ADVANCED WIRELESS ATTACKS AGAINST ENTERPRISE NETWORKS

@s0lst1c3 @gdssecurity

net user author /domain

Gabriel Ryan

Security Engineer @ Gotham Digital Science

Appsec | Penetration Testing | Red Team | Research

@s0lst1c3

gryan@gdssecurity.com

labs@gdssecurity.com

Target Identification Within A Red Team Environment

Target Identification Within A Red Team Environment

Well executed wireless attacks require well executed recon

Target Identification Within A Red Team Environment

Typical wireless assessment:

- Tightly defined scope
- Client provides set of in-scope ESSIDs
- You may be limited to certain access points

Target Identification Within A Red Team Environment

Red team engagements:

- Scope loosely defined
- Scenario based: demonstrated that a scenario is possible (i.e. CREST)
- Open: completely compromise infrastructure of target organization
- Loose rules dictating what you can and can’t do

Target Identification Within A Red Team Environment

Typical red team rules of engagement make wireless target identification very messy.

Target Identification Within A Red Team Environment: Example Scenario

Target Identification Within A Red Team Environment

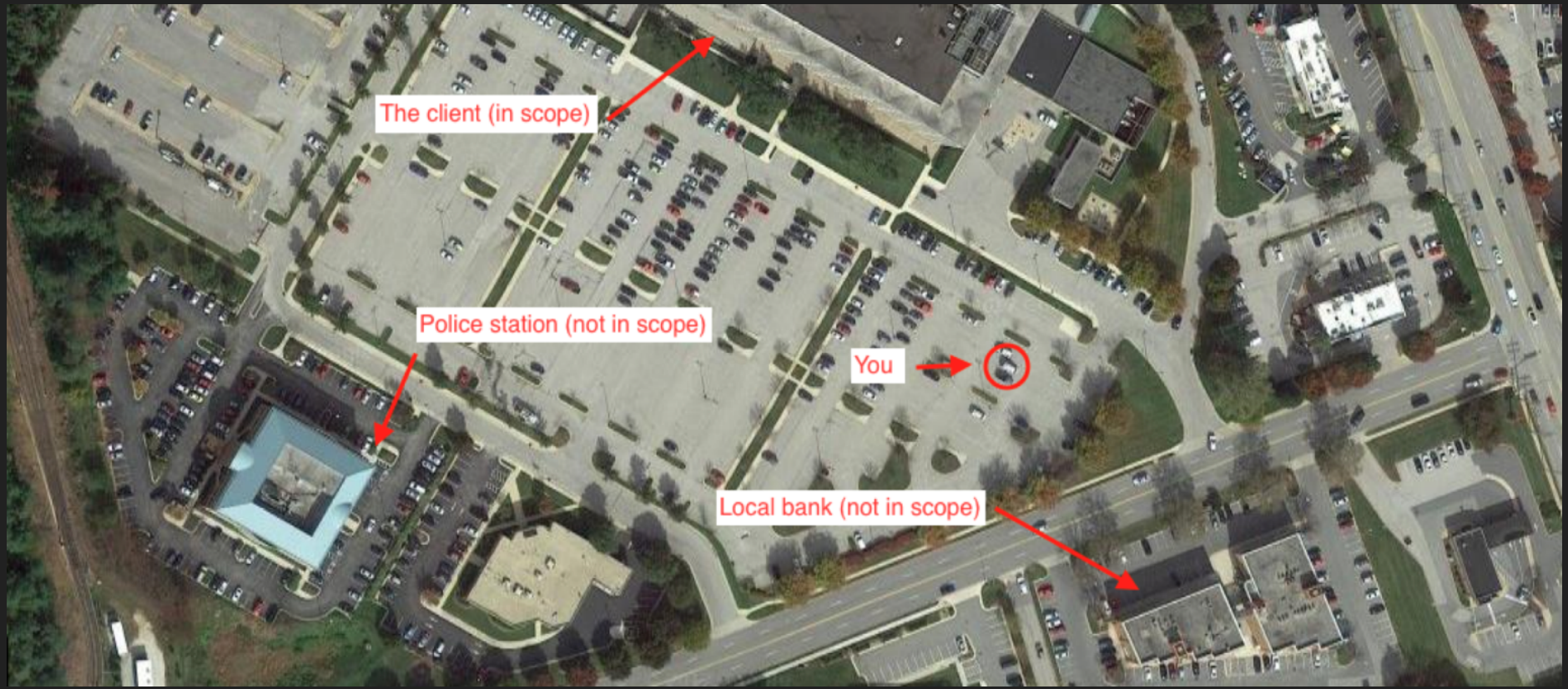
Evil Corp has requested that your firm perform a full scope red team assessment of their infrastructure over a five week period.

- 57 offices across US and Europe
- Most locations have one or more wireless networks
- Evil Corp particularly interested in wireless as attack vector

Target Identification Within A Red Team Environment

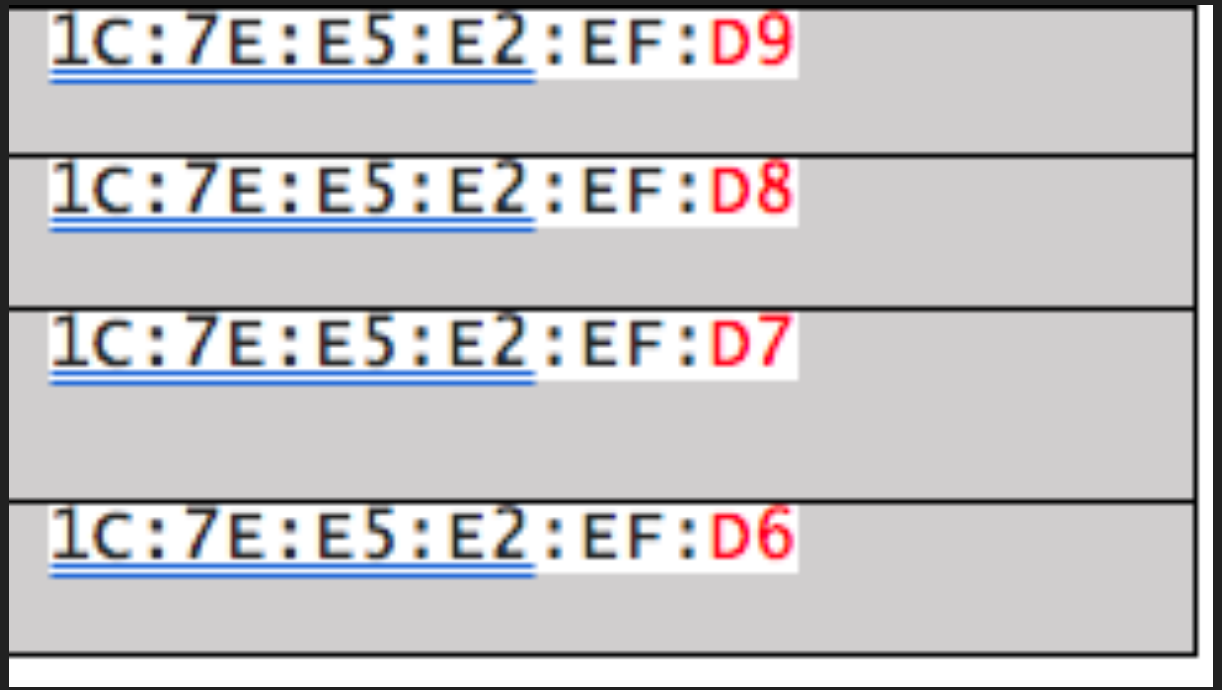
Arrive at client site, perform site-survey

#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
283	0	1	54	WPA2	CCMP	MGT	
83	1	1	54	WPA2	CCMP	MGT	
273	0	1	54	WPA2	CCMP	MGT	
43	0	1	54	WPA2	CCMP	MGT	
24	0	6	54	WPA2	CCMP	MGT	
82	2	6	54	WPA2	CCMP	MGT	ECwnet1
832	0	6	54	WPA2	CCMP	MGT	ECwnet1
23	0	6	54	WPA2	CCMP	MGT	
42	0	11	54	WPA2	CCMP	MGT	ECMNV32
173	0	11	54	WPA2	CCMP	MGT	ECMNV32
21	1	11	54	WPA2	CCMP	MGT	
12	0	6	54	WPA2	TKIP	MGT	
234	0	6	54	WPA2	TKIP	MGT	
11	1	6	54	WPA2	TKIP	MGT	
12	0	1	54	WPA2	TKIP	MGT	
132	0	1	54	WPA2	TKIP	MGT	prN67n
10	0	3	54	WPA2	CCMP	MGT	ASFWW
431	0	3	54	WPA2	CCMP	MGT	ASFWW


The client (in scope)
Police station (not in scope)
You
Local bank (not in scope)

Scoping A Wireless Assessment: Red Team Style

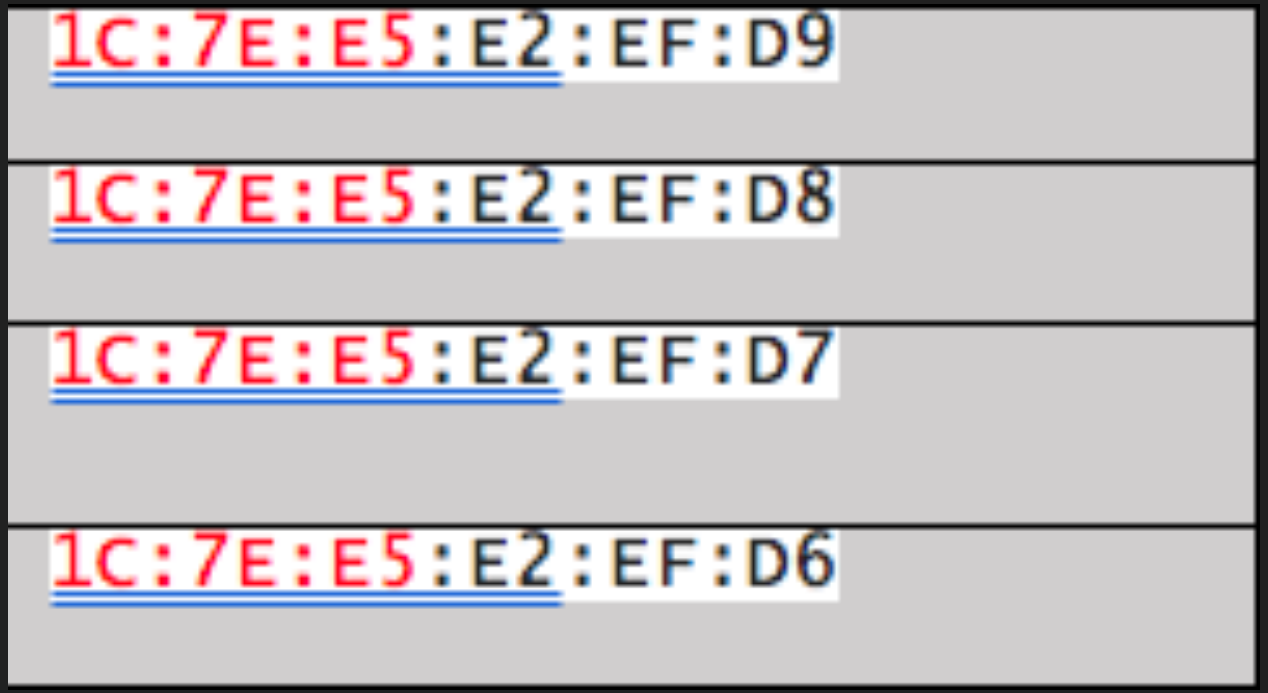
Sequential BSSID Patterns



Linguistic Inference

The process of identifying access points with ESSIDs that are linguistically similar to words or phrases that the client uses to identify itself.

OUI Prefixes



Geographic Cross-referencing

Evil Corp has 57 locations worldwide, If

1. We see the ***same*** ESSID's appear at ***two*** Evil Corp locations
2. AND no other 3rd party is present at both of these locations as well

We can conclude that the ESSID is used by Evil Corp.

