveigh tools and the hashes NTLM/ NTLMv1/v2 / Net-NTLMv1/v2 are Windows environment specific.

Probably, we should cover this in Exploitation phase. However as we have just mentioned Responder/ Inveigh here, it makes sense to include this here

- NTLM: NTLM hashes are stored in the Security Account Manager (SAM) database and in the Domain Controller's NTDS.dit database.

  ```
  aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42
  ```

  The LM hash is the one before the semicolon (:) and the NT hash is the one after the semicolon. Starting with Windows Vista and Windows Server 2008, by default, only the NT hash is stored.

- NTLMv1/v2 / Net-NTLMv1/v2 : Net-NTLM hashes are used for network authentication (they are derived from a challenge/response algorithm and are based on the user's NT hash). Here is an example of a Net-NTLMv2 (a.k.a NTLMv2) hash:

  ```
  admin::N46iSNekpT:08ca45b7d7ea58ee:88dcbe4446168966a153a0064958dac6:
  ```

From a pentesting perspective:

- You CAN perform Pass-The-Hash attacks with NTLM hashes.
- You CANNOT perform Pass-The-Hash attacks with Net-NTLM hashes.

Todo

so Net-NTLM needs to be cracked, how?

The above has been taken from Practical guide to NTLM Relaying in 2017 (A.K.A getting a foothold in under 5 minutes) He has explained it very well and also showed how to own the network using relaying the hashes from Responder to get a system shell. Another good blog to understand this is SMB Relay Demystified and NTLMv2 Pwnage with Python

Fingerprinting

We can either do **Passive fingerprinting** (learning more about the company, without them knowing it) or **Active fingerprinting** (process of transmitting packets to a remote host and analysing corresponding replies (which very likely will be logged)).

Passive fingerprinting and **Active fingerprinting** can be done by using various methods such as:

Passive Fingerprinting	Active Fingerprinting
• whois	• Finding DNS, MX, AAAA, A
• ASN Number	• DNS Zone Transfer(s)
• Enumeration with Domain Name	• SRV Records
• Publicly available scans of IP Addresses	• Port Scanning
• Reverse DNS Lookup using External Websites	

Do you remember from earlier? We need to find answers to

Questions (What are the)	Answer
Different domain/ subdomains present?	whois, DNS-MX/AAAA/A/SRV, Enumeration with Domain Name
Different IP Address/ Network ranges/ ASN Number assigned?	DNS, ASN-Number, DNS-Zone-Transfer
Different Services/ Ports running on those IP Addresses?	Public Scans of IP/ Port Scanning
Email addresses or People working in the organization?	harvestor, LinkedIn
What are the different Operating Systems/ Software used?	FOCA
Any breaches which happened in the organization?	

The active and passive fingerprinting would help us to get those answers!

Passive Fingerprinting:

Whois

Whois provides information about the registered users or assignees of an Internet resource, such as a Domain name, an IP address block, or an autonomous system.

whois acts differently when given an IP address then a domain name.

- For a Domain name, it just provides registrar name etc.
- For a IP address, it provides the net-block, ASN Number etc.

```
whois <Domain Name/ IP Address>
-H Do not display the legal disclaimers some registries like to show you.
```

Googling for

```
"Registrant Organization" inurl: domaintools
```

Also helps for to search for new domains registered by the same organization. “Registrant Organization” is present in the output of whois. This technique was used by person who compromised FinFisher in his writeup.

Todo

Add example so people don’t have to (re)read or skim through the pastebin article

ASN Number

We could find the AS Number that participates in the Border Gateway Protocol (BGP) used by particular organization which could further inform about the IP address ranges used by the organization. An ASN Number could be found by using Team CMRU whois service

```
whois -h whois.cymru.com " -v 216.90.108.31"
```

If you want to do bulk queries refer @ IP-ASN-Mapping-Team-CYMRU

Hurricane Electric Internet Services also provide a website BGPToolkit which provides your IP Address ASN or search function by Name, IP address etc. It also provides AS Peers which might help in gathering more information about the company in terms of its neighbors.

Todo

Commandline checking of subnet and making whois query efficient.

Recon-ng

Todo

Add the following line to unconfuse people?

