The Black Art of Wireless Post-Exploitation:

Bypassing Port-Based Access Controls Using Indirect Wireless Pivots

GreHacks

Gabriel Ryan (solstice)

net user author /domain

Gabriel Ryan

Researcher @ Gotham Digital Science

World’s best Red Team you’ve never heard of ;D

@s0lst1c3

gryan@gdssecurity.com

New in this presentation:

Hostile Portal Attacks:

- Steal Active Directory creds from WPA2-EAP networks ***without network access***

Indirect Wireless Pivots:

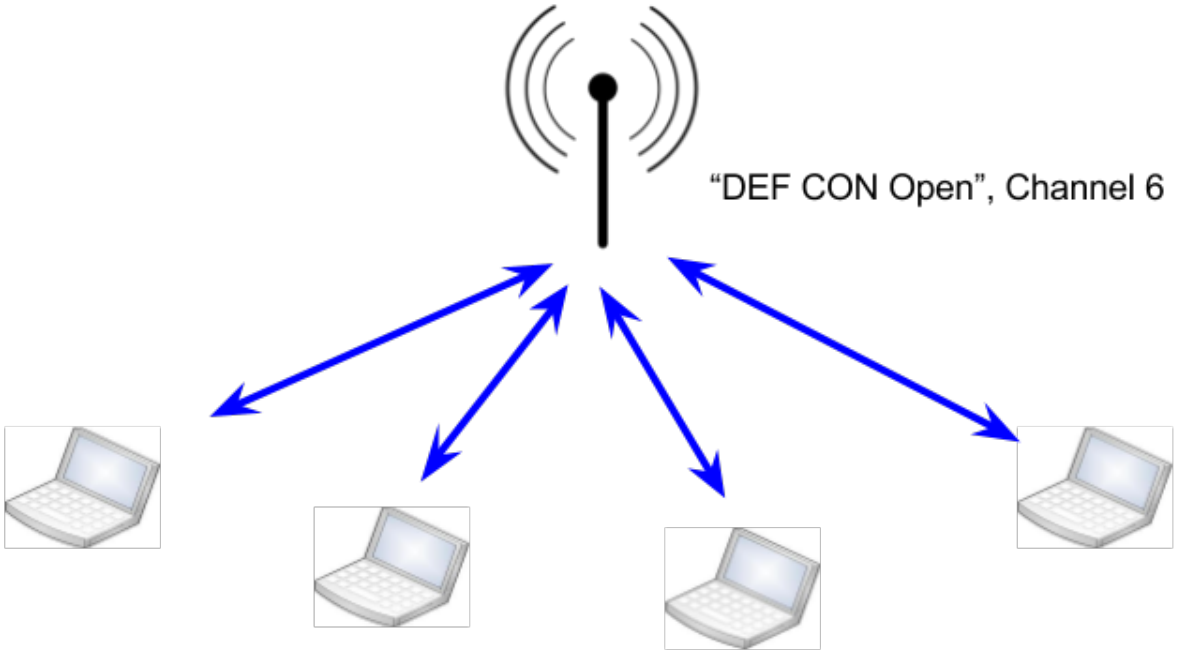
- Use Rogue AP attacks to ***bypass port-based access control mechanisms***

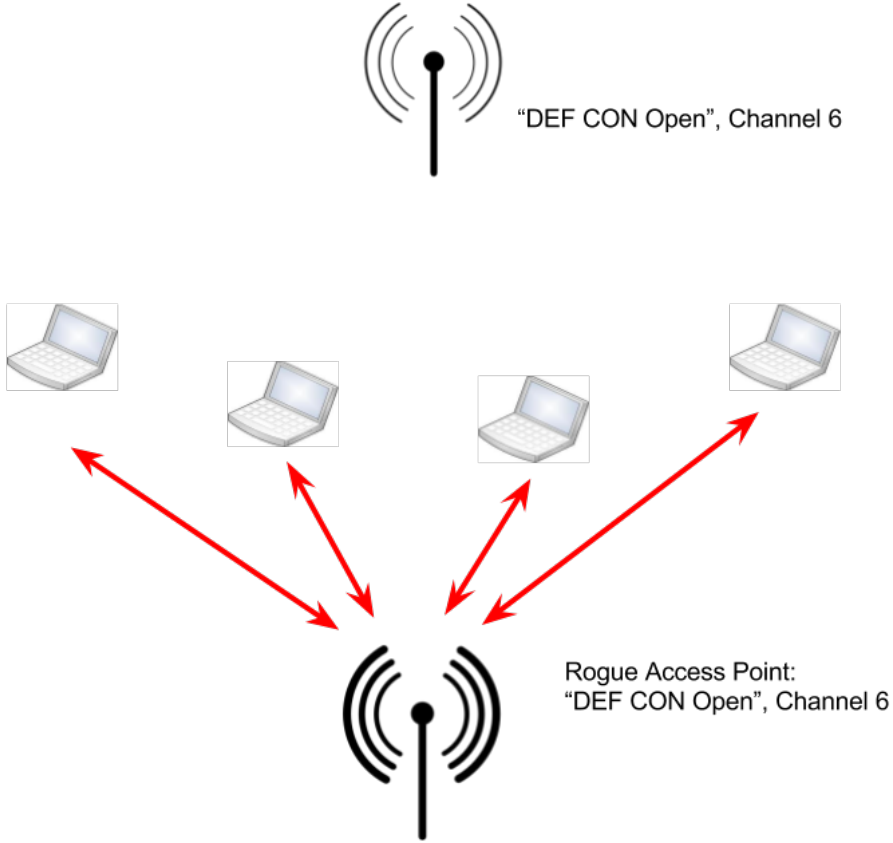
WPA2-EAP

Wireless Theory: Evil Twin Attacks

Rogue access point attacks:

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


“DEF CON Open”, Channel 6


“DEF CON Open”, Channel 6
Rogue Access Point:
“DEF CON Open”, Channel 6

Evolution of Rogue Access Point Attacks

2002 - Evil Twin attacks documented in “Wireless LAN Security FAQ” - C. W. Klaus [1]

2003 – asleap - Joshua Wright [2]

2004 - Karma Attacks - Dino Dai Zovi and Shane Macaulay [3]

2008 - Freeradius-wpe - Joshua Wright and Brad Antoniewitz [4]

2014 - Improved Karma Attacks (Mana) - Dominic White and Ian de Villiers [5]

2017 – Lure10 Attacks – George Chatzisofroniou [30]

Evolution of Rogue Access Point Attacks

Rogue AP attacks primarily used to fill two roles:

1. MITM attacks (stealing creds)
2. Breaching WPA/WPA2 networks (gaining access to WLAN)

In this talk: rogue AP attacks as a means of ***lateral movement***.

