HideZeroOne
INE – Cyber Sec
www.hide01.ir

Incident Handling & Response Professional

Preparing & Defending Against Scanning

Section 03 | Module 02

OUTLINE

Table of Contents

Module 02 | Preparing & Defending Against Scanning

2.1 Scanning: Definition

2.2 Scanning Techniques & Defense

OUTLINE

Learning Objectives

By the end of this module, you should have a better understanding of:

- ✓ The scanning techniques used by attackers
- ✓ How to prepare and defend against scanning activities

OUTLINE

2.1

Scanning: Definition

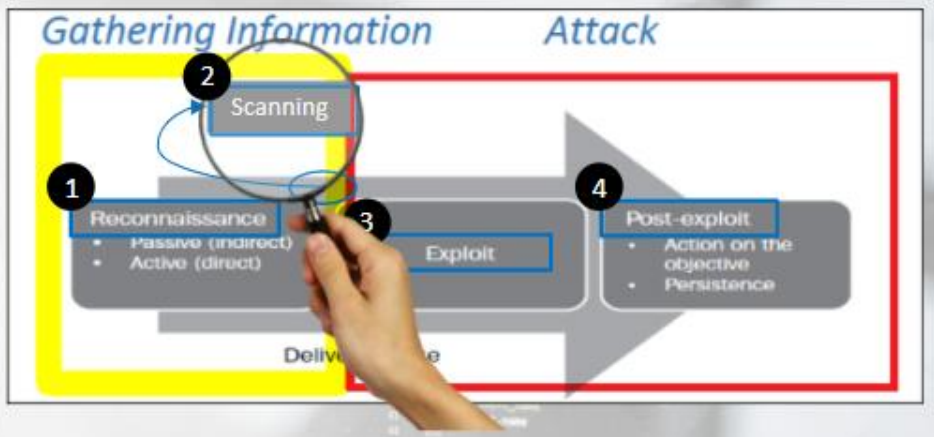
OUTLINE

2.1 Scanning: Definition

Once attackers complete their reconnaissance activities, their efforts are concentrated on identifying openings in the target network.

The process of sending transmissions to end nodes and analyzing the responses in order to identify information about the target system is known as **active scanning**.

Passive scanning is also possible by sniffing and analyzing network traffic in order to identify what a target system is running, users, infrastructure and vulnerabilities.

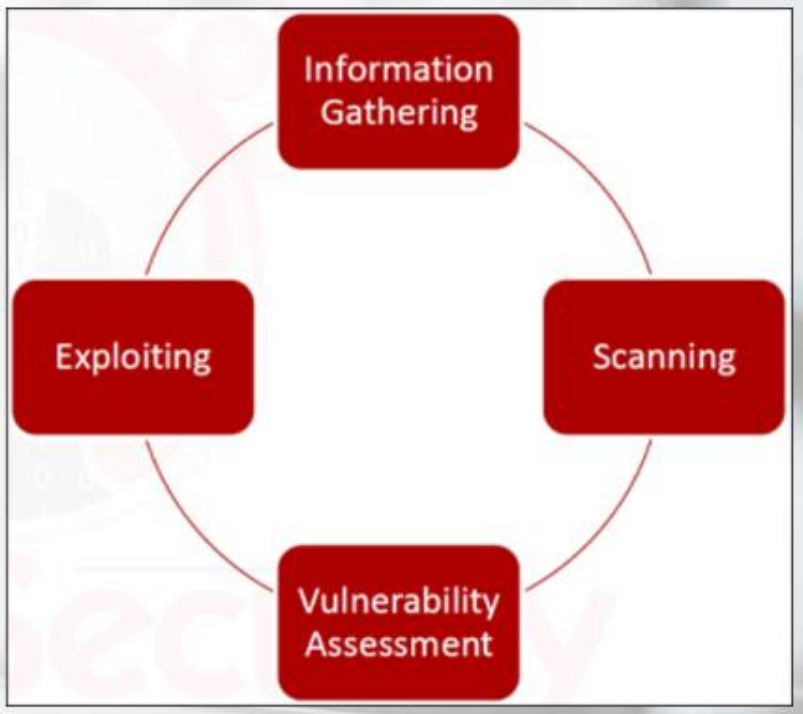


OUTLINE

2.1 Scanning: Definition

An attack (or a penetration test) is a cyclic process.

This means that once attackers gain initial foothold, they will perform information gathering and scanning activities once again, in order to gain a better understanding of the network and identify additional weak spots.