Todo

Note that is this context hosts are subdomains

- use recon/domains-hosts/bing_domain_web : Harvests hosts from Bing.com by using the site search operator.
- use recon/domains-hosts/google_site_web : Harvests hosts from google.com by using the site search operator.
- use recon/domains-hosts/brute_hosts : Brute forces host names using DNS.
- use recon/hosts-hosts/resolve : Resolves the IP address for a host.
- use reporting/csv : Creates a CSV file containing the specified harvested data.

Jason Haddix has created a dynamic resource script for sub-domain discovery which is available here. Simply provide the domain name and it runs the necessary modules, creates a new workspace and save the report.

Todo

Check API option too, why google_site_web is failing, add a module to add ASN Info and Location Info too.

The Harvester

The harvester provides email addresses, virtual hosts, different domains, shodan results etc. for the domain. It provides really good results, especially if you combine with shodan results as it may provide server versions and what's OS is running on a provided IP address.

```
Usage: theharvester options
   -d: Domain to search or company name
   -b: data source: google, googleCSE, bing, bingapi, pgp
                    linkedin, google-profiles, people123, jigsaw,
                    twitter, googleplus, all
   -v: Verify host name via dns resolution and search for virtual hosts
   -f: Save the results into an HTML and XML file
   -c: Perform a DNS brute force for the domain name
   -t: Perform a DNS TLD expansion discovery
   -e: Use this DNS server
   -h: use SHODAN database to query discovered hosts             |
```

Todo

Combine these results with recon-ng and DNS Dumpsters and create one csv with all results.

Enumeration with Domain Name (e.g. example.com) using external websites

If you have domain name you could use

DNS Dumpster API

We can utilize DNS Dumpster's API to know the various sub-domain related to a domain.

```
curl -s http://api.hackertarget.com/hostsearch/?q=example.com > hostsearch
```

and the various dns queries by

```
curl -s http://api.hackertarget.com/dnslookup/?q=example.com > dnslookup
```

Google Dorks (search operators)

- **site**: Get results from certain sites or domains.
- **filetype:suffix**: Limits results to pages whose names end in suffix. The suffix is anything following the last period in the file name of the web page. For example: filetype:pdf
- **allinurl/ inurl**: Restricts results to those containing all the query terms you specify in the URL. For example, [allinurl: google faq] will return only documents that contain the words "google" and "faq" in the URL, such as "www.google.com/help/faq.html".
- **allintitle/ intitle**: Restricts results to those containing all the query terms you specify in the title.

Three good places to refer are Search Operators, Advanced Operators and Google Hacking Database.

Other Tools

- Mcafee Site Digger searches Google's cache to look for vulnerabilities, errors, configuration issues,proprietary information, and interesting security nuggets on web sites.
- SearchDiggityv3 is Bishop Fox's MS Windows GUI application that serves as a front-end to the most recent versions of our Diggity tools: GoogleDiggity, BingDiggity, Bing, LinkFromDomainDiggity, CodeSearchDiggity, DLPDiggity, FlashDiggity, MalwareDiggity, PortScanDiggity, SHODANDiggity, BingBinaryMalwareSearch, and NotInMyBackYard Diggity.

Publicly available scans of IP Addresses

- Exfiltrated provides the scans from the 2012 Internet Census. It would provide the IP address and the port number running at the time of scan in the year 2012.

- Shodan: provides the same results may be with recent scans. You need to be logged-in. Shodan CLI is available at Shodan Command-Line Interface

Shodan Queries

```
title   : Search the content scraped from the HTML tag
html    : Search the full HTML content of the returned page
product : Search the name of the software or product identified in the banner
net     : Search a given netblock (example: 204.51.94.79/18)
version : Search the version of the product
port    : Search for a specific port or ports
os      : Search for a specific operating system name
country : Search for results in a given country (2-letter code)
city    : Search for results in a given city
```

Todo

Learn how to access Shodan with API

- Censys is a search engine that allows computer scientists to ask questions about the devices and networks that compose the Internet. Driven by Internet-wide scanning, Censys lets researchers find specific hosts and create aggregate reports on how devices, websites, and certificates are configured and deployed. A good feature is the Query metadata which tells the number of Http, https and other protocols found in the IP network range.