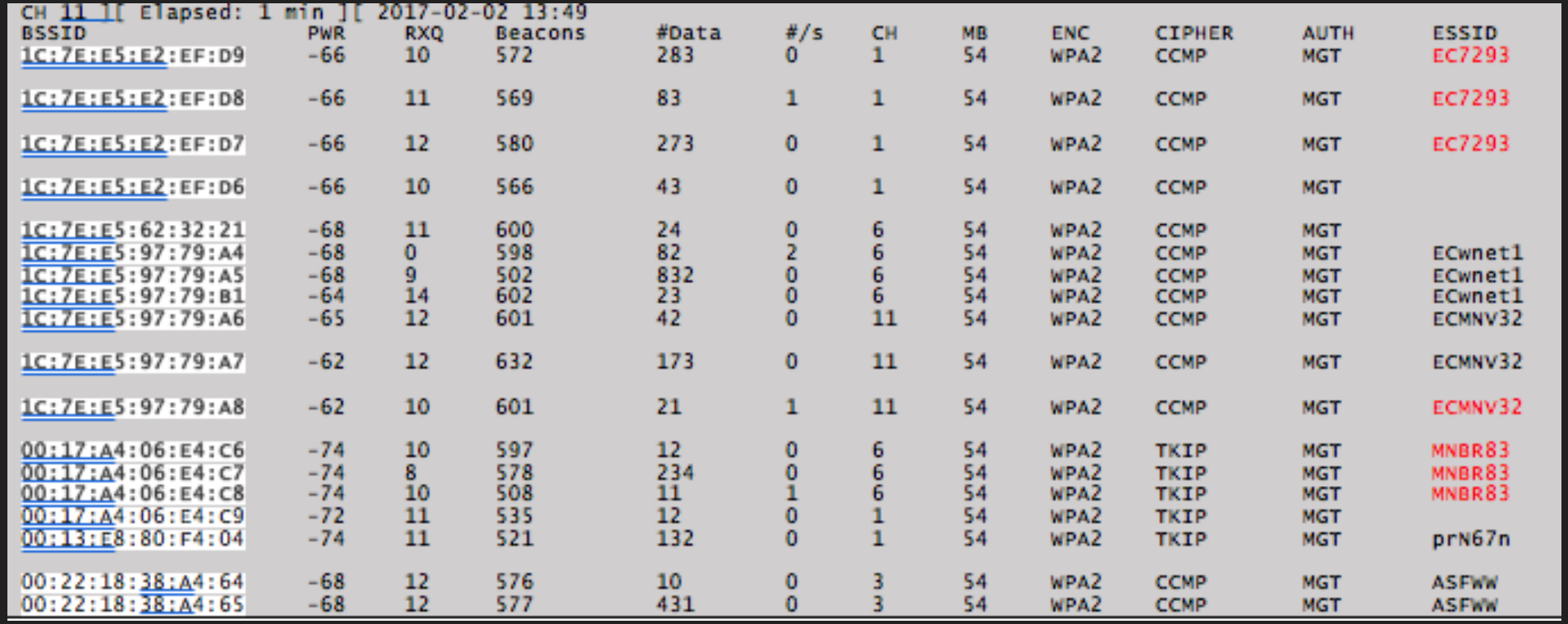
Before we continue...

Attempt to decloak any wireless access points that have hidden ESSIDs:

- Briefly deauthenticating one or more clients from each of the hidden networks
- Our running airodump-ng session will then sniff the ESSID of the affected access point as the client reassociates

```
root@localhost:~# aireplay-ng -b
$BSSID -c $CLIENT_MAC wlan0
```

If successful, the ESSID of the affected access point will appear in our airodump-ng output.



CH 11][Elapsed: 1 min][2017-02-02 13:49

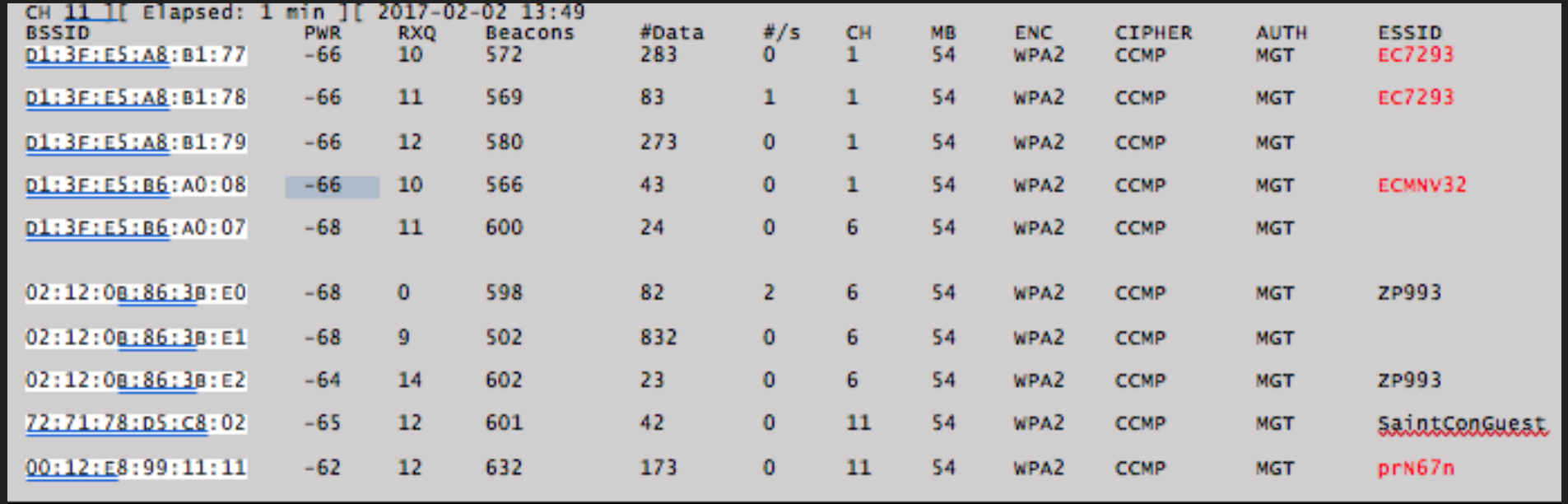
BSSID	PWR	RXQ	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
1C:7E:E5:E2:EF:D9	-66	10	572	283	0	1	54	WPA2	CCMP	MGT	EC7293
1C:7E:E5:E2:EF:D8	-66	11	569	83	1	1	54	WPA2	CCMP	MGT	EC7293
1C:7E:E5:E2:EF:D7	-66	12	580	273	0	1	54	WPA2	CCMP	MGT	EC7293
1C:7E:E5:E2:EF:D6	-66	10	566	43	0	1	54	WPA2	CCMP	MGT	
1C:7E:E5:62:32:21	-68	11	600	24	0	6	54	WPA2	CCMP	MGT	
1C:7E:E5:97:79:A4	-68	0	598	82	2	6	54	WPA2	CCMP	MGT	ECwnet1
1C:7E:E5:97:79:A5	-68	9	502	832	0	6	54	WPA2	CCMP	MGT	ECwnet1
1C:7E:E5:97:79:B1	-64	14	602	23	0	6	54	WPA2	CCMP	MGT	ECwnet1
1C:7E:E5:97:79:A6	-65	12	601	42	0	11	54	WPA2	CCMP	MGT	ECMNV32
1C:7E:E5:97:79:A7	-62	12	632	173	0	11	54	WPA2	CCMP	MGT	ECMNV32
1C:7E:E5:97:79:A8	-62	10	601	21	1	11	54	WPA2	CCMP	MGT	ECMNV32
00:17:A4:06:E4:C6	-74	10	597	12	0	6	54	WPA2	TKIP	MGT	MNBR83
00:17:A4:06:E4:C7	-74	8	578	234	0	6	54	WPA2	TKIP	MGT	MNBR83
00:17:A4:06:E4:C8	-74	10	508	11	1	6	54	WPA2	TKIP	MGT	MNBR83
00:17:A4:06:E4:C9	-72	11	535	12	0	1	54	WPA2	TKIP	MGT	
00:13:E8:80:F4:04	-74	11	521	132	0	1	54	WPA2	TKIP	MGT	prN67n
00:22:18:38:A4:64	-68	12	576	10	0	3	54	WPA2	CCMP	MGT	ASFWW
00:22:18:38:A4:65	-68	12	577	431	0	3	54	WPA2	CCMP	MGT	ASFWW

We have now identified six unique ESSIDs at the client site

- Cross reference these ESSIDs with the results of similar site surveys

Geographic Cross-referencing

Drive to second Evil Corp branch 30 miles away, discover that none of the third-parties present at previous location are present



CH 11][Elapsed: 1 min][2017-02-02 13:49

BSSID	PWR	RXQ	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
D1:3F:E5:A8:B1:77	-66	10	572	283	0	1	54	WPA2	CCMP	MGT	EC7293
D1:3F:E5:A8:B1:78	-66	11	569	83	1	1	54	WPA2	CCMP	MGT	EC7293
D1:3F:E5:A8:B1:79	-66	12	580	273	0	1	54	WPA2	CCMP	MGT	
D1:3F:E5:B6:A0:08	-66	10	566	43	0	1	54	WPA2	CCMP	MGT	ECMNV32
D1:3F:E5:B6:A0:07	-68	11	600	24	0	6	54	WPA2	CCMP	MGT	
02:12:0B:86:3B:E0	-68	0	598	82	2	6	54	WPA2	CCMP	MGT	ZP993
02:12:0B:86:3B:E1	-68	9	502	832	0	6	54	WPA2	CCMP	MGT	
02:12:0B:86:3B:E2	-64	14	602	23	0	6	54	WPA2	CCMP	MGT	ZP993
72:71:78:D5:C8:02	-65	12	601	42	0	11	54	WPA2	CCMP	MGT	SaintConGuest
00:12:E8:99:11:11	-62	12	632	173	0	11	54	WPA2	CCMP	MGT	prN67n

Expanding scope by identifying sequential BSSIDs

CH 11][Elapsed: 1 min][2017-02-02 13:49

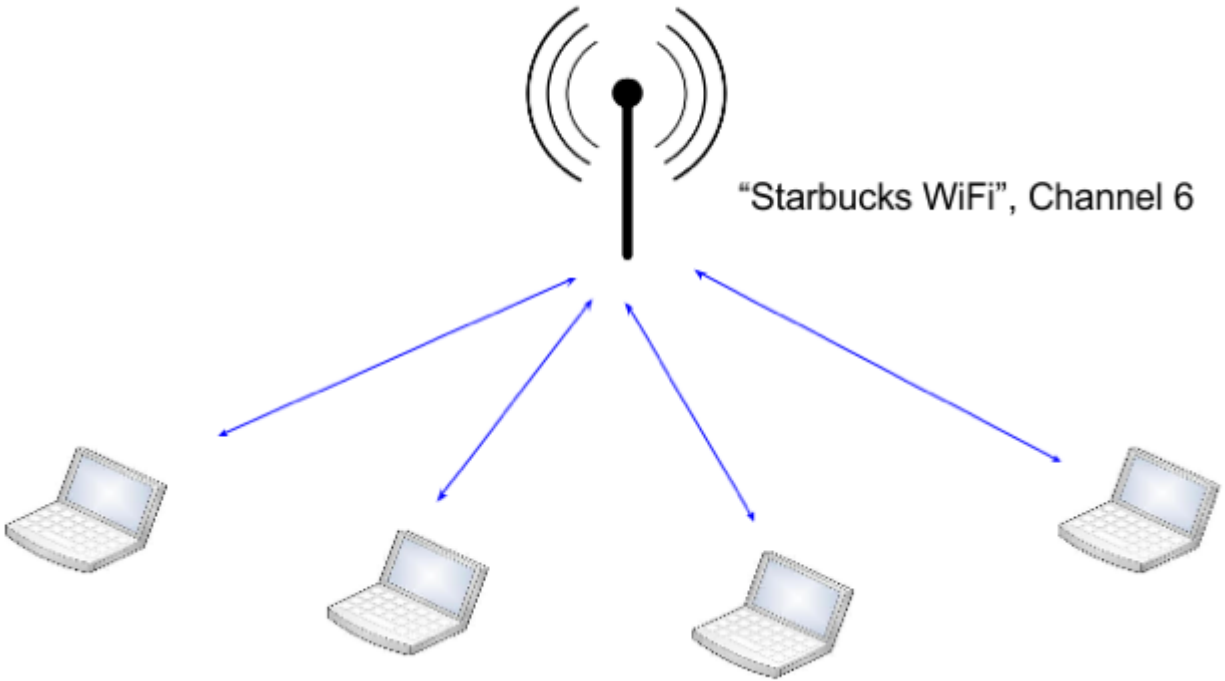
BSSID	PWR	RXQ	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
1C:7E:E5:E2:EF:D9	-66	10	572	283	0	1	54	WPA2	CCMP	MGT	EC7293
1C:7E:E5:E2:EF:D8	-66	11	569	83	1	1	54	WPA2	CCMP	MGT	EC7293
1C:7E:E5:E2:EF:D7	-66	12	580	273	0	1	54	WPA2	CCMP	MGT	EC7293
1C:7E:E5:E2:EF:D6	-66	10	566	43	0	1	54	WPA2	CCMP	MGT	
1C:7E:E5:62:32:21	-68	11	600	24	0	6	54	WPA2	CCMP	MGT	
1C:7E:E5:97:79:A4	-68	0	598	82	2	6	54	WPA2	CCMP	MGT	ECwnet1
1C:7E:E5:97:79:A5	-68	9	502	832	0	6	54	WPA2	CCMP	MGT	ECwnet1
1C:7E:E5:97:79:B1	-64	14	602	23	0	6	54	WPA2	CCMP	MGT	ECwnet1
1C:7E:E5:97:79:A6	-65	12	601	42	0	11	54	WPA2	CCMP	MGT	ECMNV32
1C:7E:E5:97:79:A7	-62	12	632	173	0	11	54	WPA2	CCMP	MGT	ECMNV32
1C:7E:E5:97:79:A8	-62	10	601	21	1	11	54	WPA2	CCMP	MGT	ECMNV32
00:17:A4:06:E4:C6	-74	10	597	12	0	6	54	WPA2	TKIP	MGT	MNBR83
00:17:A4:06:E4:C7	-74	8	578	234	0	6	54	WPA2	TKIP	MGT	MNBR83
00:17:A4:06:E4:C8	-74	10	508	11	1	6	54	WPA2	TKIP	MGT	MNBR83
00:17:A4:06:E4:C9	-72	11	535	12	0	1	54	WPA2	TKIP	MGT	
00:13:E8:80:F4:04	-74	11	521	132	0	1	54	WPA2	TKIP	MGT	prN67n