OUTLINE

2.2

Scanning Techniques & Defense

OUTLINE

2.2 Scanning Techniques & Defense

Let's cover the scanning activities that can be performed by attackers. Specifically, we'll cover the following scanning techniques, as well as how to defend against them:

- War Dialing
- War Driving
- Nmap/Masscan/Nessus Scans
- WebRTC-based Scans

2.2.1 War Dialing

Attackers will always search for weak spots in an organization's perimeter. Among the first things they will attempt is try to identify available modems. Once modems are identified they are then inspected for insecure configurations.

As you can imagine if a modem is insufficiently secured it can grant attackers access to a corporate system and/or network.

2.2.1 War Dialing

The process of auto-dialing phone blocks in an attempt to identify modems and then checking for insecure configurations in any identified one is known as **War Dialing**.

2.2.1 War Dialing

Attackers may also focus on a single number and try to brute force passwords through a demon dialer.

As we have already covered in the previous module an organization's phone block(s) can be easily identified though numerous open sources or social engineering.

2.2.1 War Dialing

Since having access to an old-school analog line is not always possible, attackers are nowadays performing war dialing activities though the internet and VoIP accounts.

A tool that can facilitate war dialing assessments is WarVox.

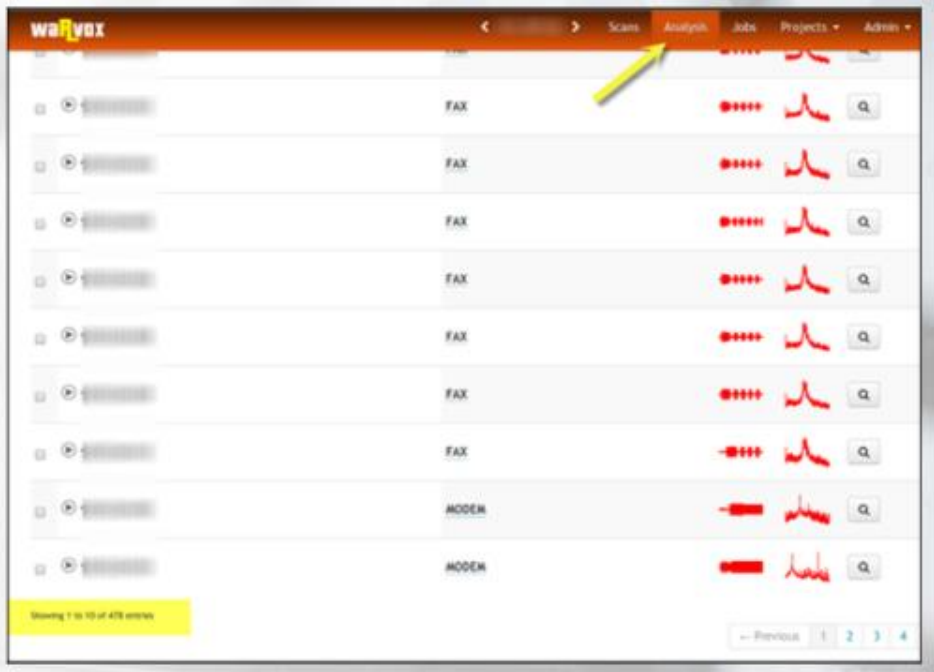
https://github.com/rapid7/warvox

2.2.1 War Dialing

WarVox performs war dialing through VoIP accounts and is also capable of caller ID spoofing. Note that the VoIP service provider should support the Inter-Asterisk eXchange (IAX) protocol for WarVox to function properly.

For every number dialed, WarVox creates an MP3 audio file which contains the answer/communication.

WarVox provides signatures that are applied against these MP3 files in order to identify if a modem, voicemail box, fax or human answered the call.



OUTLINE

2.2.1 War Dialing

War dialer logs contain critical info such as login prompts or banners. Attackers leverage such information to perform password guessing attacks against identified modems.