  Censys.io queries

  ```
  ip:192.168.0.0/24 -- CIDR notation
  ```

Reverse DNS Lookup using External Websites

Even after doing the above, sometimes we miss few of the domain name. Example: Recently, In one of our engagement, the domain name was example.com and the asn netblock was 192.168.0.0/24. We did recon-ng, theharvester, DNS reverse-lookup via nmap. Still, we missed few of the websites hosted on same netblock but with different domain such as example.in. We can find such entries by using ReverseIP lookup by

DomainTools Reverse IP Lookup

Reverse IP Lookup by Domaintools: Domain name search tool that allows a wildcard search, monitoring of WHOIS record changes and history caching, as well as Reverse IP queries.

PassiveTotal

Passive Total : A threat-analysis platform created for analysts, by analysts.

Server-Sniff

Server Sniff : A website providing IP Lookup, Reverse IP services.

Robtex

Robtex : Robtex is one of the world's largest network tools. At robtex.com, you will find everything you need to know about domains, DNS, IP, Routes, Autonomous Systems, etc. There's a nmap nse http-robtex-reverse-ip which can be used to find the domain/ website hosted on that ip.

```
nmap --script http-robtex-reverse-ip --script-args http-robtex-reverse-ip.host
Starting Nmap 7.01 ( https://nmap.org ) at 2016-04-20 21:39 IST
Pre-scan script results:
| http-robtex-reverse-ip:
|   xxxxxxindian.com
|_  www.xxxxxindian.com
```

Active Fingerprinting

Most probably by now we have gathered all the public available information without interacting with the client’s infrastructure. Next, we can use **DNS enumeration** to gather more information about the client. The below information could be gathered externally as well as internally. However, the amount of information gathered from internal network would definitely be more than when done externally.

Finding DNS, MX, AAAA, A using

host

```
host <domain> <optional_name_server>
host -t ns <domain>            -- Name Servers
host -t a <domain>             -- Address
host -t aaaa <domain>          -- AAAA record points a domain or subdomain to a
host -t mx <domain>            -- Mail Servers
host -t soa <domain>           -- Start of Authority
host <IP>                      -- Reverse Lookup
```

Example:

```
host -t ns zonetransfer.me
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
```

nslookup

```
nslookup - <optional_name_server>
set type=mx
set type=ns
```

DNS Zone Transfer

If a DNS server is badly configured it might be possible to get a hold of all of it records. This is interesting because if gives us an overview of what IP to hostname translations the DNS server is aware off.

DNS Zone Transfer can be done with:

host

```
host -l <Domain Name> <DNS Server>
```

Try zonetransfer using host for zonetransfer.me using their name servers.

Dig

```
dig axfr <domain_name> @nameserver
```

Try zonetransfer using dig for zonetransfer.me using their name servers.

dnsrecon

```
dnsrecon -d <domain> -t axfr
```

dnsrecon could also be used for other purposes such as finding nameservers, mailserver, forward reverse lookup

```
-d, --domain      <domain>          Domain to Target for enumeration.
-r, --range       <range>           IP Range for reverse look-up brute force i
-n, --name_server <name>            Domain server to use, if none is given the
```

DNSEnum

DNS Enumeration tool

```
dnsenum <domain>
```

SRV Records

Service record (SRV record) is a specification for data in the Domain Name System defining the location, i.e. the hostname and port number, of servers for specified services. An SRV record has the following form:

** _service._proto.name. TTL class SRV priority weight port target. **

- **Retrieving an SRV record:**

```
$ dig _sip._tls.example.com SRV

$ host -t SRV _sip._tls.example.com

$ nslookup -querytype=srv _sip._tls.example.com

$ nslookup
 > set querytype=srv
 > _sip._tls.example.com
```

- **Usage:**

  SRV records are used by the below standardized communication protocols:

```
Teamspeak 3 (since version 3.0.8 - Neither priority nor weight is ta
Minecraft (since version 1.3.1, _minecraft._tcp)
CalDAV and CardDAV
Client SMTP Authorization
DNS Service Discovery (DNS-SD)
IMPS
Kerberos
LDAP
Puppet
SIP
XMPP
Mail submission, Post Office Protocol, and Internet Message Access P
Libravatar uses SRV records to locate avatar image servers
Microsoft Lync
Citrix Receiver
```

Checkout the brute_srv function in dnsrecon tool script to get familiar with the different SRV names and services.