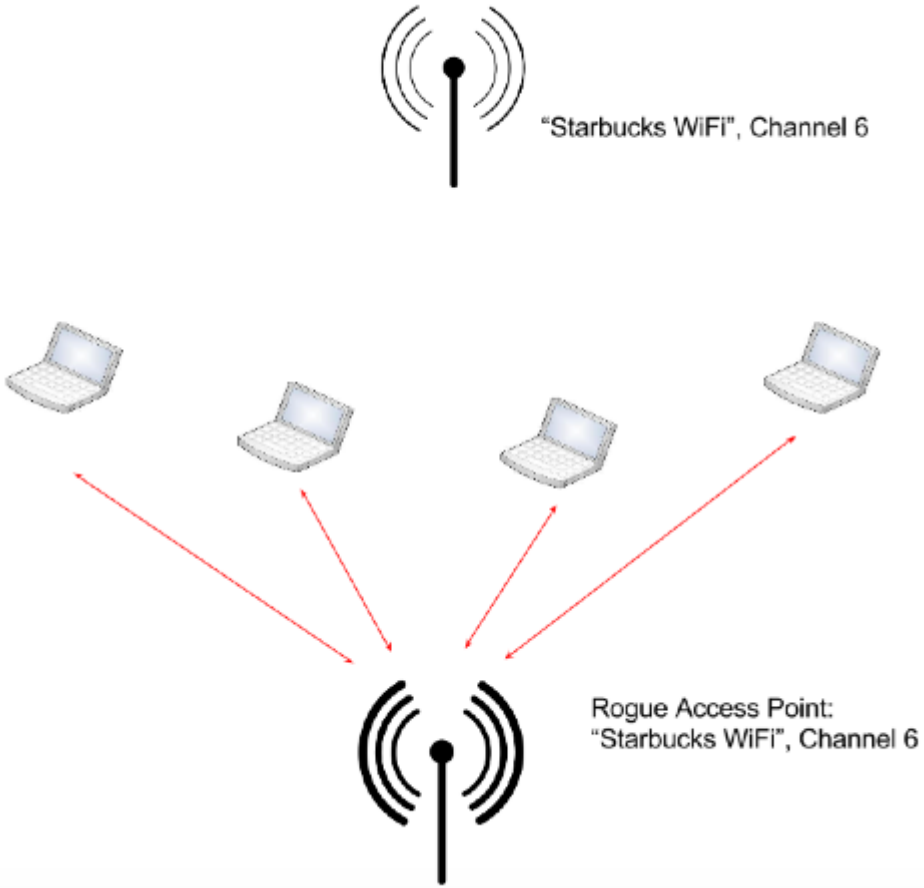
Attacking and Gaining Entry to WPA2-EAP Networks

Attacking & gaining access to WPA2-EAP Networks:

Rogue access point attacks:

- Bread and butter of modern wireless penetration tests
- Stealthy MITM attacks
- Steal RADIUS credentials
- Captive portals


“Starbucks WiFi”, Channel 6


"Starbucks WiFi", Channel 6
Rogue Access Point:
"Starbucks WiFi", Channel 6

Wireless Theory: WPA2-EAP Networks

Wireless Theory: EAP

Most commonly used EAP implementations:

- PEAP
- TTLS

Wireless Theory: EAP

Logically:

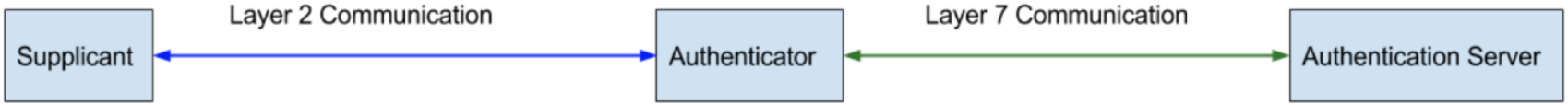
- Authentication occurs between supplicant and authentication server

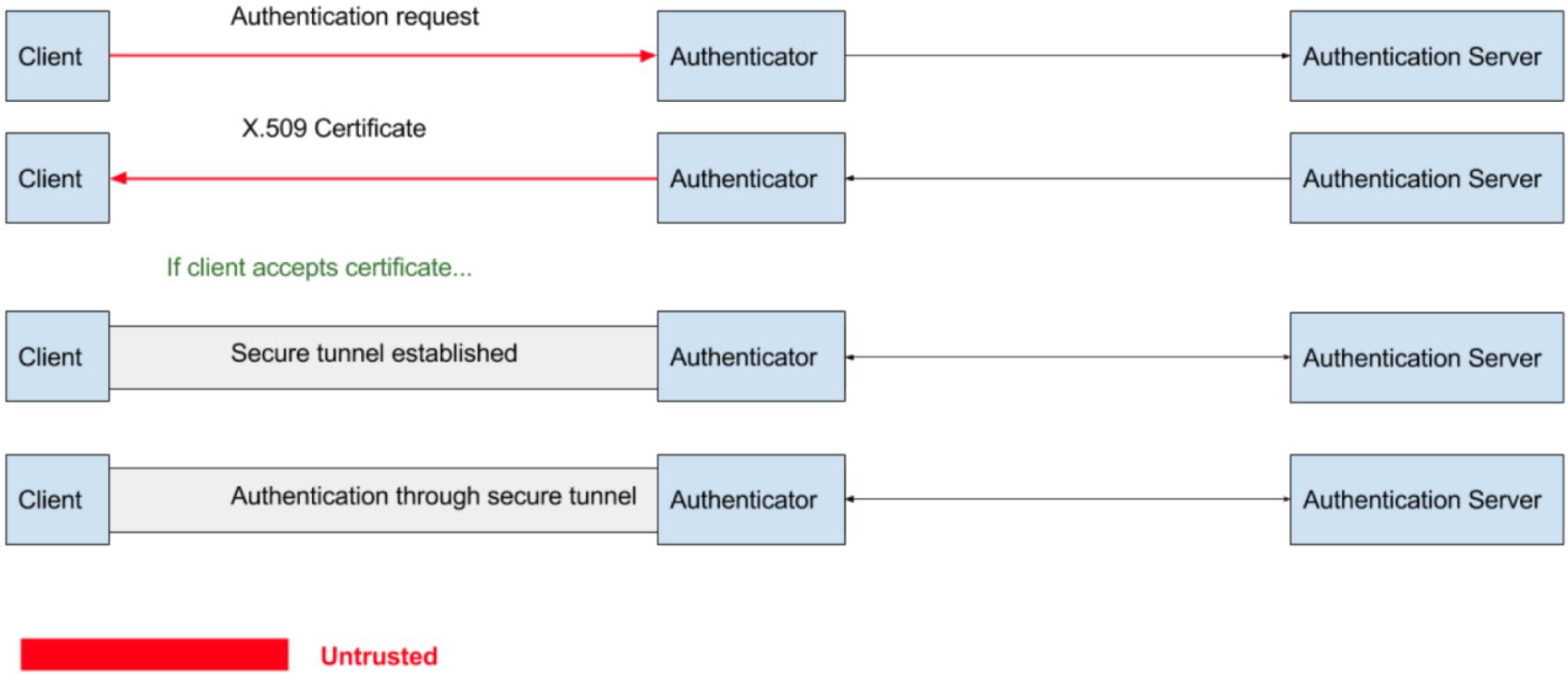

Client
Authentication request
Authentication Server
Client
X.509 Certificate
Authentication Server
If client accepts certificate...
Client
Secure tunnel established
Authentication Server
Client
Authentication through secure tunnel
Authentication Server

Wireless Theory: EAP

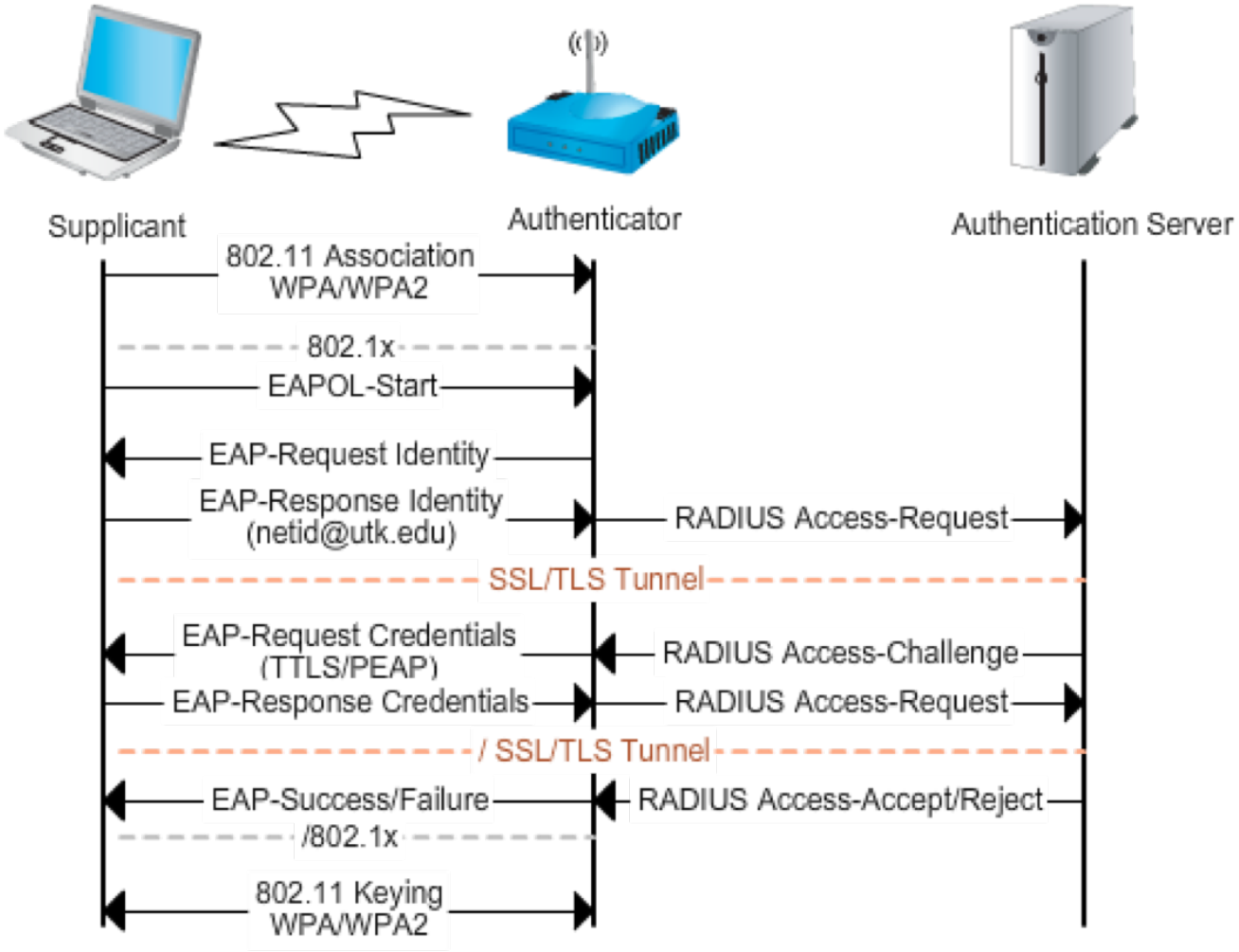
Without secure tunnel, auth process sniffable.