Evil Twin Attacks Against WPA2-EAP

WPA2-EAP

Logically:

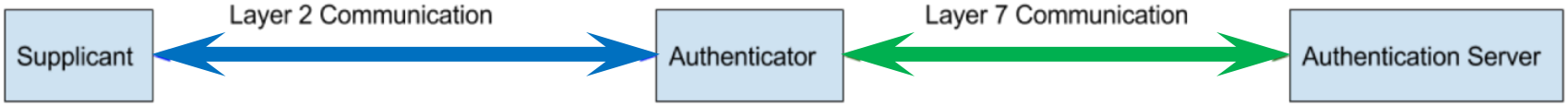
- Authentication occurs between supplicant and authentication server [6][7][8]

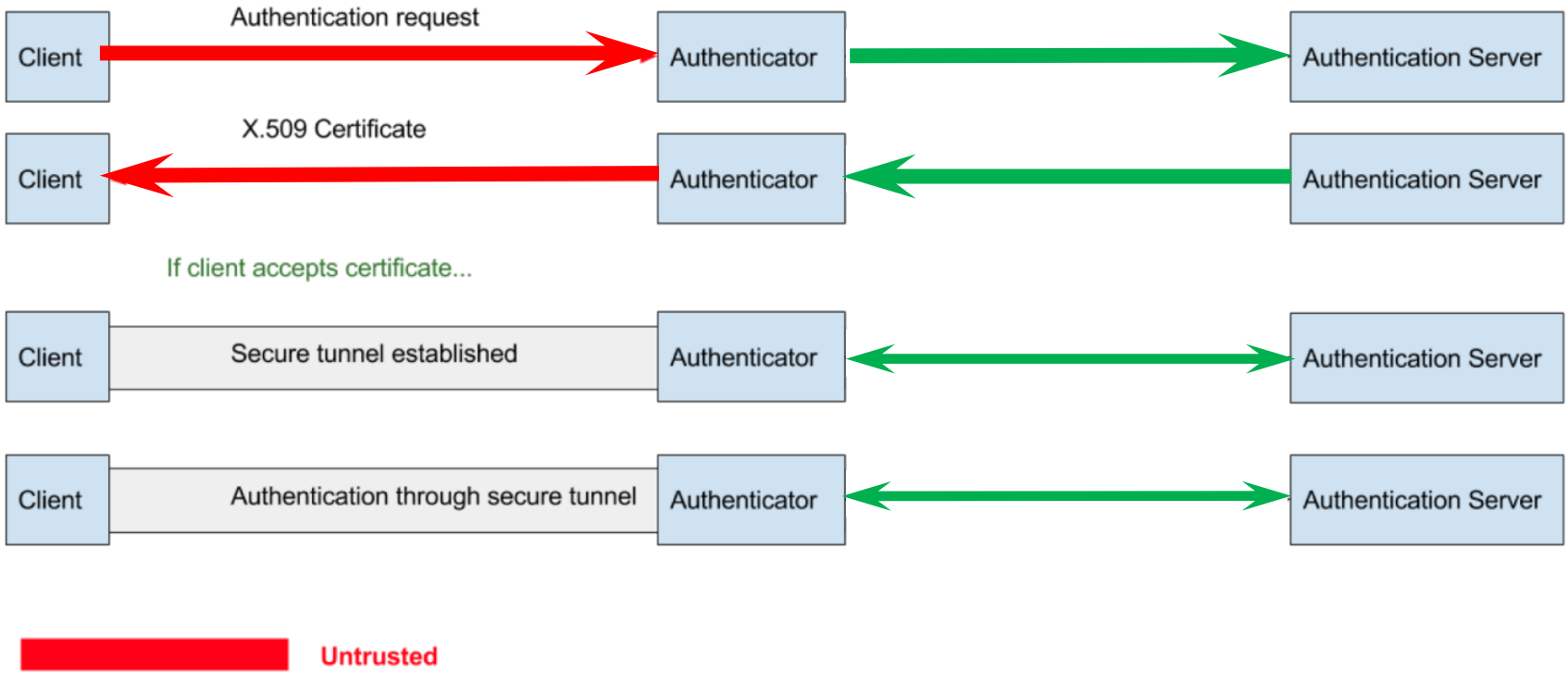

Client
Authentication request
Authentication Server
Client
X.509 Certificate
Authentication Server
If client accepts certificate...
Client
Secure tunnel established
Authentication Server
Client
Authentication through secure tunnel
Authentication Server

Wireless Theory: EAP

Without secure tunnel, auth process can be sniffed:

- Attacker sniffs challenge and response then derives password offline
- Legacy implementations of EAP susceptible to this (i.e. EAP-MD5… eapmd5hash by Joshua Wright in 2008 [13])


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

The attack:

- freeradius-wpe by Brad Antoniewicz in 2008 [4]
- Force supplicant to authenticate with attacker using evil twin attack [4]

Cracking MS-CHAPv2

Dictionary Attack:

- success rate inversely proportional to the strength of the password [31]

Cracking MS-CHAPv2:

Divide and Conquer Attack (Moxie Marlinspike and David Hulton, 2012):

- MS-CHAPv2 uses same 56-bit DES encryption as NTLMv1 [31] [32]
- Security reducible to the strength of a single DES encryption [31] [32]
- Goal: recover NT hash rather than plaintext password [31]
- 100% success rate in less than 24 hours when using an FPGA cracking rig such as Crack.sh (previously Cloudcracker) [33]

DEMO

Solution: EAP-TLS

- Introduced in 2008 (wow!) by RFC 5216 [10]
- Mutual authentication using x.509 certifications a requirement for most implementations [10]
- Strength lies in the use of client-side certificates

Poor adoption rate:

- Wildly unpopular [11]
- Client-side certs make EAP-TLS seem considerably more difficult to integrate into existing network architecture (more on this later)
- Classic security vs. convenience scenario

Security vs. Convenience

Network administrators forced to choose between:

- authentication mechanisms with known weaknesses

OR

- a highly secure yet seemingly impractical authentication mechanism

Market Gap

Market gap created for products that meet the following requirements:

- can be used to compensate for the security issues found in EAP-PEAP/EAP-TTLS
- are easy to use

The “solution”:

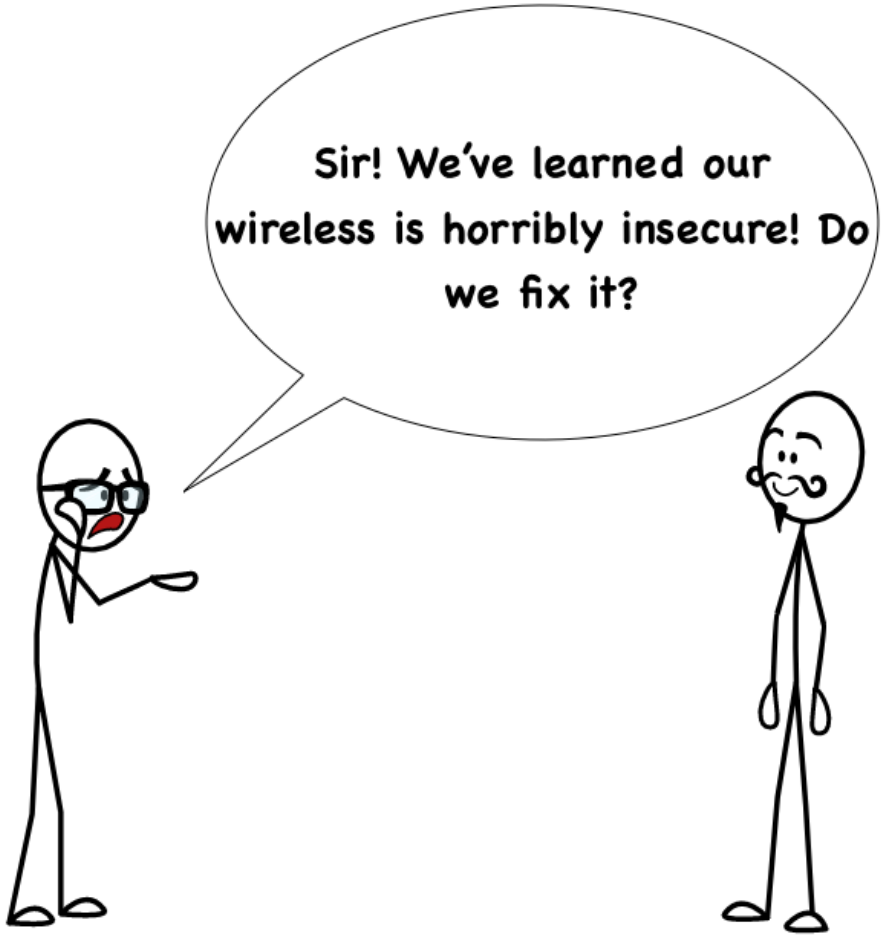
The current trend:

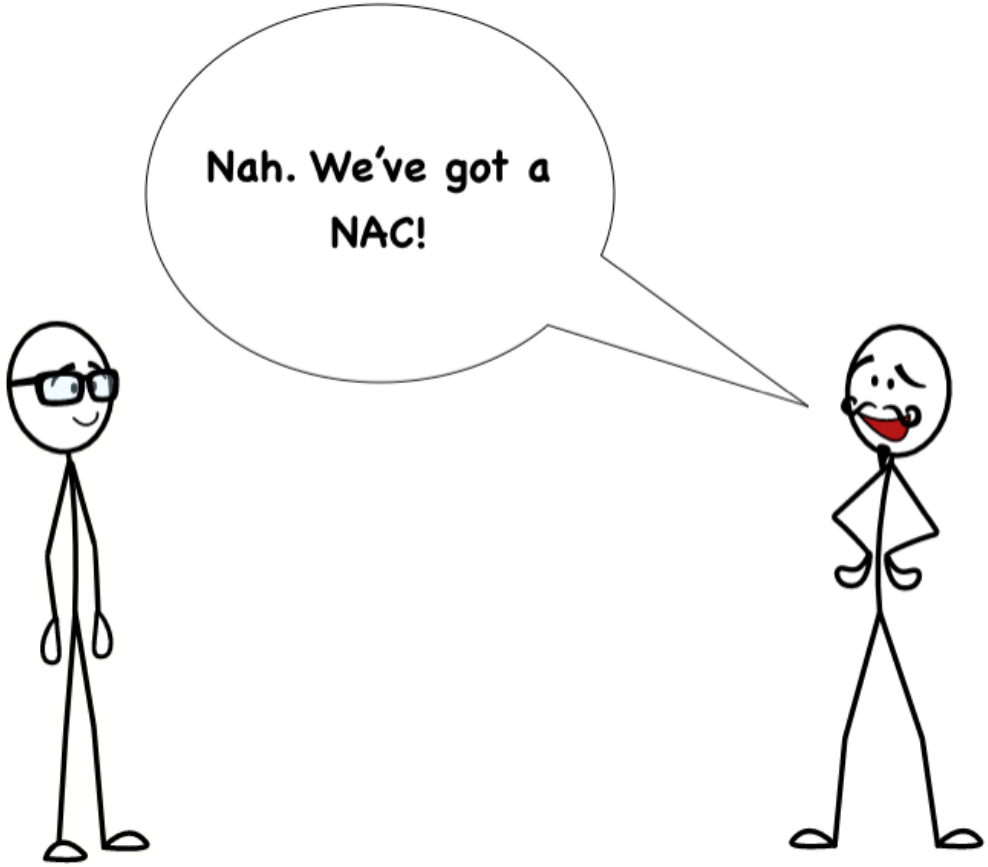
- Focus on breach ***containment***, rather than breach ***prevention***

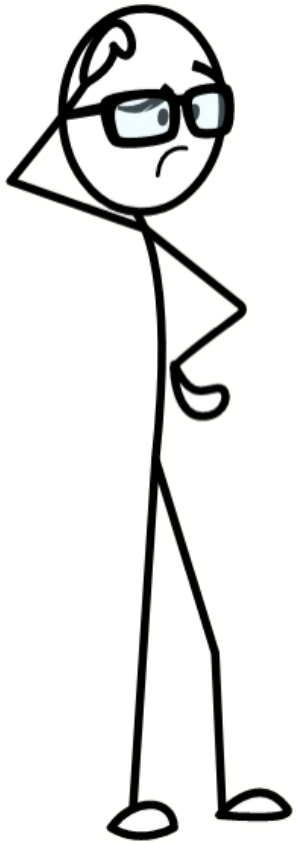
Containment vs. Prevention

Does this actually work?

Classic WLAN Access Control Mechanisms


Sir! We've learned our wireless is horribly insecure! Do we fix it?


Nah. We've got a NAC!



Using NACs For WLAN Breach Containment

Network Access Control (NAC) Mechanisms:

- One of the most popular methods of containing wireless breaches
- Distinguish between authorized and unauthorized network endpoints [12]

Using NACs For WLAN Breach Containment

1. New endpoint is added to the wireless network
2. NAC identifies whether new endpoint is authorized or unauthorized device
3. If unauthorized, placed in quarantine VLAN

Two varieties of NAC:

- Agent-based [12]
- Agentless [12]

Agent-based NACs:

- Software component installed on authorized endpoints [12]
- Agents communicate with “brain” of NAC [12]
- Highly effective
- Nearly as impractical as EAP-TLS

Agentless NACs:

- Passive fingerprinting [12]
- Active scanning [12]
- Easier to deploy than agent-based NACs [12]
- Unable to examine internals of network components [12]
- Can be bypassed by masquerading as an authorized device [12]

Recurring dilemma: insecurity vs. impracticality

Yet another market gap:

High demand for a solution that offers the deep interrogation capabilities of an agent-based NAC, but without the additional overhead. [13]

Next Generation NACs: The Best Of Both Worlds?