Internal Infrastructure Mapping

All the steps active-fingerprinting which are DNS related recon could also be performed during an internal penetration testing provided we have access to the internal DNS Server. After, we have gathered all the information from DNS enumeration. We haven't enumerated the internal infrastructure. We apply the below methods to enumerate further.

Internal Network Range Identification

In many instances, we are provided or expected to find vulnerabilities in a 10.0.0.0/8 network which would contain around 16 million IP Addresses. Scanning 16 million IP address in a considerable time is difficult. In which case, we need faster and targeted result. So, how do we find out the IP range(s)?

Ping Gateway IP Addresses

Let's say internally, we got an IP address 192.168.56.101 netmask 255.255.255.0 with a default gateway of 192.168.56.1. It is a high probability that the rest of the network ranges would have been defined as /24 CIDR as well. In that case, a ping sweep for the range of 192.168.*.1 with a watch on the TTL would possibly reveal what the other network ranges are.

```
nmap -sn -v -PE 192.168.*.1
```

Todo

Provide output example?

DNS Enumeration

If you are connected to a internal dns server, you may query it with

```
dig -t any <domainname>
```

which should result in an output containing different name servers, mail servers, A, AAAA, SOA records which would possibly give you a inner scenario how the network has been designed as there can be different nameservers, domain controllers for different locations, internal departments etc.

Todo

Convert dig output directly into hostname, ip address format.

Internal Portal Links

Most of the organizations have internal portals which serves as a one-stop spot with links to every possible portal link. This could also result in some internal range exposure.

Todo

Write the script for grep and printing host and IP address and combine it with DNS Enumeration.

Reverse DNS Lookup

Nmap provides a List scan option which does the reverse lookup. It provides the hostnames of the IP Address

```
nmap -sL 10.0.0.0/8
```

It can also be used with the below options:

```
--randomize-hosts  : make the scans less obvious to various network monitoring
--dns-servers server1,server2 : By default, it would use the dns servers which
```

Example:

```
nmap -sL 45.33.32.156

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-21 12:29 CET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Nmap done: 1 IP address (0 hosts up) scanned in 0.23 seconds
```

Identifying Alive IP Addresses

Nmap by default provides a -sn Ping scan option. Default host discovery is done with -sn which consists of an ICMP echo request, TCP SYN to port 443, TCP ACK to port 80, and an ICMP timestamp request by default. This works as if ICMP echo request is blocked, nmap would know if a host is alive if it receives any response from port 443 or 80 or timestamp reply.

Let’s see what nmap does when we do a ping scan.

```
nmap -sn -n 10.0.0.230
#My IP is 10.0.0.1
```

It is very important to mention that the -n option (No DNS resolution) should be used going forward as we have already did DNS resolution while using nmap’s List scan. Since DNS can be slow even with Nmap’s built-in parallel stub resolver, this option can slash scanning times. TCP Dump output is presented here. As both the IP addresses are in the same subnet, nmap would use ARP Ping scan to find the alive IP Address.

```
22:11:27.292054 ARP, Request who-has 10.0.0.230 (Broadcast) tell 10.0.0.1, ler
22:11:27.361100 ARP, Reply 10.0.0.230 is-at 8c:64:22:3b:2b:2d (oui Unknown), l
```

However, this behaviour can be changed using –disable-arp-ping.

```
nmap -sn 10.0.0.230 --disable-arp-ping
```