- Attacker sniffs challenge and response then derives password offline
- Legacy implementations of EAP susceptible to this (i.e. EAP-MD5)


Supplicant
Layer 2 Communication
Authenticator
Layer 7 Communication
Authentication Server


Client
Authentication request
Authenticator
Authentication Server
Client
X.509 Certificate
Authenticator
Authentication Server
If client accepts certificate...
Client
Secure tunnel established
Authenticator
Authentication Server
Client
Authentication through secure tunnel
Authenticator
Authentication Server
Untrusted

Wireless Theory: EAP-PEAP


Supplicant
Authenticator
Authentication Server
802.11 Association
WPA/WPA2
802.1x
EAPOL-Start
EAP-Request Identity
EAP-Response Identity
(netid@utk.edu)
RADIUS Access-Request
SSL/TLS Tunnel
EAP-Request Credentials
(TTLS/PEAP)
RADIUS Access-Challenge
EAP-Response Credentials
RADIUS Access-Request
/ SSL/TLS Tunnel
EAP-Success/Failure
RADIUS Access-Accept/Reject
/802.1x
802.11 Keying
WPA/WPA2

Wireless Theory: EAP-PEAP

Until tunnel is established, AP is essentially an open access point

- Auth process over secure tunnel, but encrypted SSL/TLS packets sent over open wireless
- WPA2 doesn’t kick in until auth is complete

Wireless Theory: EAP-PEAP

Open wireless networks vulnerable to Evil Twin attacks beacuse there is no way to verify the identity of the AP.

EAP-PEAP networks vulnerable to Evil Twin attacks because there is no way to verify the identity of the authenticator.

Wireless Theory: EAP-PEAP

In theory, x509 cert issued by auth server used to verify identify:

- Only true if invalid certs rejected
- Many supplicants do not perform proper cert validation
- Many others configured to accept invalid certs automatically
- Even if supplicant configured correctly, onus is placed on user to reject invalid certs

Wireless Theory: EAP-PEAP

To compromise EAP-PEAP:

- Attacker first performs evil twin against target AP
- Client connects to rogue AP
- If client accepts invalid cert, authenticates with attacker
- Challenge/Response cracked offline

Wireless Theory: EAP-PEAP

To compromise EAP-PEAP:

1. Attacker first performs evil twin against target AP
2. Client connects to rogue AP
3. If client accepts invalid cert, authenticates with attacker
4. Challenge/Response cracked offline

Lab Exercise: Evil Twin Attack Using Hostapd-WPE

Wireless MITM Attacks

Wireless MITM Attacks

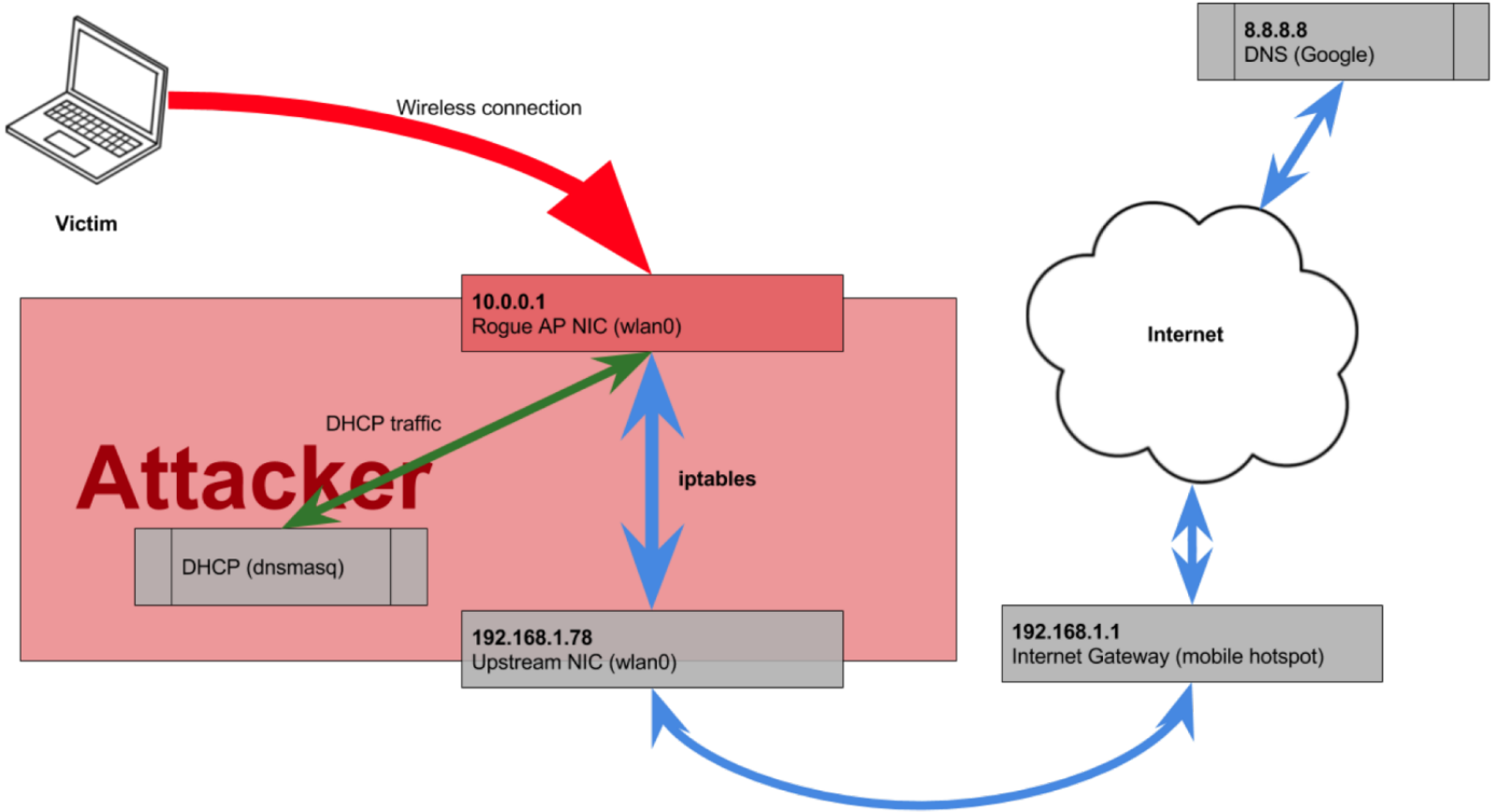
Wireless MITM Attack:

- Uses evil twin or karma, attacker acts as functional access point
- Does not degrade target network
- Stealthy: attack does not generate additional traffic on network

Wireless MITM Attacks

Extending our Rogue AP to perform MITM:

- Configured to behave like a wireless router
- DHCP server to provide IPs to clients
- DNS for name resolution
- OS must support packet forwarding
- Need means of redirecting packets from one interface to another

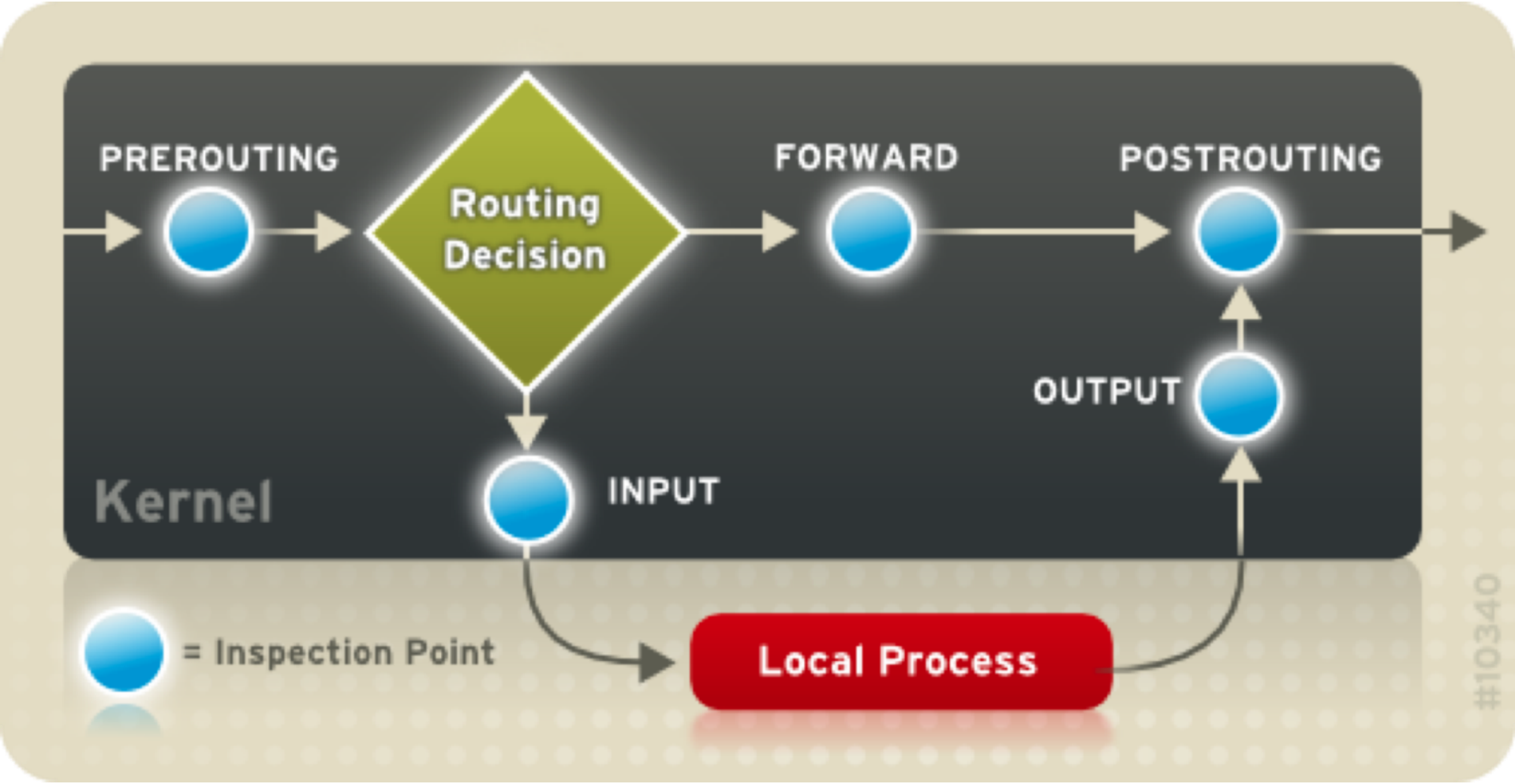

Victim
Wireless connection
8.8.8.8
DNS (Google)
Internet
10.0.0.1
Rogue AP NIC (wlan0)
DHCP traffic
Attacker
iptables
DHCP (dnsmasq)
192.168.1.78
Upstream NIC (wlan0)
192.168.1.1
Internet Gateway (mobile hotspot)

Wireless MITM Attacks

Our setup:

- Dnsmasq → DHCP server
- DHCP Option → Google as DNS server
- OS → Linux… packet forwarding, iptables

Lab 2: Configuring Linux As A Router


PREROUTING
Routing Decision
FORWARD
POSTROUTING
OUTPUT
Kernel
INPUT
= Inspection Point
Local Process
#10340

Classic HTTPS Downgrade Attack

Classic HTTPS Downgrade Attack

We’ve turned our OS into a wireless router:

- Let’s now focus on using it to steal creds

Classic HTTPS Downgrade Attack

SSLStrip:

- Creates a log of HTTP traffic sent to/from victim
- Logs credentials and other sensitive data
- Breaks encryption of of any HTTPS traffic it encounters