PLEASE SIR, MAY I BORROW ONE?


YOUR
REQUEST
IS DENIED.


YOU WANT TO DO WHAT??!

[VENDOR REDACTED]

- Uses WMI to interrogate new devices [14]
- Capable of performing internal checks *without* the use of an agent

[VENDOR REDACTED]

- Authenticates over SMB using a single administrative service account [14]
- Service account given remote login privileges to all authorized devices at the Group Policy level [14]
- Allows [NOPE] to perform deep interrogations without the use of an agent [14]

Single Point of Failure

- Attempts to authenticate with any new endpoint placed on the network using special service account [14]
- Service account has access to nearly everything on the network

… i.e. - Godmode hashes sent to any new device that is added to the network.

Risks: SMB Relay Attacks

- SMB signing disabled by default on everything but the domain controller (Group Policy is downloaded over SMB) [15]
- No MITM required: the NAC appliance is trying to authentication with *you*

SMB Signing

- The SMB Relay issue can be mitigated by digitally signing packets
- SMB Signing: digitally signing packets to confirm their authenticity
- Does *not* address the issue of hashes being sent directly to untrusted endpoints

[VENDOR REDACTED]

- Can be installed to remediate this issue
- Is essentially a form of agent
- [NOPE] chief selling point is that no agent is required

No magic bullet

- “Security ***With*** Convenience” – this is a paradox

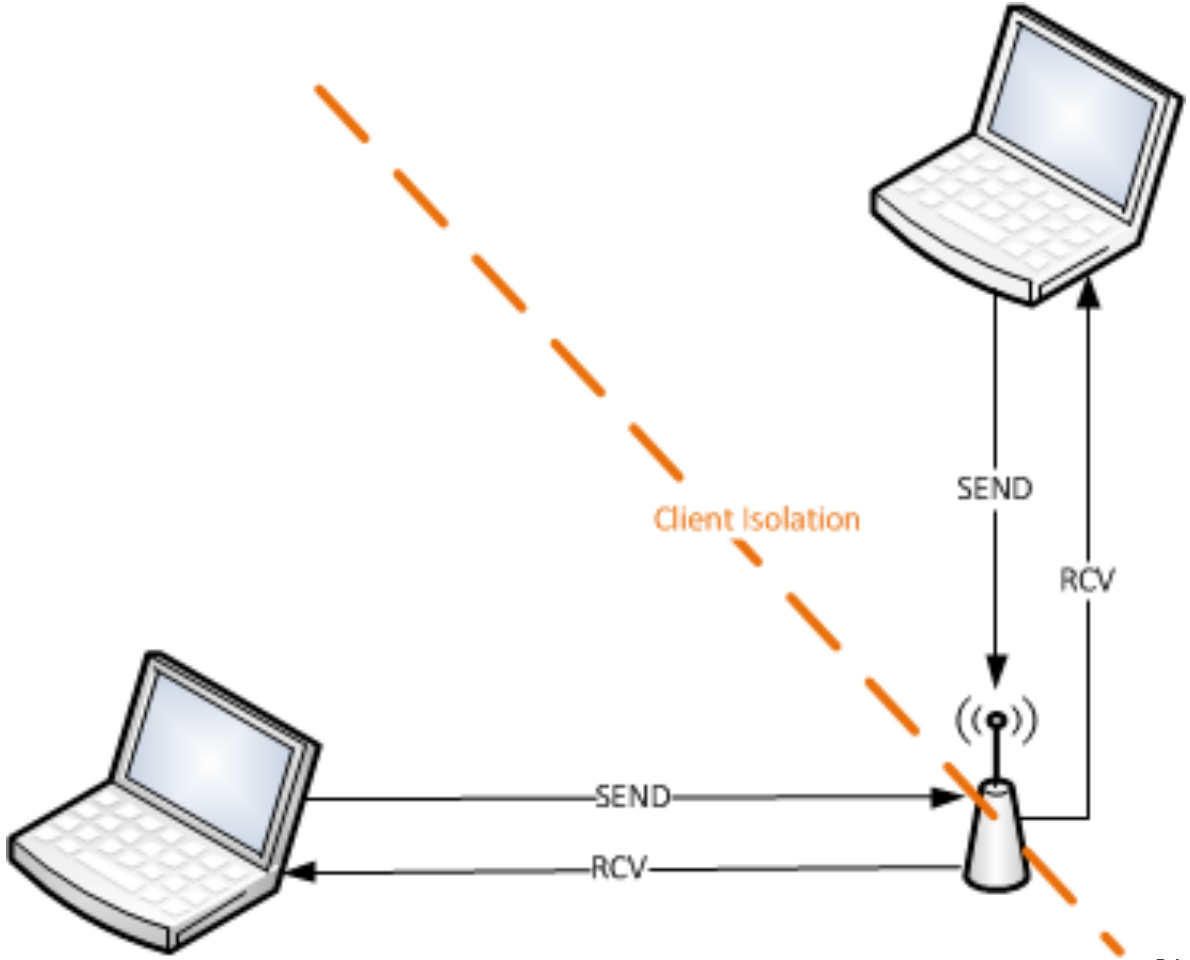
What about Client Isolation?

Wireless Client Isolation

- Prevents wireless clients from communicating with each other
- Often used as a security control
- Typical use case: open networks [16]

How 802.11 Is Supposed To Work:

- AP mediates all communication on network [16]
- In theory, client isolation would work [16]



[16]

The Problem:

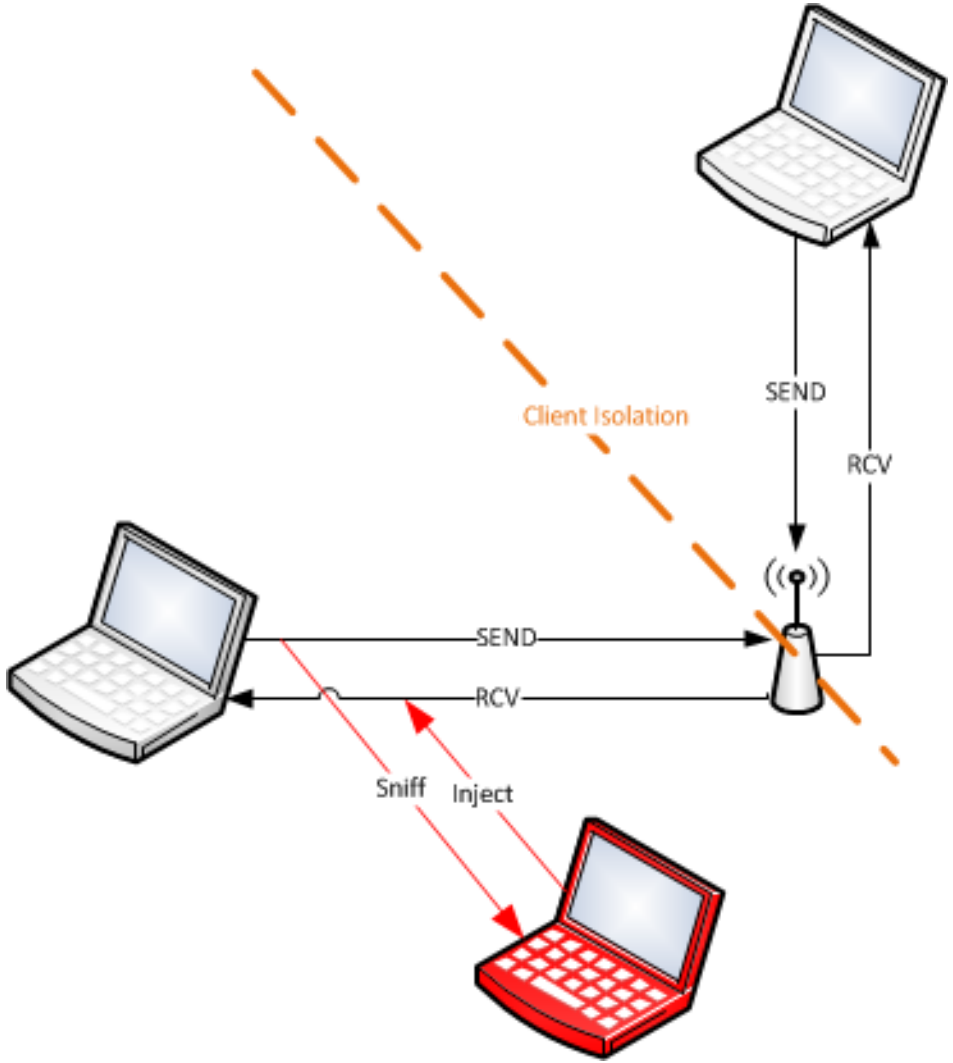
- Client isolation is a ***logical*** control, not a physical control
- The problem: “how do you prevent radio transceivers from communicating with one another?” [16]
- Cedric Blancher in 2005: You can’t. [17]

Introducing Wifitap:

- First released by the late Cedric Blancher in 2005 [17]
- Revived by Oliver Lavery of Gotham Digital Science in 2013 [16]

Introducing Wifitap:

- Reads packets from victim to AP using WiFi interface in monitor mode [16]
- Injects responses to those packets as if they came from the AP [16]


Client Isolation
SEND
RCV
SEND
RCV
Sniff
Inject

Introducing Wifitap: how it works

- Bridges a Linux tun/tap device with a WiFi interface in monitor mode [16]
- To interact with network, you interact with the tun/tap interface [16]
- Allows you to communicate directly with wireless clients *without* associating with the AP [16]

Later tools (that do even more stuff):

Aircrack Suite:

- airtun-ng (supports WEP) [18]
- tkiptun-ng (supports WPA1) [19]

Theoretical Attacks:

Considerable debate as to whether these actually work. Worth mentioning for the lulz.

- Hole 196 [16]

DEMO

Food for thought

What if we’re missing the point?

NAC Isn’t The Only Problem

The role of NAC in containing WLAN breaches:

- Used to prevent attackers from accessing sensitive resources ***after*** breach occurs

NAC Isn’t The Only Problem

When an unauthorized endpoint is detected, one of two actions is typically taken:

- Endpoint is placed in quarantine
- Port is blocked

The role of NAC in a wireless environment:

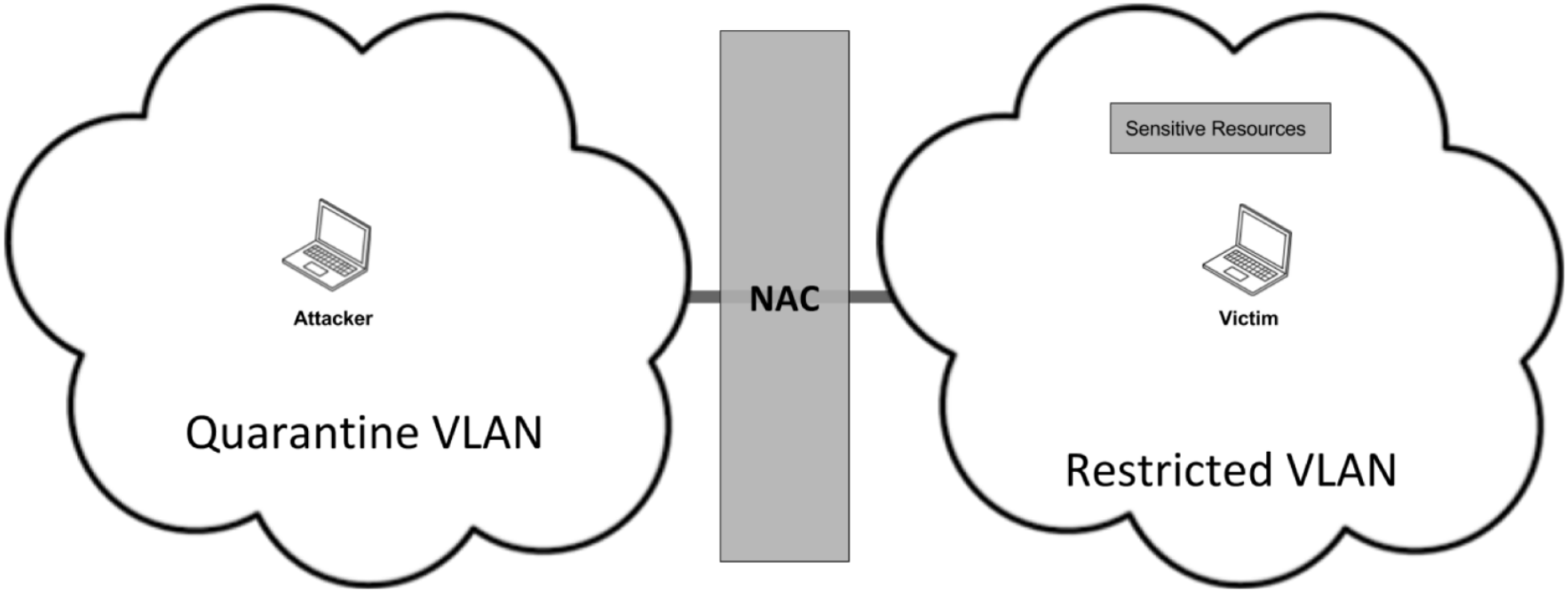
Violating access control policies causes the NAC to impose a *restriction*:

- In a wired network, this is a ***physical restriction***
- In a wireless network, this can ***only*** be a ***logical restriction***

More on this later…

The Scenario

- We are attacking a WLAN that is used to access sensitive resources
- We have already breached the perimeter


Attacker
NAC
Sensitive Resources
Victim
Quarantine VLAN
Restricted VLAN

How do we get out?