TCPdump output is as below One ICMP Echo Request, SYN to Port 443, ACK to Port 80 and a time stamp request.

```
22:14:02.742180 IP 10.0.0.1 > 10.0.0.230: ICMP echo request, id 45066, seq 0,
22:14:02.742222 IP 10.0.0.1.59246 > 10.0.0.230.https: Flags [S], seq 399442053
22:14:02.742234 IP 10.0.0.1.59246 > 10.0.0.230.http: Flags [.], ack 399442053
22:14:02.742241 IP 10.0.0.1 > 10.0.0.230: ICMP time stamp query id 38635 seq 0
22:14:02.801243 IP 10.0.0.230 > 10.0.0.1: ICMP echo reply, id 45066, seq 0, le
22:14:02.801930 IP 10.0.0.230.https > 10.0.0.1.59246: Flags [R.], seq 0, ack 3
22:14:02.805083 IP 10.0.0.230.http > 10.0.0.1.59246: Flags [R], seq 399442053
22:14:02.805930 IP 10.0.0.230 > 10.0.0.1: ICMP time stamp reply id 38635 seq 0
```

If you use the –reason option, nmap will tell why it thinks the host is alive. In the below case (received echo-reply).

```
Nmap scan report for 10.0.0.230
Host is up, received echo-reply (0.073s latency).
```

If we only want to send an ICMP Ping query (as if the host replies to it, the other three packets (SYN 443, ACK 80 and Timestamp) are an extra burden. (I may be wrong here). We can use

```
nmap -n -sn -PE --disable-arp-ping 10.0.0.230
```

TCP Dump output:

```
22:30:20.768525 IP 10.0.0.1 > 10.0.0.230: ICMP echo request, id 39366, seq 0,
22:30:20.826098 IP 10.0.0.230 > 10.0.0.1: ICMP echo reply, id 39366, seq 0, le
```

Please note that this ICMP scan would miss all the hosts which are alive but where the firewall is dropping the ICMP echo request packet. However, if you want to find more hosts, it would be advisable to separate the list of IPs which responded to ICMP from the IP address scan range and run the scan again (may be) with a SYN to 443 and an ACK to 80 using PA, PS options.

Please also note Nmap’s ICMP ping, by default, sends zero data as part of the ping. Nmap typically pings the host via ICMP if the user has root privileges, and uses a tcp-ping otherwise. This is easily detected by Snort’s IDS Rule 1-469 SID 1-469.

This could be evaded by using

```
--data <hex string> (Append custom binary data to sent packets)
--data-string <string> (Append custom string to sent packets)
--data-length <number> (Append random data to sent packets)
```

Please note that you should use these options only on ICMP Echo Request for IDS Evasion as the data gets appended to every packet (ex. port scan packets). Designing the ideal combinations of probes as suggested in the Nmap Book is

```
-PE -PA -PS 21,22,23,25,80,113,31339 -PA 80,113,443,10042
 Adding --source-port 53 might also help
```

The above combination would find more hosts than just the ping scan, however it also gonna cost a decent amount of time. Normal Time vs. Accuracy trade off.

-PE, -PA and -PS send respectively a ICMP request, TCP ACK, and TCP SYN probe.

Todo

replace section above with nping after comparing results?

Todo

https://linux.die.net/man/1/nping

Todo

using -PA twice seems odd

Port Scanning

Once you have the list of IP Addresses which are alive, we can perform portscanning on them. Nmap provides multiple options such as

```
-sS TCP SYN Stealth : Half Open SYN Scan : Nmap sends the SYN packet, Server w
-sT TCP Connect Scan : Nmap uses system to send the SYN scan : Connect full TC
```

```
-sU UDP Scan
-sA ACK Scan : Ack scan is generally used to map out firewall rule sets. (Whet
```

Please note p0f recognizes Nmap's SYN scan because of the TCP Options such as the TCP window size which is a multiple of 1024, and only the MSS option supported with a value of 1460 (Check the tcpdump output of Ping scan above, SYN Packet). Recently, a IRC user was getting filtered port while using SYN Scan whereas he was getting OPEN ports when using telnet or TCP Connect Scan. Also, A patch to allow a user to override the TCP Window size in SYN scan was posted to the Nmap Development List.