Classic HTTPS Downgrade Attack

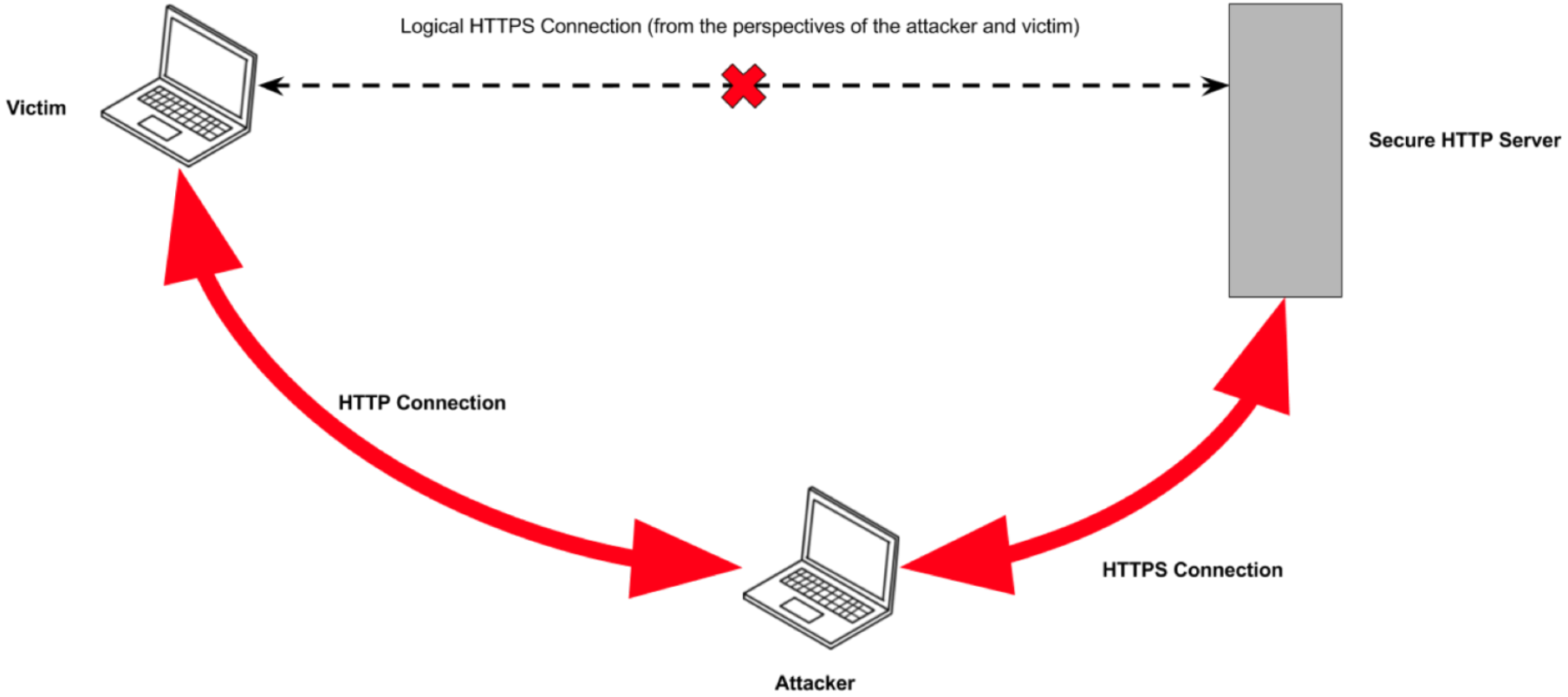
SSLStripping Attack:

- First documented by Moxie Marlinspike
- Attacker creates MITM between victim and gateway
- Attacker acts as HTTP(S) proxy
- When the victim makes a request to access a secure resource, such as a login page, the attacker receives the request and forwards it to the server.
- From the server’s perspective, the request appears to have been made by the attacker

Classic HTTPS Downgrade Attack

SSLStripping Attack:

- Encrypted tunnel is established between the attacker and the server (instead of between the victim and the server)
- Attacker then modifies the server's response, converting it from HTTPS to HTTP, and forwards it to the victim.
- From the victim's perspective, the server has just issued it an HTTP response [6].


Logical HTTPS Connection (from the perspectives of the attacker and victim)
Victim
Secure HTTP Server
HTTP Connection
HTTPS Connection
Attacker

Classic HTTPS Downgrade Attack

SSLStripping Attack:

1. All subsequent requests from the victim and the server will occur over an unencrypted HTTP connection with the attacker
2. Attacker will forward these requests over an encrypted connection with the HTTP server.
3. Since the victim’s requests are sent to the attacker in plaintext, they can be viewed or modified by the attacker
4. Server believes that it has established a legitimate SSL connection with the victim, and the victim believes that the attacker is a trusted web server
5. No certificate errors will occur on the client or the server, rendering both affected parties unaware that the attack is taking place

Classic HTTPS Downgrade Attack

Modified bash script (iptables):

iptables --table nat --append PREROUTING --protocol tcp --destination-port 80 --jump REDIRECT --to-port 10000

iptables --table nat --append PREROUTING --protocol tcp --destination-port 443 --jump REDIRECT --to-port 10000

Classic HTTPS Downgrade Attack

Modified bash script (sslstrip):

python -l 10000 -p -w ./sslstrip.log

Lab 3: HTTPS Downgrade

Downgrading Modern HTTPS Implementations With Partial HSTS Bypasses

Partial HSTS Bypasses

Repeat previous lab exercise against Bing. The attack should fail.

Why?

- HSTS

Partial HSTS Bypasses

What is HSTS?

- Enhancement of HTTPS protocol
- Designed to mitigate weaknesses exploited by tools such as SSLStrip

Partial HSTS Bypasses

Client makes HTTP(S) request to HSTS enabled site, receives the following response headers:

Strict-Transport-Security: max-age=31536000

Tells browser to always load content from domain over HTTPS.

Partial HSTS Bypasses

- Modern browsers maintain HSTS list.
- HSTS headers cause domain to be added to list.
- User attempts to access site over HTTP → browser first checks HSTS list
- If domain in HSTS list, 307 internal redirect to HTTPS

Partial HSTS Bypasses

Include subdomains attribute:

Strict-Transport-Security: max-age=31536000; includeSubdomains

*.evilcorp.com VS. evilcorp.com

Partial HSTS Bypasses

HSTS Preload list available to developers:

- Domain treated as HSTS regardless of whether browser has received headers

Partial HSTS Bypasses

HSTS: effective mitigation ***unless*** the following are true:

- Domain not added to preload list ***with*** the includeSubdomains attribute set
- Server issues HSTS headers ***without*** the includeSubdomains attribute

Partial HSTS Bypasses

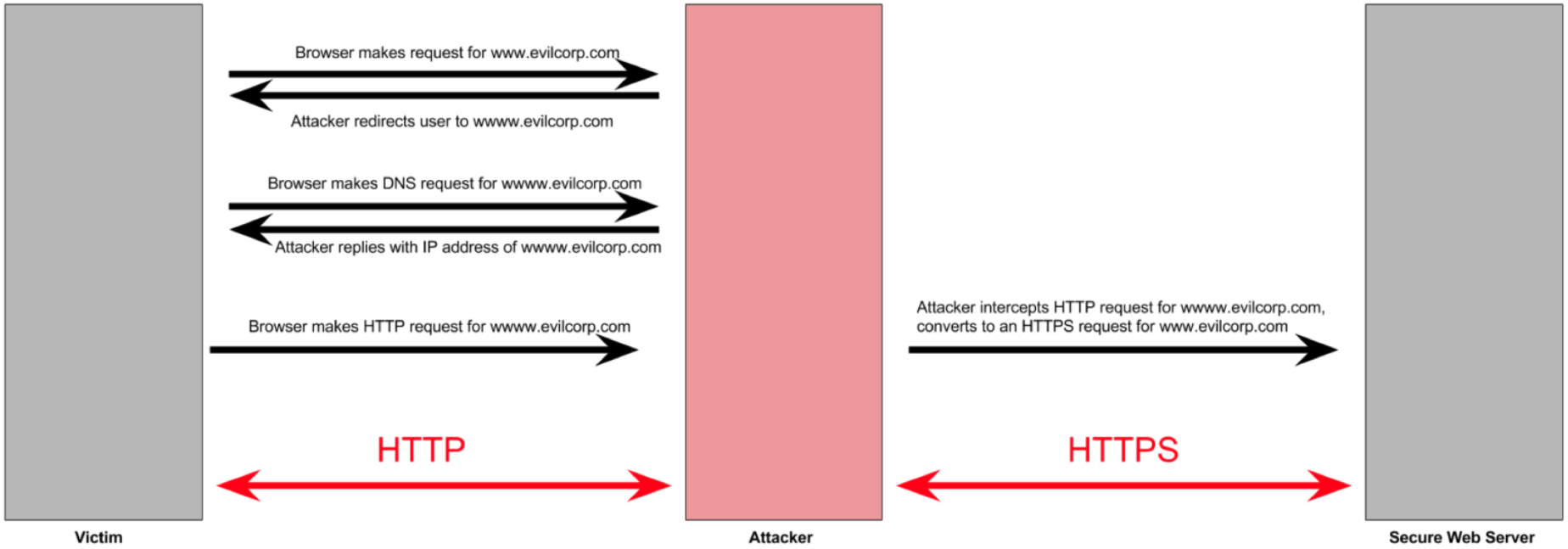
LeonardoNve: Partial HSTS bypass

- *Blackhat Asia 2014 - OFFENSIVE: Exploiting changes on DNS server configuration*
- Source code actually banned in Spain, gag order issued

Partial HSTS Bypasses

Partial HSTS bypass

- Attacker first establishes MITM as with SSL Stripping attack
- Attacker *also* proxies and modifies DNS traffic
- www.evilcorp.com → wwww.evilcorp.com
- Sidesteps HSTS, since browser has not received HSTS headers for wwww.evilcorp.com


Browser makes request for www.evilcorp.com
Attacker redirects user to wwww.evilcorp.com
Browser makes DNS request for wwww.evilcorp.com
Attacker replies with IP address of wwww.evilcorp.com
Browser makes HTTP request for wwww.evilcorp.com
Attacker intercepts HTTP request for wwww.evilcorp.com, converts to an HTTPS request for www.evilcorp.com
HTTP
HTTPS
Victim
Attacker
Secure Web Server

Partial HSTS Bypasses

Partial HSTS Bypass is effective *if*:

- Victim doesn’t notice modified subdomain
- Cert pinning not used

Partial HSTS Bypasses

Choose a believable subdomain:

- www.evilcorp.com → wwww.evilcorp.com

VS.

- mail.evilcorp.com → mailbox.evilcorp.com

Partial HSTS Bypass: Modified Bash Script

Modified bash script (iptables):

iptables --table nat --append PREROUTING --protocol udp --destination-port 53 --jump REDIRECT --to-port 53

Partial HSTS Bypass: Modified Bash Script

Modified bash script (use sslstrip2 and dns2proxy):

```
python /opt/sslstrip2/sslstrip2.py -l 10000 -p -w ./sslstrip.log &

python /opt/dns2proxy/dns2proxy.py –i $phy &
```

Lab 4: Partial HSTS Bypasses

LLMNR/NBT-NS Poisoning

LLMNR/NBT-NS Poisoning

NetBIOS name resolution:

1. Check local cache
2. Check LMHosts file
3. DNS lookup using local nameservers
4. LLMNR broadcast to entire subnet
5. NBT-NS broadcast to entire subnet

LLMNR/NBT-NS Poisoning

LLMNR/NBT-NS:

- Different mechanisms, but same logical functionality
- Best understood through example

LLMNR/NBT-NS Poisoning

Two Windows computers named Alice and Leeroy:

1. Alice wants to request file from Leeroy, but does not know Leeroy's IP
2. Alice attempts to resolve Leeroy's name locally and using DNS, but fails
3. Alice makes broadcast requests using LLMNR/NBT-NS
4. Every computer on Alice's subnet receives request
5. Honor system: only Leeroy responds

LLMNR/NBT-NS Poisoning

No honor among thieves:

1. If Alice receives two responses, first one is considered valid
2. Creates race condition
3. Attacker waits for LLMNR/NBT-NS queries, responds to all of them
4. Victim sends traffic to the attacker

LLMNR/NBT-NS Poisoning

NetBIOS name resolution used for things such as:

- Remote login
- Accessing SMB shares

Great way to quickly obtain password hashes.