Review: LLMNR/NBT-NS Poisoning

LLMNR/NBT-NS Poisoning

NetBIOS name resolution [20][21]:

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 [22]:

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

LLMNR/NBT-NS Poisoning

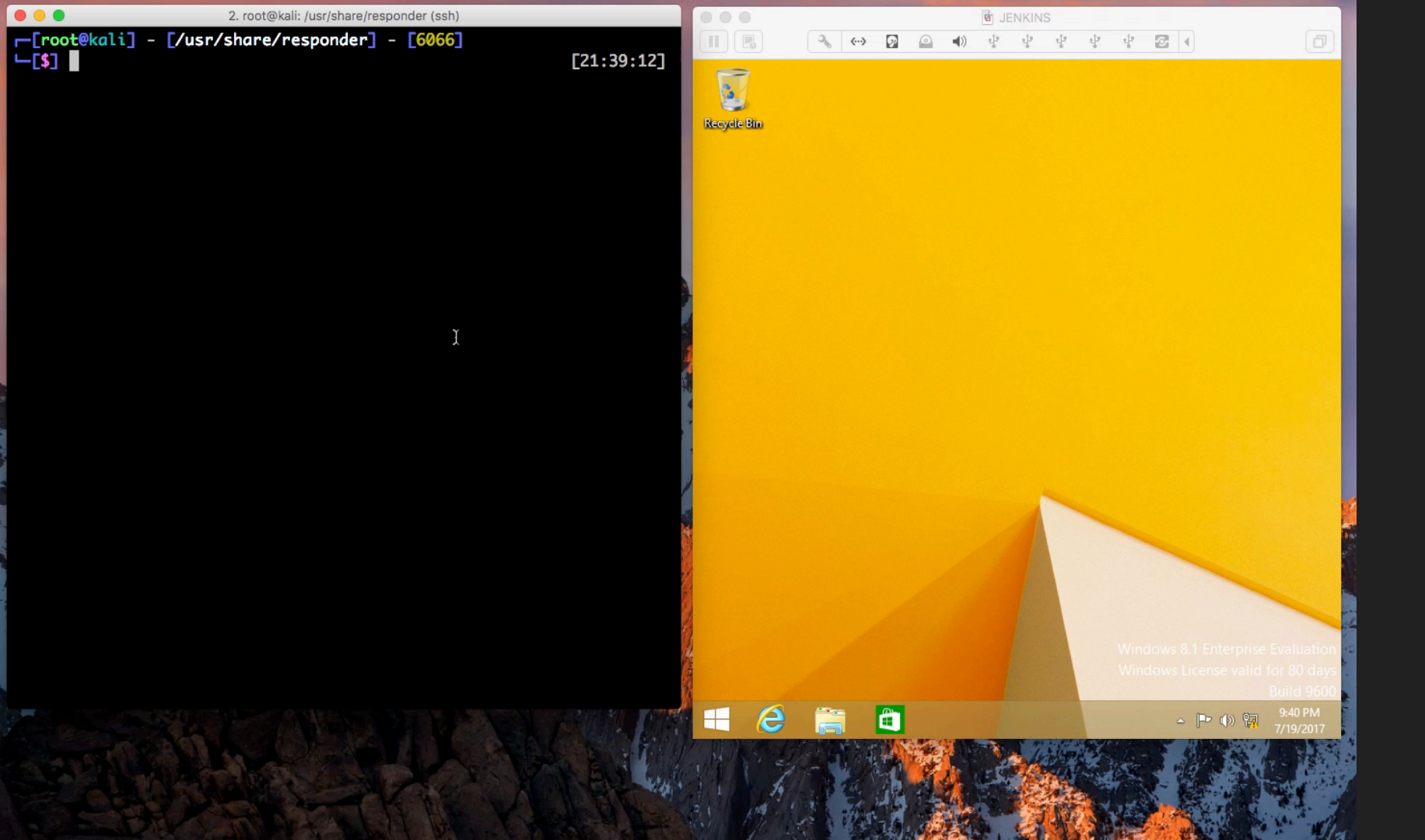
Two Windows computers named Alice and Leeroy [23]:

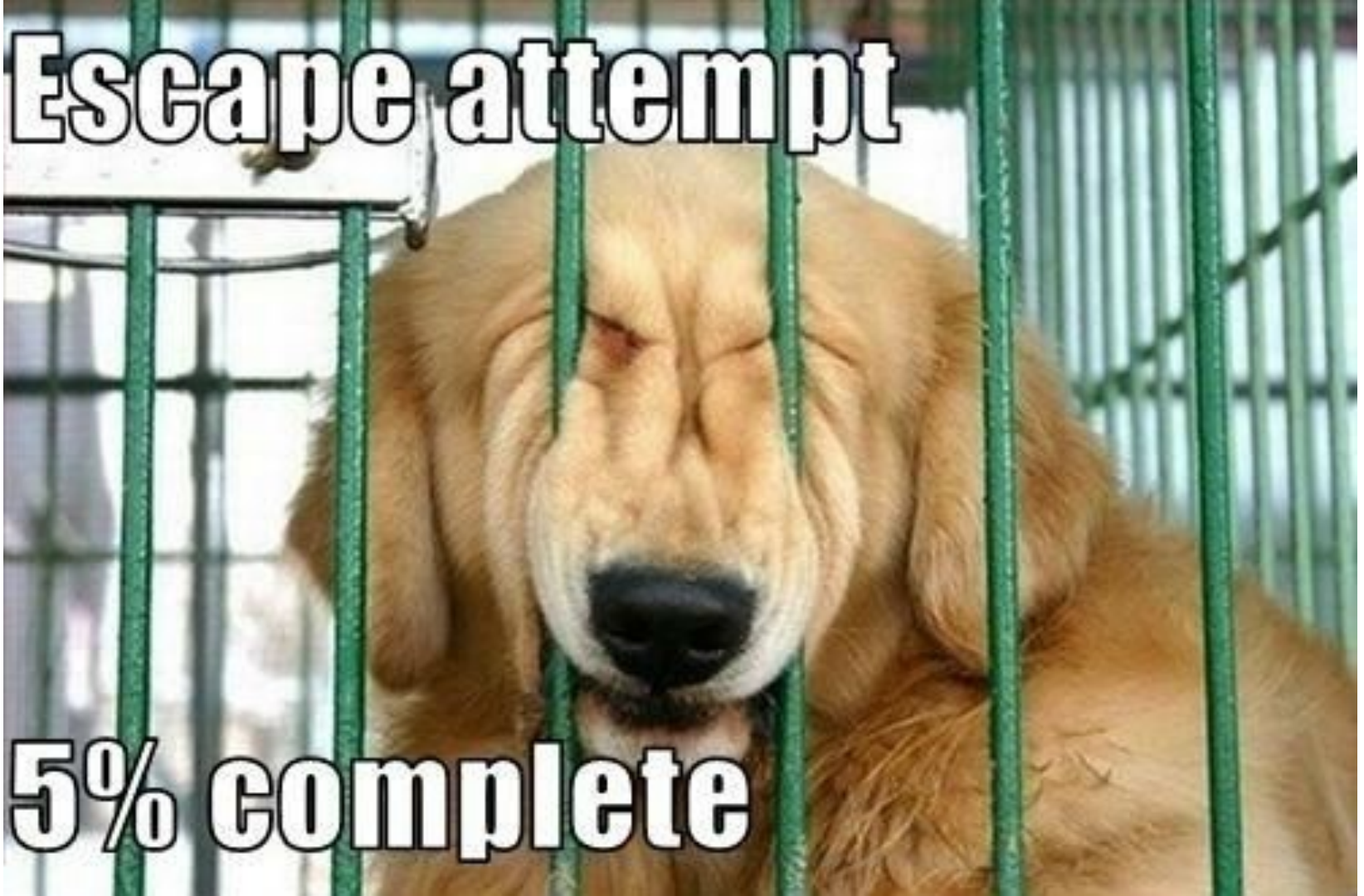
1. Alice wants to request file from Leeroy, but does not know Leeroy's IP
2. Alice attempts to resolve Leeroy's name locally and using DNS, but fails
3. Alice makes broadcast requests using LLMNR/NBT-NS
4. Every computer on Alice's subnet receives request
5. Honor system: only Leeroy responds