By default, nmap scans the 1000 most popular ports of each protocol (gathered by scanning million of IP address). Scanning 1000 ports in an unknown environment with 16 million IP Address could be challenging. Nmap also provides a Fast scan (-F) option which scans the 100 most common ports for each protocol. Otherwise it also provides –top-ports to specify an arbitrary number of ports. So, How do we know what are the ports scanned with –top-ports option? This could be figured out for respectively the top 1000 and top 100 ports by running the following commands:

```
nmap -sT -oG - -v | grep '^# Ports'
```

or

```
nmap localhost -F -oX - | grep '^<scaninfo'
```

Nmap needs an nmap-services file with frequency information in order to know which ports are the most common. See the section called Well Known Port List: nmap-services : for more information about port frequencies. We could provide ports to nmap by using -p option also, for example

```
-p 22 : Scan single port
-p 22,25,80 : Scan multiple ports with comma separated values. If -sS is speci
-p 80-85, 443, 8000-8005 : Scan port with ranges.
-p- : Scan all the ports excluding 0.
-pT:21,22,25,U:53,111,161 : Scan TCP 21,22,25 and UDP Ports 53,111,161. -sU mu
-p http* : wild cards may be used for ports with similar names. This would mat
```

Port scanning via **netcat**: Netcat might not be the best tool to use for port scanning, but can be used quickly. netcat scans TCP ports by default, but we can perform UDP scans as well.

For a TCP scan, the format is

```
nc -vvn -z xxx.xxx.xxx.xxx startport-endport
   -z flag is Zero-I/O mode (used for scanning)
   -vv will provide verbose information about the results
   -n flag allows to skip the DNS lookup
```

For a UDP Port Scan, we need to add -u flag which makes the format

```
nc -vvn -u -z xxx.xxx.xxx.xxx startport-endport
```

If we have windows machine without nmap, we can use PSnmap

Identifying service versions

Ideally, we can use -sV to probe the ports to find the running version of a service. When performing a version scan (-sV), Nmap sends a series of probes, each of which is assigned a rarity value in regards to correctly identifying a service. However, high intensity scans or in other words sending many probes takes longer. The intensity must be between 0 and 9. The default is 7.

Ideally, to avoid IDS Detection, we should avoid using the -sV option. However, we can keep the noise less by using –version intensity by which we can control the number of probes sent to determine the service. Setting this option to 0 will send only the Null probe (connect and wait for banner) and any probes that have been specifically listed as pertaining to the scanned port in nmap-service-probes. The other available options are provided below:

```
--version-light (Enable light mode) : Alias for --version-intensity 2.
--version-all (Try every single probe) : An alias for --version-intensity 9
--version-trace (Trace version scan activity) : Print debugging information.
```

Also, when -sV is specified apart from the probes, all the scripts in the Version category are executed. These scripts could be prevented from running by removing them from the script.db catalog or by building Nmap without NSE support (./configure –without-liblua). However, if –version-intensity option is less than 7, those scripts won't be executed (I might be a little wrong here).

So our scan would become approx

```
nmap <IP_Address_Range> -n --top-ports <number>/-p <Custom Port List> -sV --ve
```

Performance

So, how can we improve the performance of our nmap scan, so that result could be achieved faster. As always we will have Time Vs Accuracy Trade off.

```
-T<0-5>: Set timing template (higher is faster)
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies probe
--max-retries <tries>: Caps number of port scan probe retransmissions.
--host-timeout <time>: Give up on target after this long
--scan-delay/--max-scan-delay <time>: Adjust delay between probes
--min-rate <number>: Send packets no slower than <number> per second
--max-rate <number>: Send packets no faster than <number> per second
```

T0, T1, T2 is specifically for IDS Evasion. T3 is the default. We can set max-retries to a lower value such as 2. Currently it's 10 for T0, T1, T2, T3; 6 for T4 and 2 for T5.

Nmap Scripts

As bonsaiviking says in They See Me Scanning Part 2 If you are wild enough to try NSE scripts against an IDS-protected target, you should know how to read Lua, since the script sources are the final authority on what data is sent. But if you're just looking to get a little better at blending in, these tips should help:

- Use –script-args-file to pass script arguments to Nmap from a file. This will keep your command line clean and make it harder to accidentally miss one of the options you choose
- Obviously avoid dos, intrusive, and exploit category scripts.
- Use scripts by name instead of by category, so that you know exactly what will be run.
- Thoroughly read the documentation for each script you intend to use. Set http.useragent to something believable that blends in. Currently, the HTTP scripts all use a User-Agent header that identifies as “Nmap Scripting Engine.”