OUTLINE

2.2.1 War Dialing

Preparation & Defense

You can prepare against war dialing attacks by:

1. Proactively performing a war dialing assessment against your own network. WarVox is not the only option. Commercial war dialers also exist like Niksun's PhoneSweep.
2. Scrutinizing your organization's facilities for unwanted or not inventoried modems.

https://www.niksun.com/product.php?id=17

2.2.1 War Dialing

Preparation & Defense

You can defend against war dialing attacks by:

1. Enabling the scanning identification functionality of your PBX (if it has any)
2. Setting up a PBX Firewall or IPS
3. Applying the following PBX security tips https://cdt.ca.gov/wp-content/uploads/2017/05/PBX-SECURITY-ITS-YOUR-BUSINESS.pdf

2.2.2 War Driving

Insufficiently secured modems are not the only weak spot of an organization's perimeter. Insufficiently secured or even unsecured wireless LANs can also pose as the perfect way into an organization's network.

2.2.2 War Driving

Attackers are known for driving/passing by an organization's facilities and physically searching for insufficiently secured or unsecured wireless LANs. What they actually come up with is the organization's access point map.

The process above is known as **War Driving**.

2.2.2 War Driving

Attackers start by constantly sending numerous probe requests without a specified SSID, hoping that an AP will respond with a probe response that includes its SSID.

They do so by using tools such as InSSIDer.

As you can see on your right InSSIDer gathers information such as the SSID, the MAC address, the channel, the type of encryption in use etc.

https://www.metageek.com/products/inssider/

2.2.2 War Driving

That being said, an AP can be configured to not respond to probe requests without a specified SSID (another aspect of SSID cloaking). For such cases attackers use different tools such as Kismet.

Kismet is a powerful tool that can even passively discover APs without any beacon being sent in the process. It is essentially a sniffing tool looking for SSIDs within the sniffed traffic.

So, even if SSID cloaking is configured, all it takes is one user who sends traffic over the wireless LAN for Kismet to identify the AP's SSID.

https://www.kismetwireless.net/

2.2.2 War Driving

Attackers will also attempt to crack an AP's security by using tools such as aircrack, wepcrack and asleap.

The first two leverage flaws of the WEP algorithm to crack keys. All the attacker has to do is sniff traffic for about half an hour and then the WEP key can be cracked. A cracked WEP key means that the attacker can view all data crossing the LAN.

The third tool essentially attacks a user's Windows password hash after sniffing LEAP challenge and response messages. It should be noted that the attack is dictionary-based. After a successful attack the attacker will be able to join a LEAP-protected wireless LAN.

https://www.aircrack-ng.org/
https://sourceforge.net/projects/wepcrack/
http://www.willhackforsushi.com/?page_id=41

OUTLINE

2.2.2 War Driving

WEP and LEAP are considered weak wireless security protocols, this doesn't mean that attackers will not try to crack their stronger counterparts WPA1 and WPA2.

Attacks against WPA1 & WPA2 can be mounted through the coWPAtty tool.

coWPAtty requires a sniffed WPA1 or WPA2 four-way handshake used for authentication and prior knowledge of the SSID.

More information can be found on the following resource: http://www.ciscopress.com/articles/article.asp?p=370636

```
thallium cowpatty $ capinfos htc01-short.dump
File name: htc01-short.dump
File type: Wireshark/tcpdump/... - libpcap
File encapsulation: IEEE 802.11 plus radiotap WLAN header
Number of packets: 2
File size: 392 bytes
Data size: 336 bytes
Capture duration: 0.220341 seconds
Start time: Mon May 18 06:46:31 2009
End time: Mon May 18 06:46:31 2009
Data rate: 1524.91 bytes/s
Data rate: 12199.26 bits/s
Average packet size: 168.00 bytes
thallium cowpatty $ ./cowpatty -r htc01-short.dump -s dlink -f ~/dict/openwall
cowpatty 4.5 - WPA-PSK dictionary attack. <jwright@hasborg.com>

End of pcap capture file, incomplete four-way handshake exchange.  Try using a
different capture.
thallium cowpatty $ ./cowpatty -r htc01-short.dump -s dlink -f ~/dict/openwall -2
cowpatty 4.5 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against WPA/PSK passphrase.
Starting dictionary attack.  Please be patient.

The PSK is "12345678".

101 passphrases tested in 2.69 seconds:  37.55 passphrases/second
```

