Network and Services Discovery

A quick theorical introduction to network scanning

Alexandre Dulaunoy

http://www.foo.be/

January 8, 2016

Disclaimer/Intro

Network scanning is not exact science

- *When an information system is able to interact over the network :*
  - *The system is always giving some information about himself*
  - *The system may be used to act as a network scanning tool*
  - *Attackers are always looking for such kind of services*
- *Network scanning is playing an important role in the process of network discovery for end-user but also for the potential attackers.*

IP protocol scanning

How to know the IP protocol supported by a system. Not only limited to UDP or TCP. The approach used is quite simple :

-

We send a raw IP packet with the protocol defined (RFC3232) and wait for the reply :

- if received an ICMP protocol unreachable, the protocol is not available on the stack
- if we got nothing, the procotol is available or ICMP packets are filtered by a firewall

e.g. : Useful for testing the protocol available on a router. Is IGMP available ?

ICMP scanning

- Nifty Internet Control Message Protocol :

```
ICMP type 8 - echo request
ICMP type 13 - time stamp request
ICMP type 15 - information request
ICMP type 17 - netmask request
```

  ICMP can be used as very basic discovery tool. Not always filtered as some of them are required by RFCs and/or for proper operation.

TCP Port Scanning - Connect Scan

A connect scan is a scan using the vallina connect() approach. The full TCP 3-way handshake is done :

-

We send a SYN to the target host and port:

- We wait for the SYN-ACK from the target host
- if yes, we'll send an ACK to the target. This means that port is open.
- If we got a RST, the target host is not listening on that port.

The connect() scan is very slow, very visible (e.g. TCP Wrappers, netstat) due to the full handshake (ESTABLISHED state).

TCP Port Scanning - Half-open scan

A half-open scan is where the full TCP handshake is not done :

-

We send a SYN to the target host and port :

- We wait for the SYN-ACK from the target host
- if yes, we'll send directly a RST to the target. This means that port is open but we don't continue the handshake.
- If we got a RST, the target host is not listening on that port.

The half-open scan is less visible than the connect() scan. This scan is always faster than the connect() scan too. The main issue is the use of the SYN flag...

TCP Port Scanning - Inversed or stealh TCP scan

By default a TCP stack must respond (RFC793) to non-SYN flag on a closed port with a RST. For open port, the datagram must be discarded. But various systems are still sending RST on open ports too...

-

We send an FIN or FIN/URG/PUSH or no flag to the target host and port :

- We wait for the RST from the target host
- if yes, this means that port is closed.
- If we got nothing, this is probably an open port.

Results are not always reliable but can be used in conjunction with other scan. The main advantage is not to use SYN flag (e.g. packet filter only checking for SYN).

TCP Port Scanning - ACK TCP scan

An ACK scan is often used to check the kind of packet filter between you and the target host.

-

We send an ACK (random ack/seq) to the target host and port :

- We wait for the RST from the target host
  - Variation of the TTL and Window field is also interesting.
- if yes, this means that port is unfiltered.
- If we got nothing, this is probably a filtered port.

Hiding your active scanning

If you are an attacker, it's always better to find ways to 'somewhat' hide your various activities :

- using FTP relay (old ftp server) - RFC959
- using open proxies (e.g. CONNECT method)
- Idle scan approach
  - Using a zombie host to receive your request
  - Cover channel using IPID (IP identification) to receive the reply on open port from the zombie

TCP passive fingerprint

Signatures are built on various TCP/IP parameters :

- TTL (max. ttl is different following the OS/version)
- Window Size
- DF bit set (some OS are sending it or not)
- ToS
- ECN, Selective Acknowledgement, ...

You can build database of remote services including Operating System version, revision...(check tools presentation)

Passive discovery

Another way to discover network services is to use an alternative tools that is gathering the information for you. For example, web crawler can do the job for you and they provide nice interface for an exhaustive search :

- intext:password | intext:login | user filetype:csv
- inurl:"ViewerFrame?Mode=refresh"
- intitle:"index of"
- "Microsoft-IIS/" intitle:"index of"

Network Scanning

- How to scan an infrastructure?
- What are the ethical and legal requirements?
- How often do you need to scan an infrastructure?
- What are the impact of scanning?

Q and A

- Thanks for listening.
- http://www.foo.be/
- a@foo.be