Note

Nmap scripts are stored in /usr/share/nmap/scripts

Output Options

```
-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3, and Grepab
-oA <basename>: Output in the three major formats at once
--reason: Display the reason a port is in a particular state
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--resume <filename>: Resume an aborted scan: Filename should be .nmap or .gnma
```

At this point, it’s good to find what are the most common ports open in the scan we just performed by

```
grep "^[0-9]\+" <nmap file .nmap extension> | grep "\ open\ " | sort | uniq -c
```

or

```
xmlstarlet sel -t -m '//port/state[@state="open"]/parent::port' -v 'ancestor::
```

Exploring the Network Further

By now, we would have information about what ports are open and possibly what services are running on them. Further, we need to explore the various options by which we can get more information.

Gathering Screenshots for http* services

There are four ways (in my knowledge to do this):

- **http-screenshot NSE**: Nmap has a NSE script http-screenshot This could be executed while running nmap. It uses the wkhtml2image tool. Sometimes, you may find that running this script takes a long time. It might be a good idea to gather the http* running IP, Port and provide this information to wkhtml2image directly via scripting. You do have to install wkhtml2image and test with javascript disabled and other available options.
- **httpscreenshot** from breenmachine: httpscreenshot is a tool for grabbing screenshots and HTML of large numbers of websites. The goal is for it to be both thorough and fast which can sometimes oppose each other.
- **Eyewitness** from Chris Truncer: EyeWitness is designed to take screenshots of websites, provide some server header info, and identify default credentials if possible.
- Another method is to use html2image which is a simple Java library which converts plain HTML markup to an image and provides client-side image-maps using html element.
- **RAWR: Rapid Assessment of Web Resources**: RAWR provides with a customizable CSV containing ordered information gathered for each host, with a field for making notes/etc.; An elegant, searchable, JQuery-driven HTML report that shows screenshots, diagrams, and other information. A report on relevant security headers. In short, it provides a landscape of your webapplications. It takes input from multiple formats such as Nmap, Nessus, OpenVAS etc.

Information Gathering for http* Services

- WhatWeb recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded device. Tellmeweb is a ruby script which reads a Nmap Gnmap file and runs whatweb against all identified open http and https ports. A WhatWeb Result Parser has also been written which converts the results to CSV format. More information about advanced usage can be found at Whatweb Advance Usage.

- Wapplyzer is a Firefox plug-in. There are four ways (in my knowledge to do this) be loaded on browser. It works completely at the browser level and gives results in the form of icons.
- W3Tech is another Chrome plug-in which provides information about the usage of various types technologies on the web. It tells which web technologies are being used based on the crawling it has done. So example.com, x1.example.com, x2.example.com will show the same technologies as the domain is same (which is not correct).
- ChromeSnifferPlus is another chrome extension which identifies the different web-technologies used by a website.
- BuiltWith is another website which provides a good amount of information about the different technologies used by website.

NetBIOS Service

Netbios listens on TCP Port 139, 445 and UDP Port 137. We can use grep to identify machines we ran nmap against earlier on which these three ports or a combination of them are open. The idea is the filter those IP’s out and feed them to nbtscan and/or enum4linux

```
:Grep for UDP port 137 which is required for nbtscan
grep -E "^Host.*[ ]137/open/udp" <Nmap .gnmap file>
grep -E "^Host.*[ ]139/open/tcp" <Nmap .gnmap file>

:Grep for TCP 139 and 445 to be able to run enum4linux
grep -E "^Host.*[ ]139/open/tcp" <Nmap .gnmap file> | grep -E "^Host.*[ ]445/c

.. Todo:: validate which ports are required for enum4linux
.. Todo:: in previous example, grep TCP 135 and 445 port were mentioned as pre

#If we want that tcp port 139 or 445 must be open
grep -E "^Host.*[ ]139/open/tcp|[ ]443/open/tcp" <Nmap .gnmap file>
```

NBTSCAN

```
nbtscan
    -v              Verbose output. Print all names received from each host.
    -f filename     Take IP addresses to scan from file "filename"
```

enum4linux

A Linux alternative to enum.exe to enumerate data from Windows and Samba hosts. It is basically a wrapper around the Samba tools smbclient, rpclient, net and nmblookup. A very good usage guide is enum4linux