Source: willhackforsushi.com

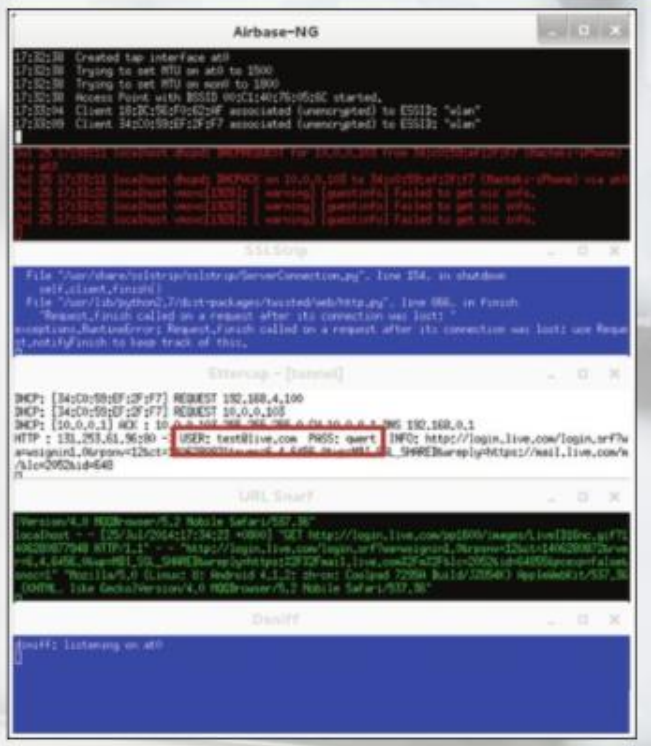
https://sourceforge.net/projects/cowpatty/
http://www.ciscopress.com/articles/article.asp?p=370636

2.2.2 War Driving

To conclude wireless attacks, attackers will most probably also try to create an open access point, that is, of course, reachable by users of the targeted organization.

Regardless of how secure a device joining an open AP may be, its traffic will be visible by the attacker, since he/she controls the medium.

There are numerous tools that facilitate the creation of rogue access points and the sniffing of sensitive data, once the victim joins in. You can see such a tool on your right. The tool depicted is easy-creds.



https://github.com/brav0hax/easy-creds

2.2.2 War Driving

Preparation & Defense

When it comes to preparing and defending against wireless LAN attacks, first thing to note is that the services set identifier (SSID) is by no means an aspect of a wireless LAN's security. SSID just acts as a wireless LAN's name and is included by default in every broadcast beacon packet.

That being said, it is better to proceed to SSID cloaking so that the SSID of an access point is not included in its beacons. Kismet-powered attacks will still succeed in identifying the SSID though.

2.2.2 War Driving

Preparation & Defense

To prepare against wireless LAN attacks, you can:

- Choose generic SSIDs to avoid been targeted
- Enforce strong wireless security protocols like WAP1 and WPA2 and prefer using AES in WPA2
- Consider securing a wireless LAN using a Virtual Private Network (VPN)
    - If you do so, remember that attackers can still sniff traffic and will try to crack any passing pre-shared IKE keys. This is why, you need to make sure that aggressive mode IKE is disabled.
- Consider deploying a Wireless IDS that can detect rogue access points as well as repeated probes and high volumes of de-authentication messages.

2.2.3 Nmap/Masscan/Nessus Scans

Scanning a whole network is a demanding task, for this reason attackers prefer using scanners like Nmap and Masscan and vulnerability scanners like Nessus to automate a big portion of their work.

We have already covered how to detect various scanning activities, using traffic analysis or flow analysis techniques.

2.2.3 Nmap/Masscan/Nessus Scans

Preparation & Defense

A good strategy to prepare against scanning is to proactively scan your whole network, disable any unwanted services that are listening for incoming connections, configure firewall rules to block/filter specific incoming traffic and patch any system that the vulnerability scanner marked as vulnerable.

OUTLINE

2.2.3 Nmap/Masscan/Nessus Scans

Preparation & Defense

As far as detection is concerned, what you need to remember is that scanning cannot always be detected based on the volume of interactions with a specific machine.

Attackers are known for using multiple compromised machines or open proxies to conceal their scanning activities behind them.

2.2.3 Nmap/Masscan/Nessus Scans

Preparation & Defense

A better strategy would be configuring and fine-tuning IDS systems so that you are informed of any scanning activity.

Find below a resource explaining Nmap's port scanning techniques and a resource on how to detect Nmap scans, to get you warmed up.

- https://nmap.org/book/scan-methods.html
- https://nmap.org/book/defenses.html

2.2.3 Nmap/Masscan/Nessus Scans

Preparation & Defense

Finally, if you want to protect Linux services from scanners, you can employ a TCP Wrapper. TCP Wrappers give the administrator the flexibility to permit or deny access to the services based upon IP addresses or domain names.