Lab 5: LLMNR/NBT-NS Poisoning Using Responder

SMB Relay Attacks

SMB Relay Attacks

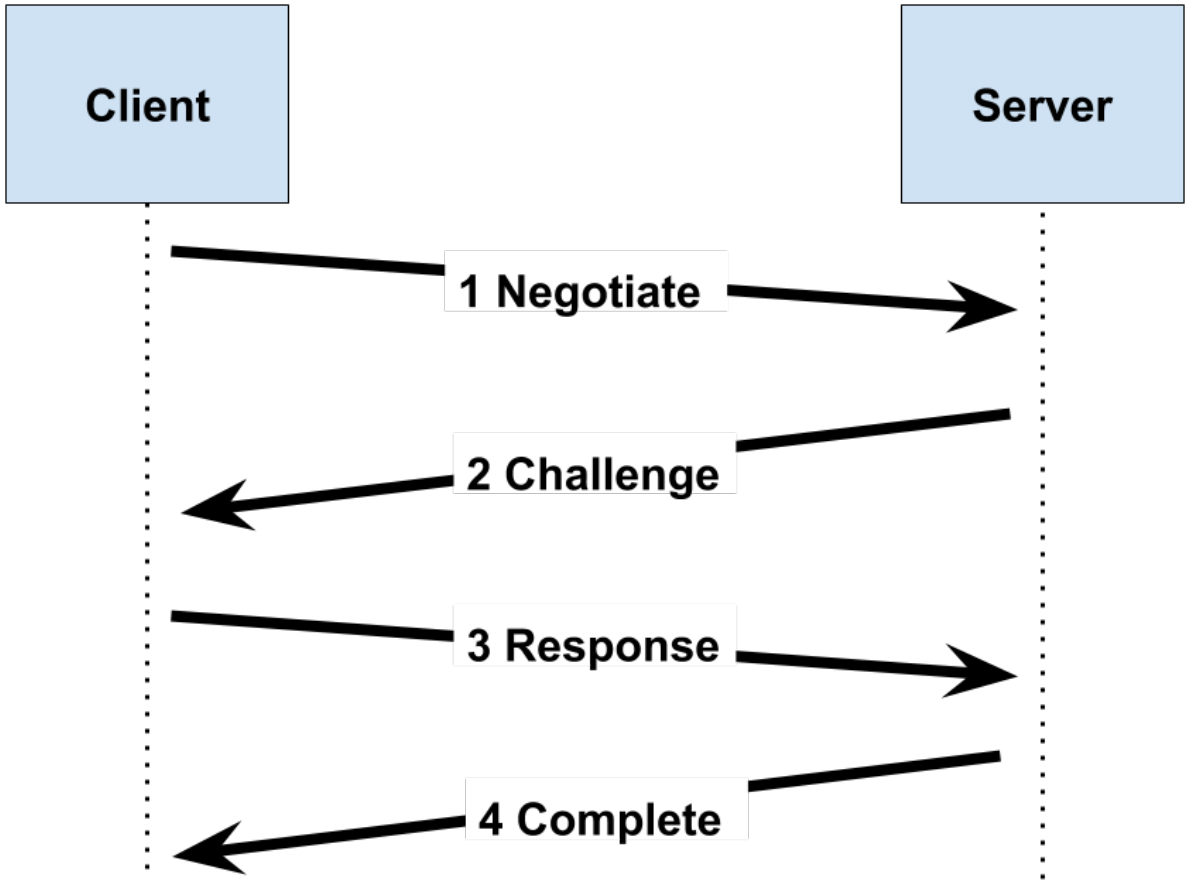
NTLM is a simple authentication protocol:

- challenge/response mechanism (remember PEAP? ;))

SMB Relay Attacks

NTLM authentication process:

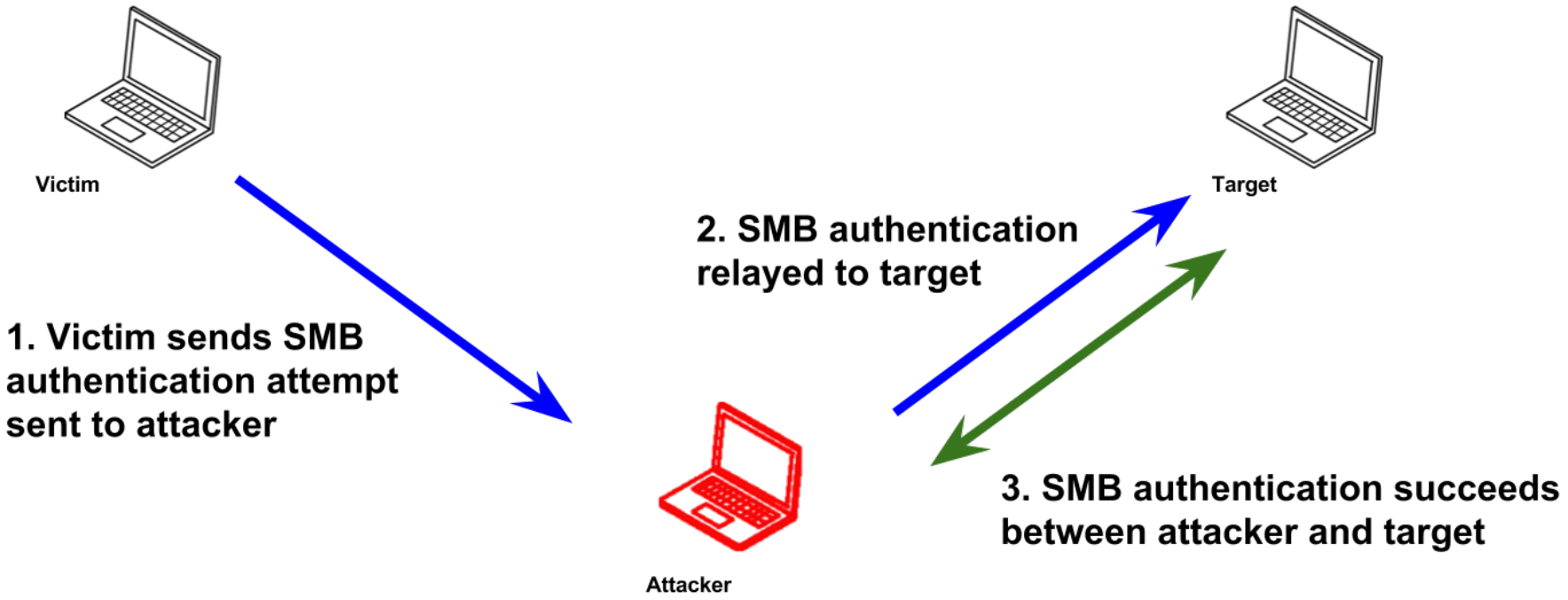
1. Client attempts to authenticate
2. Server issues challenge to client (string of characters)
3. Client encrypts the challenge using its password hash (uh oh)
4. Client sends encrypted challenge back to server as response
5. Server decrypts response using *same* password hash (oh no)
6. If decrypted response === challenge, then auth attempt succeeds


Client
Server
1 Negotiate
2 Challenge
3 Response
4 Complete

SMB Relay Attacks

SMB Relay Attack:

1. Attacker uses MITM or similar to intercept/view NTLM traffic
2. Attacker waits for client to authenticate using NTLM
3. Auth attempt relayed by the attacker to the target server
4. Target server sends challenge back to attacker, which is relayed to client
5. Client response is relayed back to target server
6. Attacker is authenticated with server


Victim
Target
2. SMB authentication relayed to target
1. Victim sends SMB authentication attempt sent to attacker
Attacker
3. SMB authentication succeeds between attacker and target

SMB Relay Attacks

Very powerful:

- Sysadmins often use automated scripts for routine tasks
- Scripts use NTLM for authentication
- Prime candidates for LLMNR/NBT-NS poisoning and SMB relay attacks
- Ironically, agentless NACs and virus scanners use NTLM as well

SMB Relay Attacks

Mitigation: SMB signing

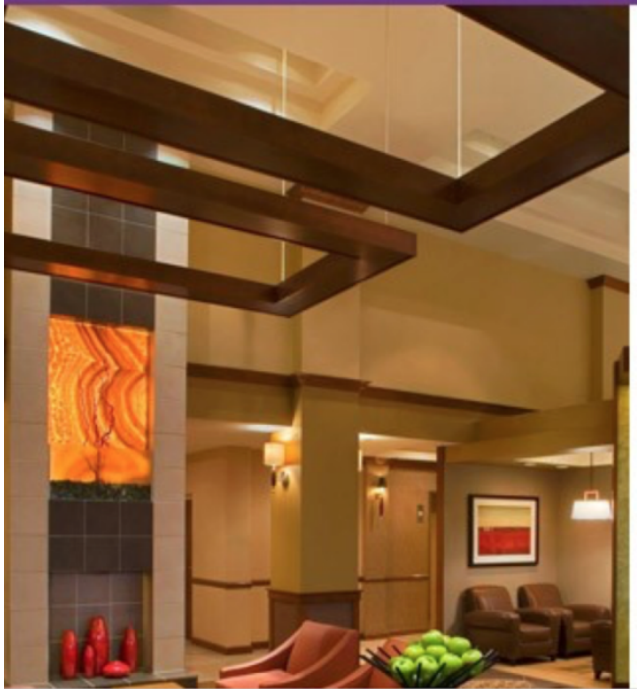
- Packets digitally signed to confirm their authenticity
- Most modern Windows operating systems support SMB signing
- Only Domain Controllers have SMB signing enabled by default

Lab 6: SMB Relay Attack Using impacket and Empire

Hostile-Portal Attacks

Join “@Hyatt_WiFi”

HYATT PLACE®



Choose Your Login Option

FIRST-TIME USERS

▸ **Internet Plan**

▾ **In-House Guest**

Room Number:

Last Name:

Submit

Hostile-Portal Attacks

Captive Portal

- Used to restrict access to an open WiFi-network

dnsmasq config file: original

dhcp-range=10.0.0.80,10.0.0.254,6h

dhcp-option=6,8.8.8.8

dhcp-option=3,10.0.0.1 #Gateway

dhcp-authoritative

log-queries

dnsmasq config file: revised

dhcp-range=10.0.0.80,10.0.0.254,6h

dhcp-option=6,10.0.0.1 # set phy as nameserver

dhcp-option=3,10.0.0.1 #Gateway

dhcp-authoritative

log-queries

Use dnsspoof to resolve all queries to AP

root@localhost:~# echo ’10.0.0.1’ > dnsspoof.conf

root@localhost:~# dnsspoof -i wlan0 -f ./dnsspoof.conf

First shortcoming:

Devices may ignore DHCP options

→ nameserver specified manually instead

Solution: redirect DNS traffic using iptables

iptables --table nat --append PREROUTING --protocol udp -

-destination-port 53 --jump REDIRECT --to-port 53

Second problem: DNS caching

Captive portal will fail until entries in cache expire

Hacky workaround: redirect HTTP(S) using iptables

root@localhost:~# iptables --table nat --append PREROUTING --protocol tcp --destination-port 80 --jump REDIRECT --to-port 80

root@localhost:~# iptables --table nat --append PREROUTING --protocol tcp --destination-port 443 --jump REDIRECT --to-port 443

Lab 7: Configuring Linux As A Captive Portal

Hostile Portal Attacks

Steal Active Directory creds from wireless network ***without network access***.

Redirect to SMB

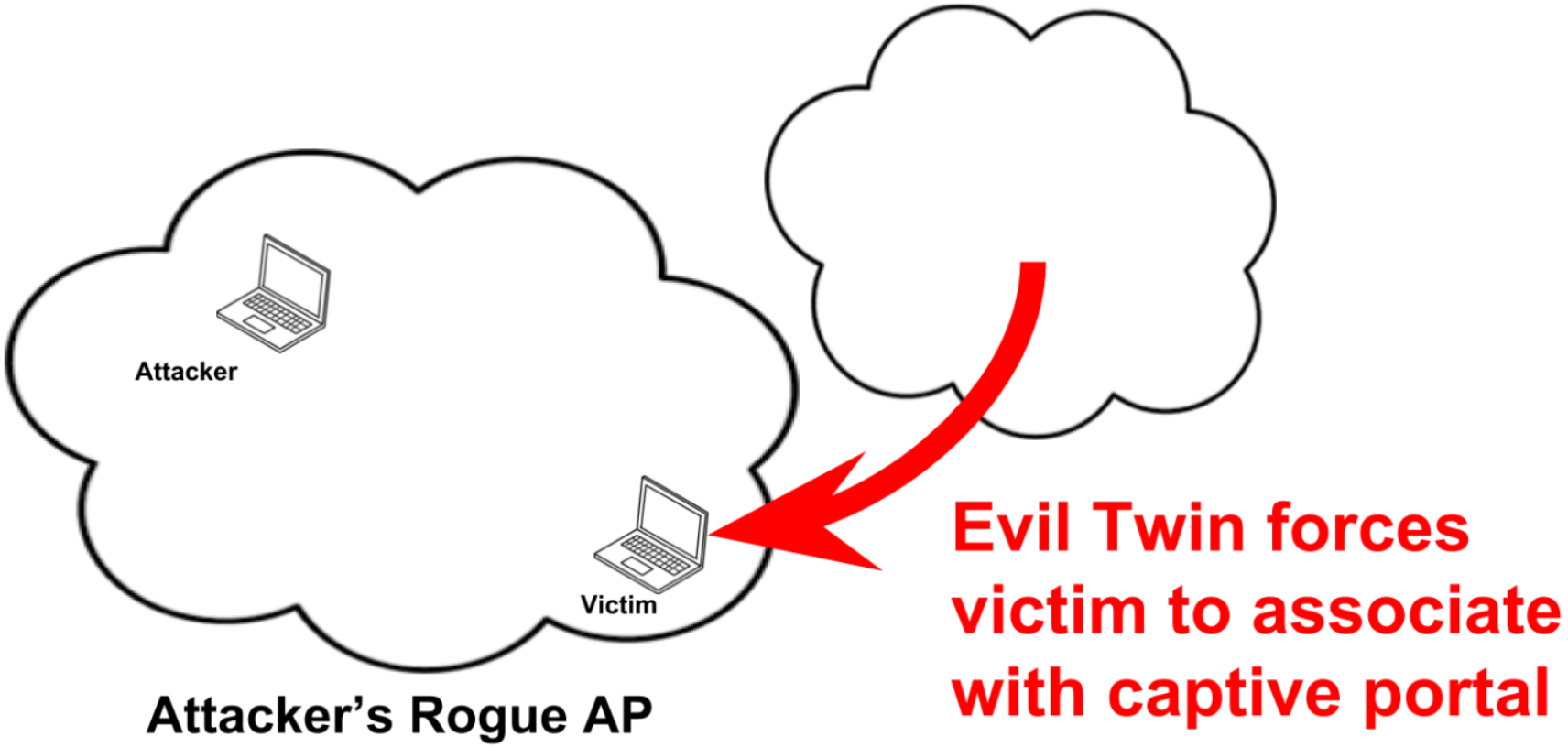
- The idea is to force the victim to visit an HTTP endpoint that redirects to a non-existent SMB share, triggering LLMNR/NBT-NS
- Fast way to get hashes
- Requires social engineering