SNMP Enumeration

For SNMP Enumeration, UDP Port 161 should be open. To gather more information we can use:

- **snmpcheck:**

```
snmpcheck -t <IP address>
     -c : SNMP community; default is public
     -v : SNMP version (1,2); default is 1
     -w : detect write access (separate action by enumeration)
```

- **snmpwalk:**

Also allows us to interact with the SNMP version 3 and extract particular nodes of a MIB tree.

```
snmpwalk -c public -v1 <IP Address>  : Enumerates the Entire MIB Tre
snmpwalk -c public -v1 <IP Address>  <MIB Tree Number> : Enumerates
    -v 1|2c|3     specifies SNMP version to use
    -c COMMUNITY  set the community string
```

- **OneSixtyOne:**

Onesixtyone allows you to brute force the community strings

Todo

give example?

Attack Surface Area - Reconnaissance Tools

Aquatone: A tool for domain flyovers

Aquatone is a set of tools for performing reconnaissance on domain names. It can discover subdomains on a given domain by using open sources as well as the more common subdomain dictionary brute force approach. After subdomain(s) discovery, AQUATONE can scan the identified hosts (subdomains) for common web ports and HTTP headers, HTML bodies and screenshots can be gathered and consolidated into a report for easy analysis of the attack surface. A detailed blog is available at AQUATONE: A tool for domain flyovers

Todo

move the earlier mention recon-ng, dnsenum, dnsrecon section?

Todo

provide an example?

DataSploit

The Datasploit tool performs various OSINT techniques, aggregates all the raw data, and returns the gathered data in multiple formats.

Functional Overview:

- Performs OSINT on a domain / email / username / phone and find out information from different sources.
- Correlates and collaborate the results, shows them in a consolidated manner.
- Tries to figure out credentials, api-keys, tokens, subdomains, domain history, legacy portals, etc. related to the target.

- Use specific script/ launch automated OSINT to consolidate data.
- Performs Active Scans on collected data.
- Generates HTML, JSON reports along with text files.

Spiderfoot

SpiderFoot is an open source intelligence automation tool. Its goal is to automate the process of gathering intelligence about a given target, which may be an IP address, domain name, hostname or network subnet. SpiderFoot can be used offensively, i.e. as part of a black-box penetration test to gather information about the target or defensively to identify what information your organization is freely providing for attackers to use against you.

Todo

add example?

Intrigue.io

Intrigue makes it easy to discover information about the attack surface connected to the Internet. Intrigue utilizes common OSINT sources via "tasks" to create "entities". Each discovered entity can be used to discover more information, either automatically or manually.

Todo

to the Internet > about the attacker surface of a given domain/host connected to the Internet?

Todo

demo?

Appendix-I : Interesting Stories

Initial Compromise

- Apache and Java Information Disclosures Lead to Shells : Richard De La Cruz talks about a recent Red-Team engagement, where a series of information disclosures were discovered on a site allowing the team to go from zero access to full compromise in a matter of hours.

Summary:

- Information disclosures in Apache HTTP servers with mod_status enabled allowed our team to discover .jar files, hosted on the site.
- Static values within the exposed .jar files allowed our team to extract the client's code signing certificate and sign malicious Java executables as the client.
- These malicious .jar files were used in a successful social engineering campaign against the client.

Todo

A lot of tools are being mentioned in this blog article. It might therefore be interesting to give a run down. Which might be highly situationals

Changelog

- **Added Open-Ports via XML** by *bitvijays*

Be the first to comment.

ALSO ON **TECH.BITVIJAYS.COM**

Infrastruture PenTest Series : Part 3 - Exploitation¶

1 comment • 2 years ago

Avatar **Anthony Esdaile** — Amazing work what a great contribution to the community i was wondering if there was anything comparable

Infrastruture PenTest Series : Part 2 - Vulnerability Analysis¶

1 comment • 2 years ago

Avatar **Maina Mathenge** — very informative

Feedback

2 comments • 2 years ago

Avatar **Rowan Sheridan** — this sums up my thoughts too

Subscribe   Add Disqus to your site   Disqus' Privacy Policy   DISQUS

tech.bitvijays.com »