Note that, TCP Wrappers always allow the protected service(s) to be advertised. When scanned, the system will list the service(s) as being open, but when the attacker tries to exploit the open port, TCP Wrappers will reject the incoming connection if it is not originated from an approved host or domain.

http://www.admin-magazine.com/Articles/Secure-Your-Server-with-TCP-Wrappers

OUTLINE

2.2.4 WebRTC-based Scans

WebRTC, according to its website, is a free, open project that provides browsers and mobile applications with Real-Time Communications (RTC) capabilities via simple APIs.

https://webrtc.org/

OUTLINE

2.2.4 WebRTC-based Scans

Firefox and Chrome have implemented WebRTC that allows requests to STUN servers to be made that will return the local and public IP addresses for the user.

These request results are available to JavaScript, so attackers can obtain users local and public IP addresses by simply luring them into visiting a website containing specifically crafted JavaScript code.

https://temasys.io/webrtc-ice-sorcery/

OUTLINE

2.2.4 WebRTC-based Scans

You can find JavaScript code that abuses WebRTC to obtain a user's internal IP in the following resource.

- https://github.com/beefproject/beef/wiki/Module%3A-Get-Internal-IP-WebRTC

OUTLINE

2.2.4 WebRTC-based Scans

This WebRTC “capability” poses as a threat to anonymity, but attackers haven’t stopped there.

They have combined WebRTC with XHR requests (or other cross-origin interactions) to scan a user’s LAN from inside a malicious page. Find below two implementations of such an attack.

- https://blog.beefproject.com/2016/06/mapping-your-lan-from-web-browser.html
- https://portswigger.net/blog/exposing-intranets-with-reliable-browser-based-port-scanning

https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest

2.2.4 WebRTC-based Scans

Preparation & Defense

WebRTC in Mozilla Firefox is supported since Firefox 22, and it's enabled by default.

To disable `RTCPeerConnection` and protect IP addresses leakage, go to `about:config` and toggle `media.peerconnection.enabled` to `false`.

WebRTC in Google Chrome and Chromium-based web browsers is supported and enabled by default since Chrome version 23. To protect IP addresses from leaking, you can use the official webrtc.org extension WebRTC Network Limiter.

https://chrome.google.com/webstore/detail/webrtc-network-limiter/npeicpdbkakmehahjeeohfdhnlpdklia

References

OUTLINE

References

WarVox

https://github.com/rapid7/warvox

PhoneSweep

https://www.niksun.com/product.php?id=17

PBX Security Tips

https://cdt.ca.gov/wp-content/uploads/2017/05/PBX-SECURITY-ITS-YOUR-BUSINESS.pdf

InSSIDer

https://www.metageek.com/products/inssider/

References

Kismet

https://www.kismetwireless.net/

aircrack

https://www.aircrack-ng.org/

wepcrack

https://sourceforge.net/projects/wepcrack/

asleap

http://www.willhackforsushi.com/?page_id=41

References

coWPAtty

https://sourceforge.net/projects/cowpatty/

Cracking Wi-Fi Protected Access (WPA), Part 2

http://www.ciscopress.com/articles/article.asp?p=370636

easy-creds

https://github.com/brav0hax/easy-creds

Nmap Network Scanning: Chapter 5. Port Scanning Techniques and Algorithms

https://nmap.org/book/scan-methods.html

References

Nmap Network Scanning: Chapter 11. Defenses Against Nmap

https://nmap.org/book/defenses.html

TCP Wrapper

http://www.admin-magazine.com/Articles/Secure-Your-Server-with-TCP-Wrappers

WebRTC

https://webrtc.org/

ICE and WebRTC: What Is This Sorcery? We Explain...

https://temasys.io/webrtc-ice-sorcery/

References

Beefproject / beef – Module: Get Internal IP WebRTC

https://github.com/beefproject/beef/wiki/Module%3A-Get-Internal-IP-WebRTC

XHR

https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest

Mapping your LAN from a web browser: Introducing the Network extension for BeEF

https://blog.beefproject.com/2016/06/mapping-your-lan-from-web-browser.html

Exposing Intranets with reliable Browser-based Port scanning

https://portswigger.net/blog/exposing-intranets-with-reliable-browser-based-port-scanning

References

WebRTC Network Limiter

https://chrome.google.com/webstore/detail/webrtc-network-limiter/npeicpdbkakmehahjeeohfdhnlpdklia