LLMNR/NBT-NS Poisoning

No honor among thieves [23]:

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


2. root@kali: /usr/share/responder (ssh)
┌─[root@kali] - [/usr/share/responder] - [6066]
└─[$]
[21:39:12]
JENKINS
Recycle Bin
Windows 8.1 Enterprise Evaluation
Windows License valid for 80 days
Build 9600
9:40 PM
7/19/2017


Escape attempt
5% complete

Review: Redirect to SMB

Redirect to SMB

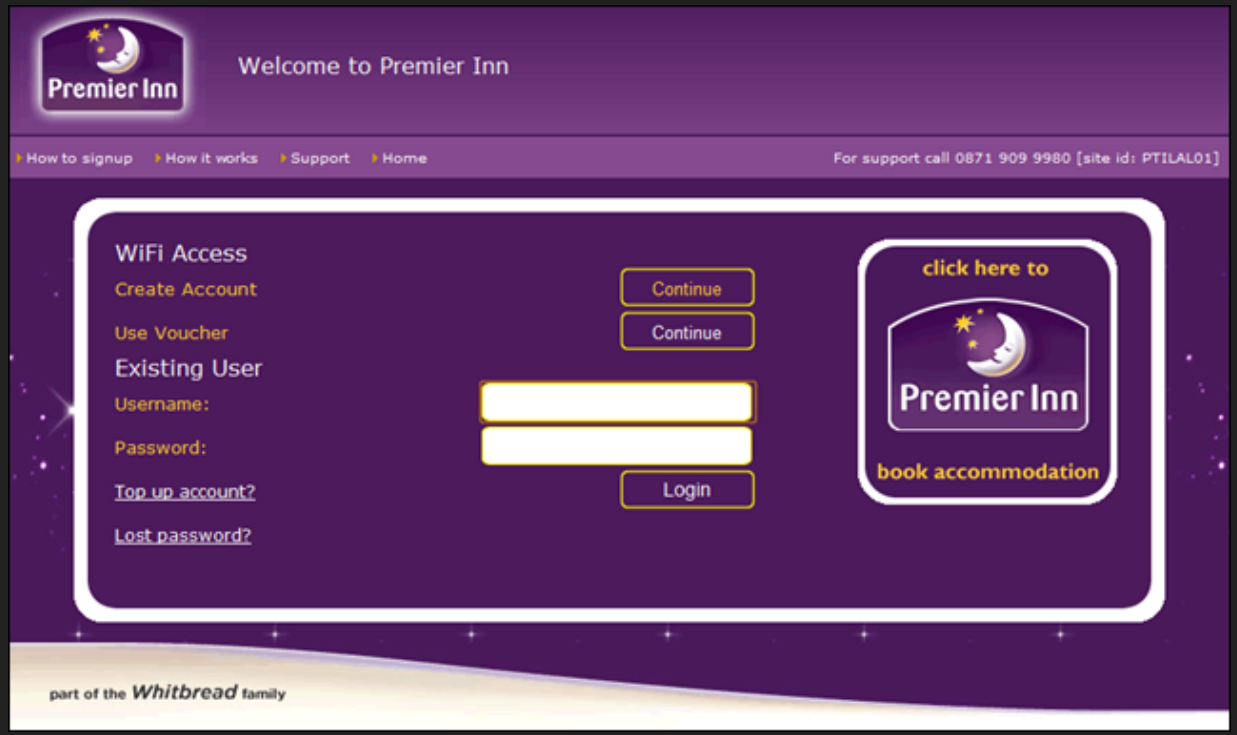
- The idea is to force the victim to visit an HTTP endpoint that redirects to an SMB share on attacker's machine, triggering NTLM authentication
- Variation: redirect to non-existent SMB share, triggering LLMNR/NBT-NS [24]
- Fast way to get hashes
- Requires social engineering