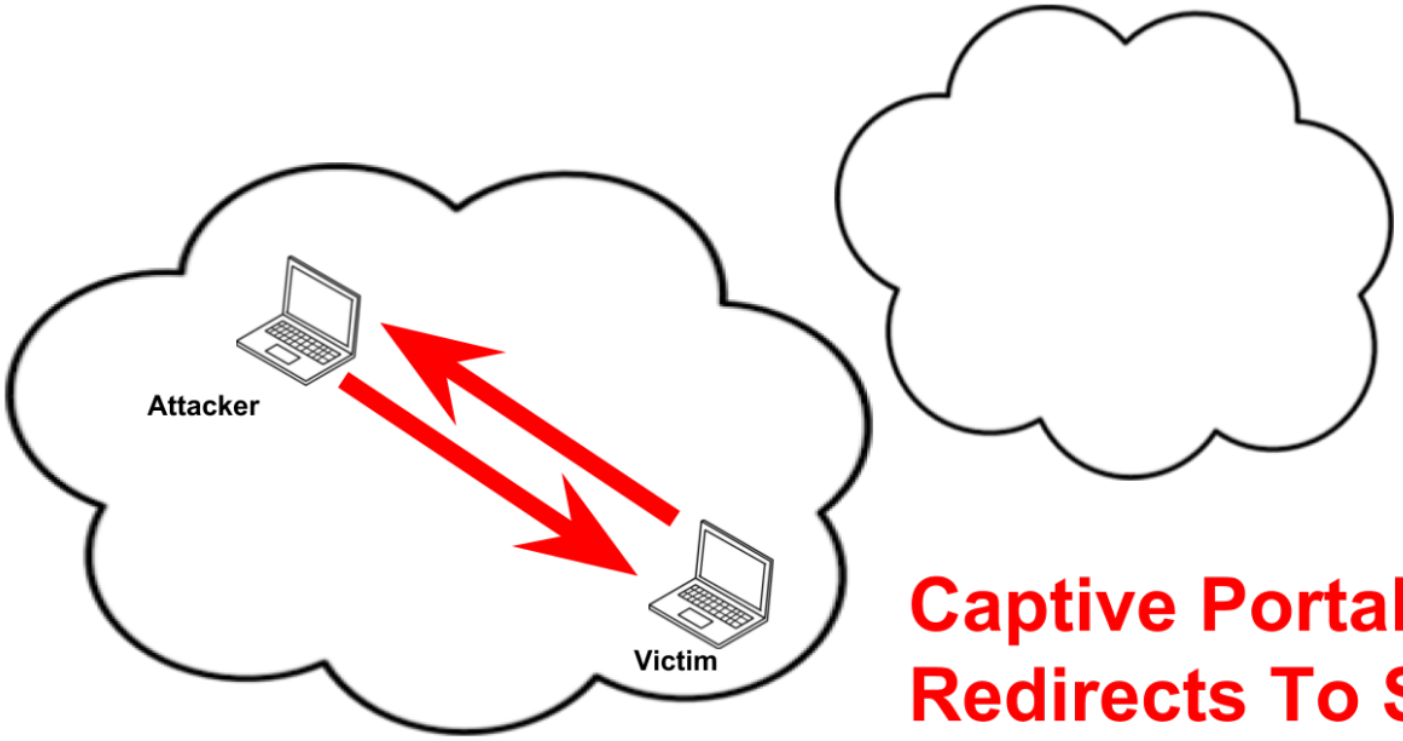
Hostile Portal Attack

- Based on Redirect to SMB Attack
- Victim forced to connect to AP using evil twin attack
- All HTTP(S) traffic is redirected to a non-existent SMB share instead of a captive portal attack, triggering LLMNR/NBT-NS

Attacker

Victim


Attacker
Victim
Attacker's Rogue AP
Evil Twin forces victim to associate with captive portal



Attacker's Rogue AP

Captive Portal Redirects To SMB

Wireless Theory: Hostile Portal Attacks

Consider the following scenario:

- We have breached a wireless network that is used to provide access to sensitive internal network resources

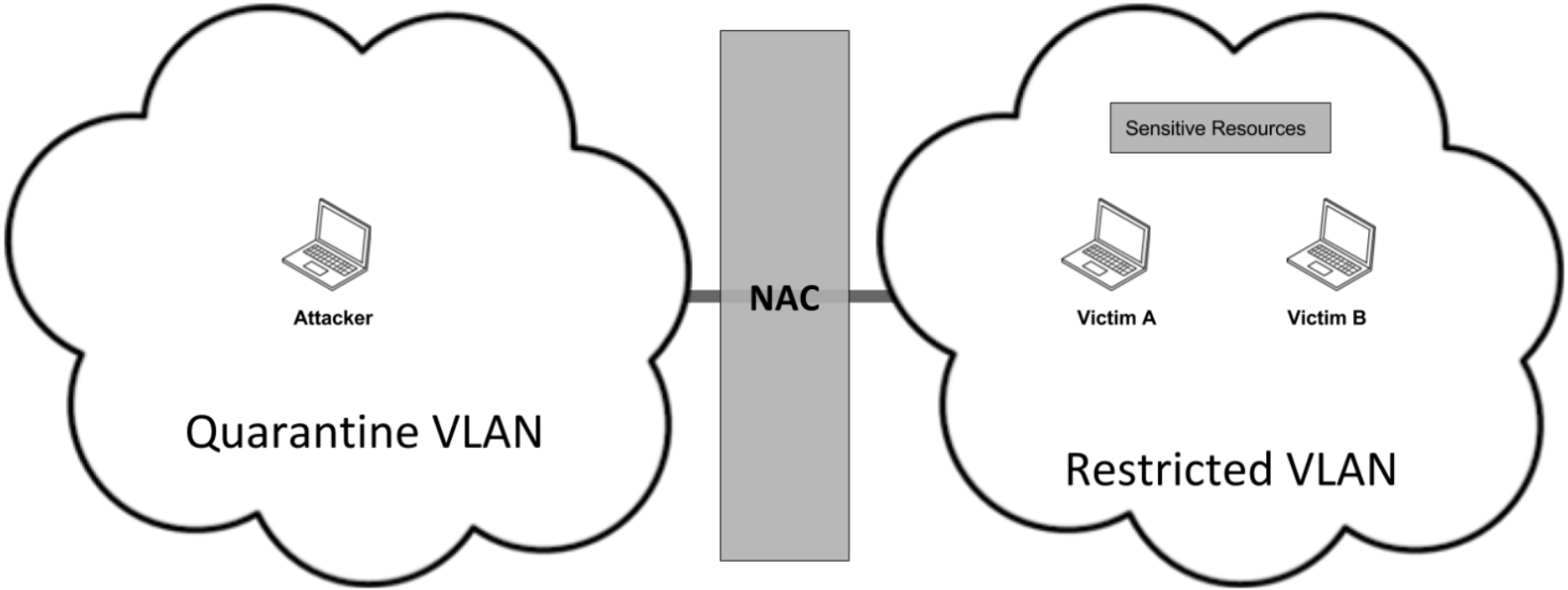
Wireless Theory: Hostile Portal Attacks

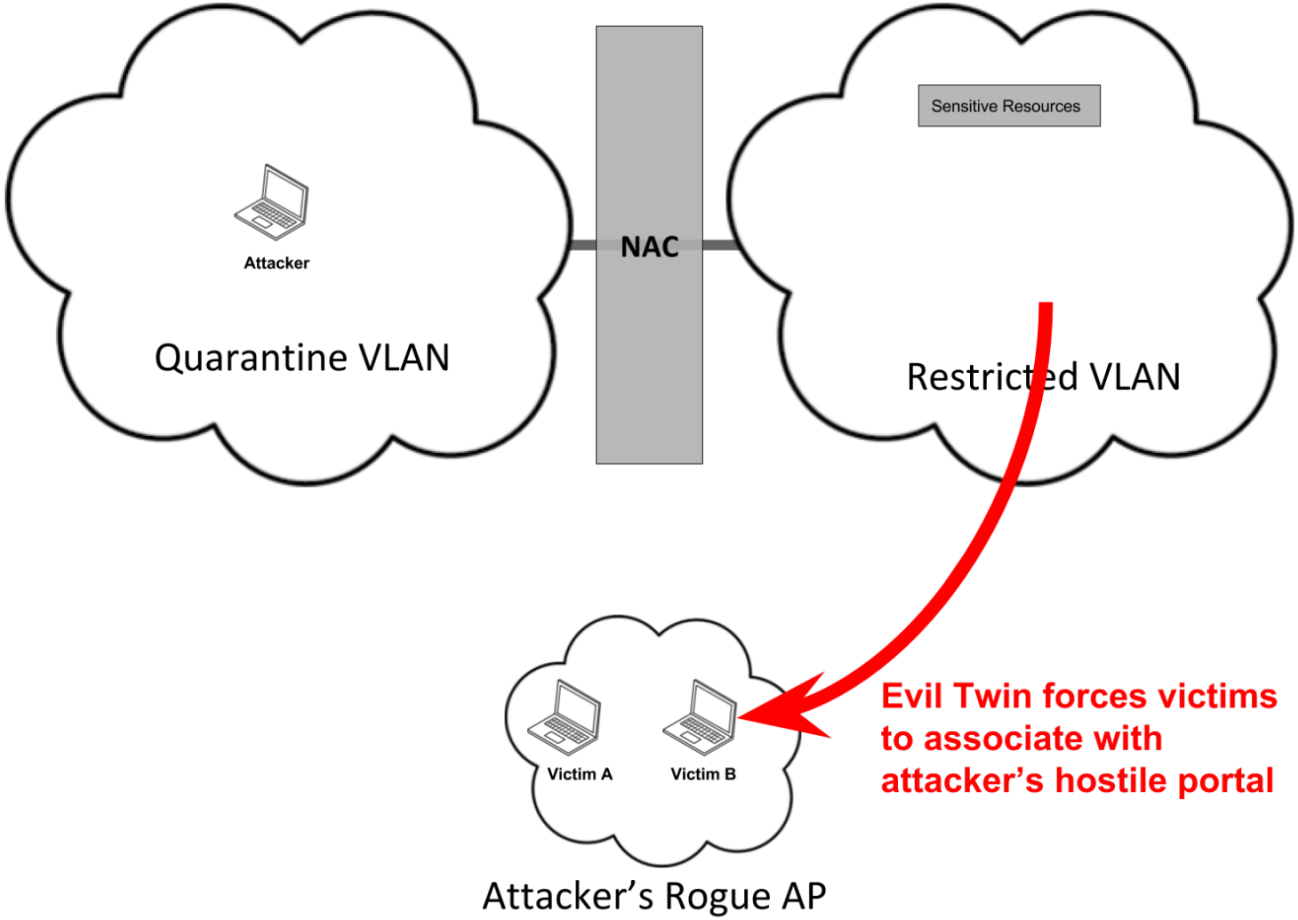
Network architecture:

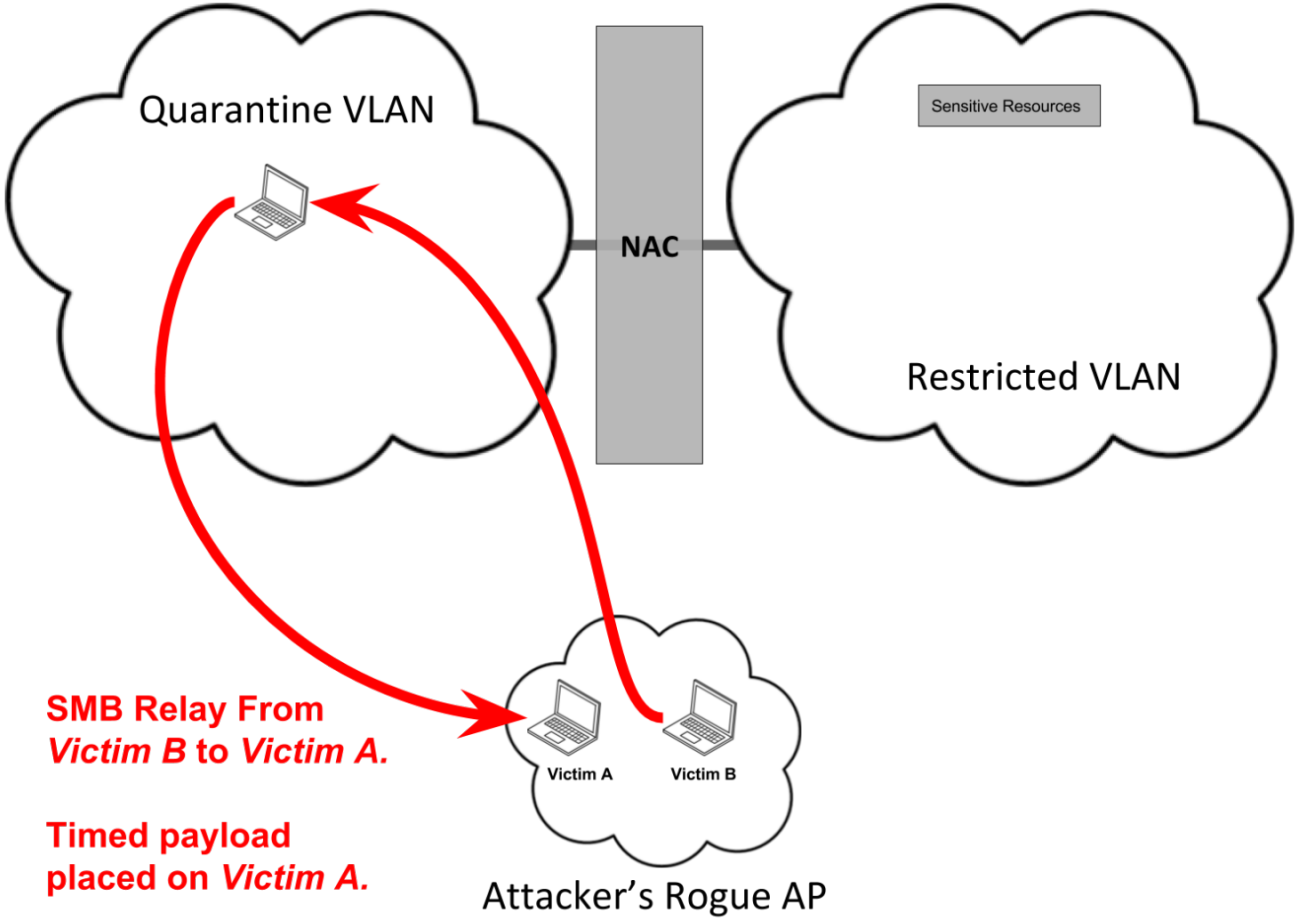
- We have been placed in a quarantine VLAN by a NAC system
- Sensitive resources on restricted VLAN, cannot be accessed from quarantine VLAN

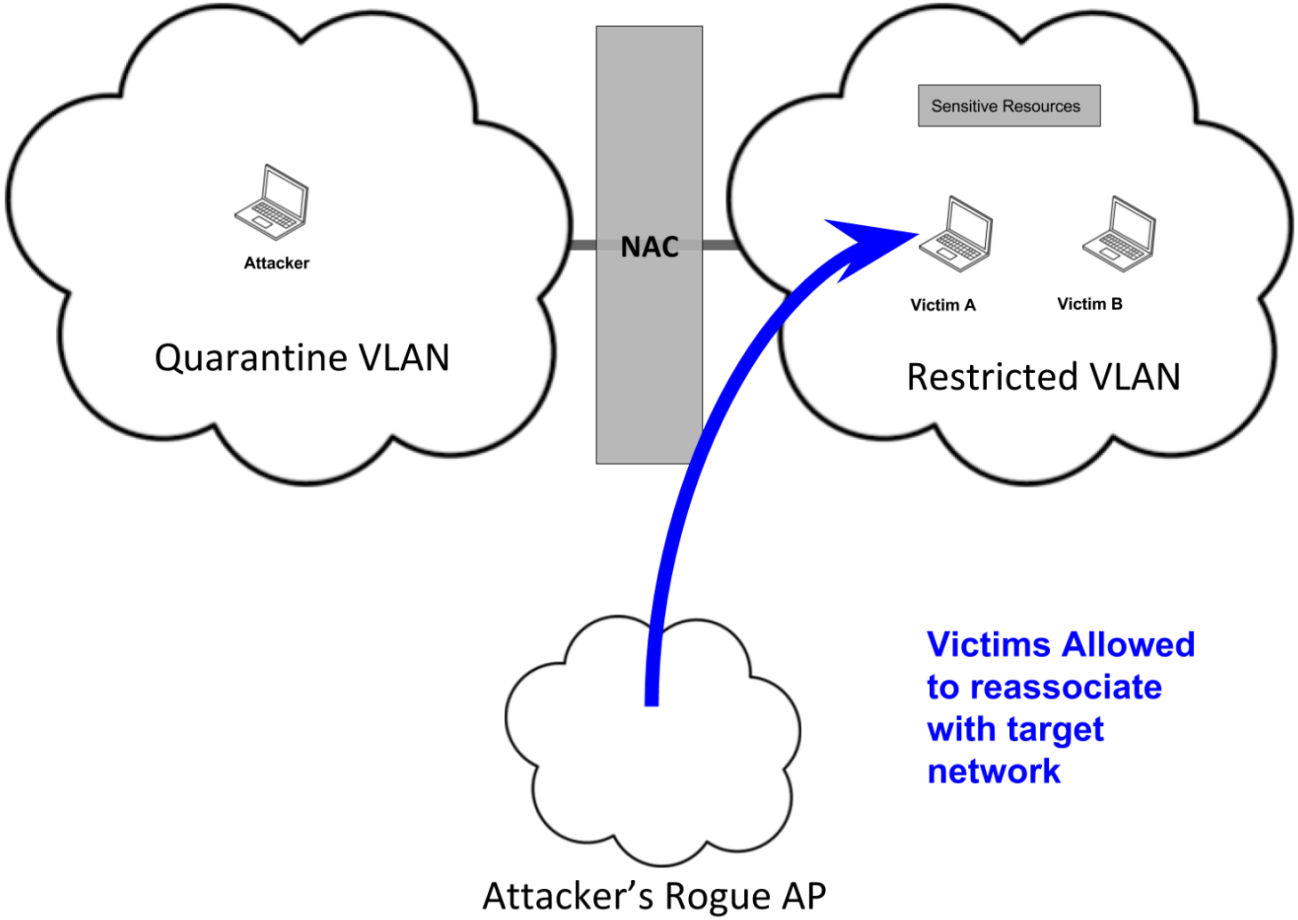
Indirect Wireless Pivots

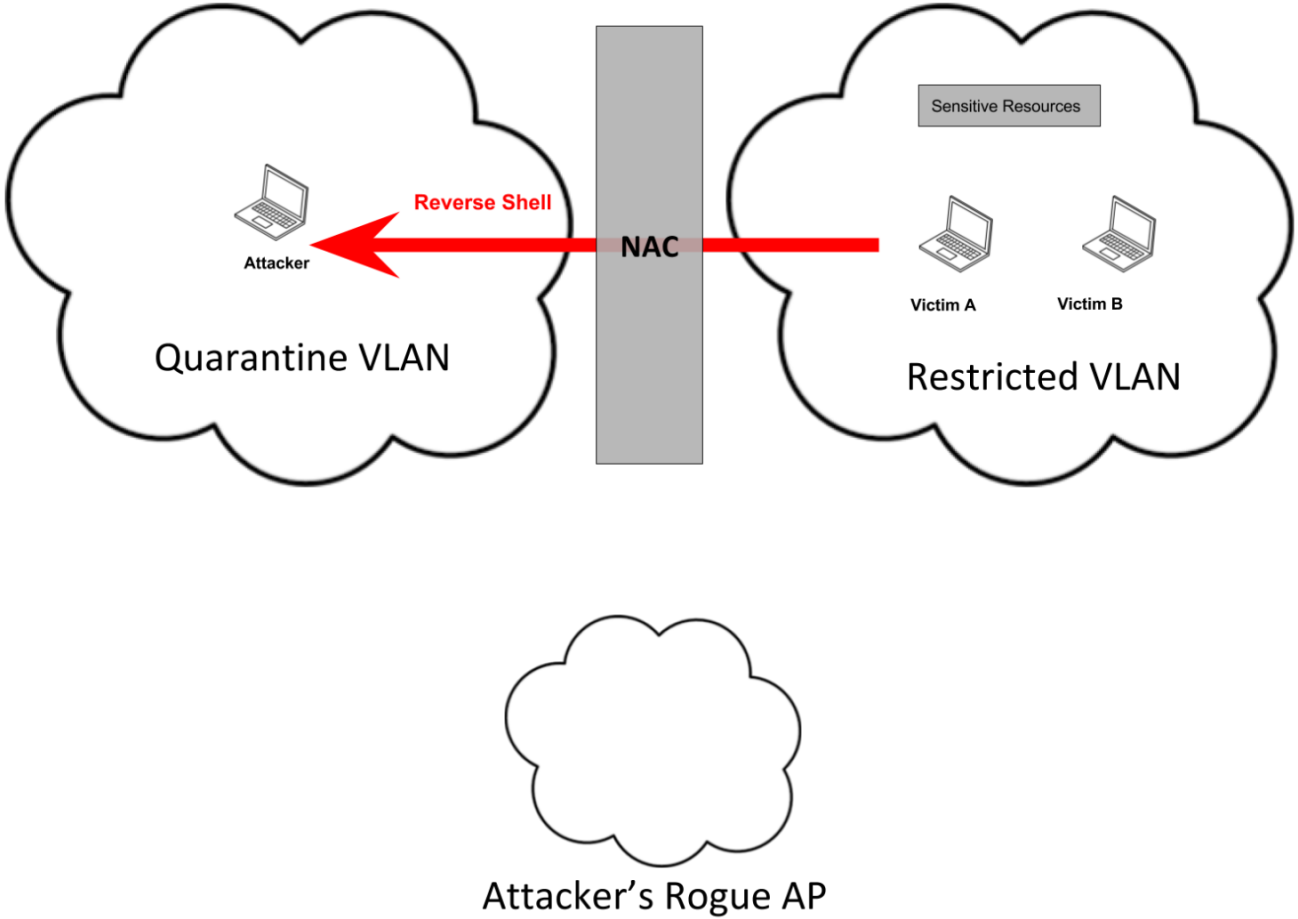
Use Evil Twin attacks to
bypass port-based access control mechanisms


Attacker
Quarantine VLAN
NAC
Sensitive Resources
Victim A
Victim B
Restricted VLAN


Attacker
Quarantine VLAN
NAC
Sensitive Resources
Restricted VLAN
Victim A
Victim B
Attacker's Rogue AP
Evil Twin forces victims
to associate with
attacker's hostile portal


Quarantine VLAN
NAC
Sensitive Resources
Restricted VLAN
SMB Relay From
Victim B to Victim A.
Timed payload
placed on Victim A.
Victim A
Victim B
Attacker's Rogue AP


Attacker
Quarantine VLAN
NAC
Sensitive Resources
Victim A
Victim B
Restricted VLAN
Victims Allowed to reassociate with target network
Attacker's Rogue AP


Sensitive Resources
Reverse Shell
NAC
Attacker
Victim A
Victim B
Quarantine VLAN
Restricted VLAN
Attacker's Rogue AP

Lab 8: Hostile Portal Attacks and Indirect Wireless Pivots