Hostile Portal Attacks

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

Captive Portal

- Used to “restrict” access to an open WiFi-network



Captive Portal

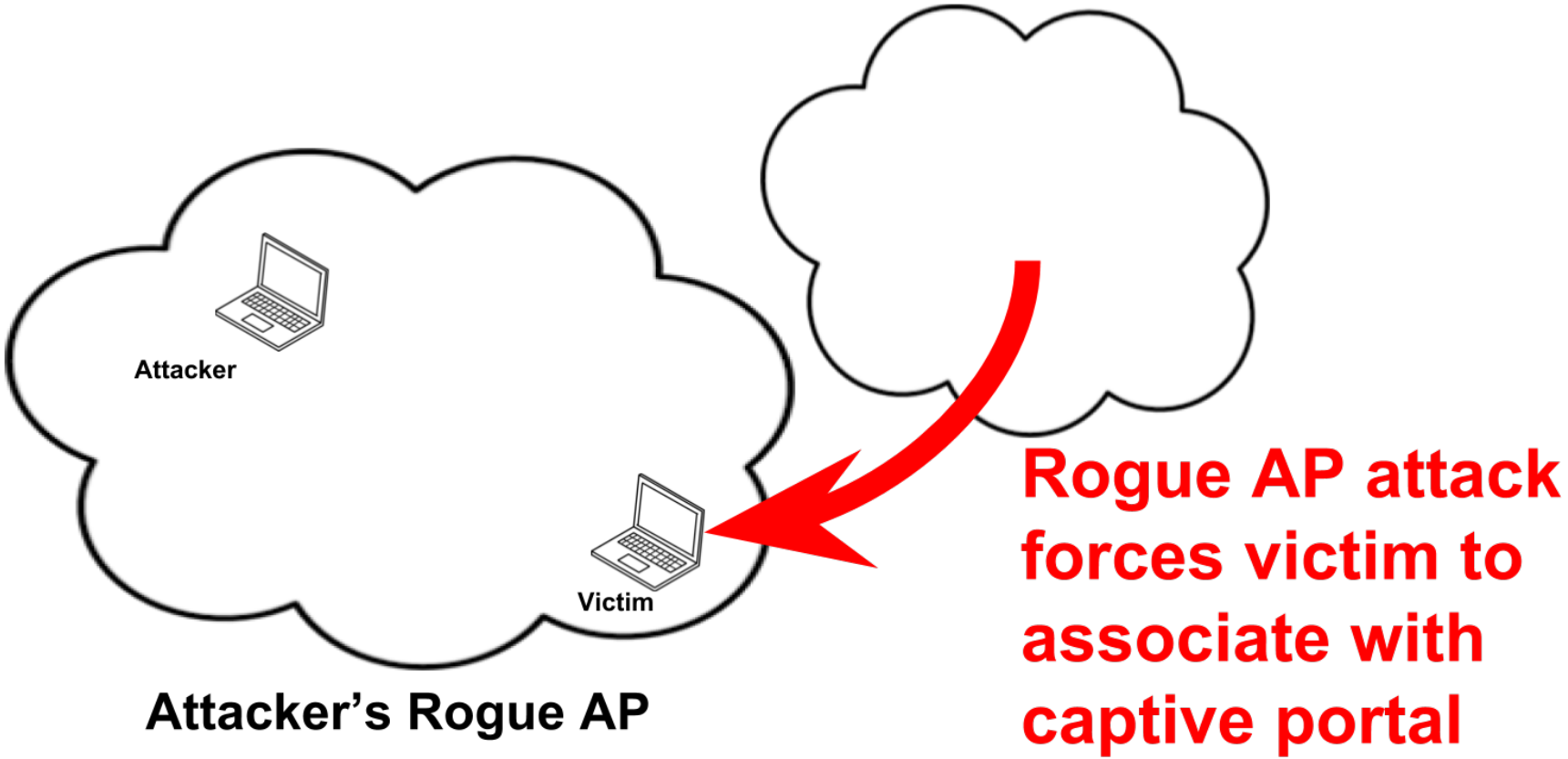
- All DNS queries resolved to captive portal
- All DNS traffic redirected to captive portal (optional)
- All HTTP traffic redirected to captive portal (optional)

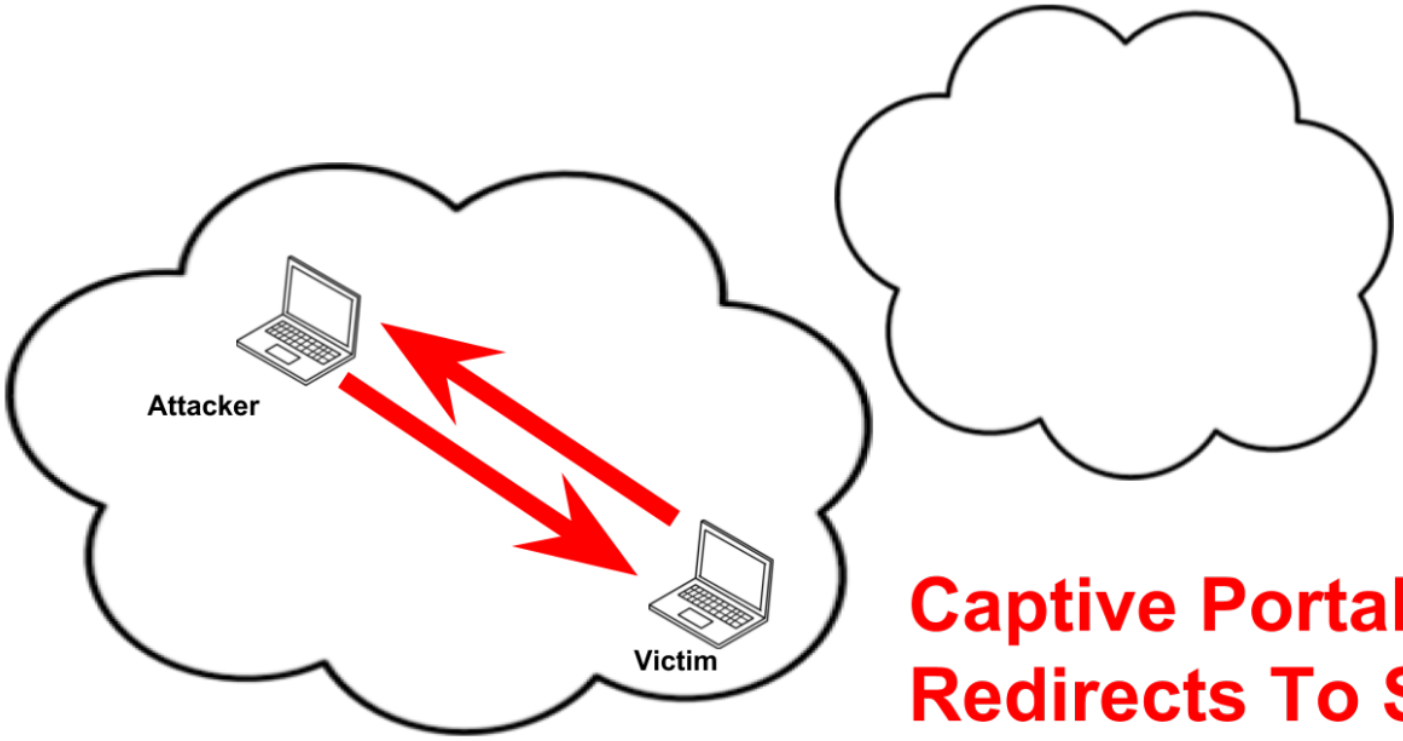
Hostile Portal Attack

- Based on Redirect to SMB Attack
- Victim forced to connect to attacker using Rogue AP attack
- All HTTP traffic redirected to SMB share on attacker’s machine instead of a captive portal attack
- All LLMNR/NBT-NS lookups are poisoned

Attacker

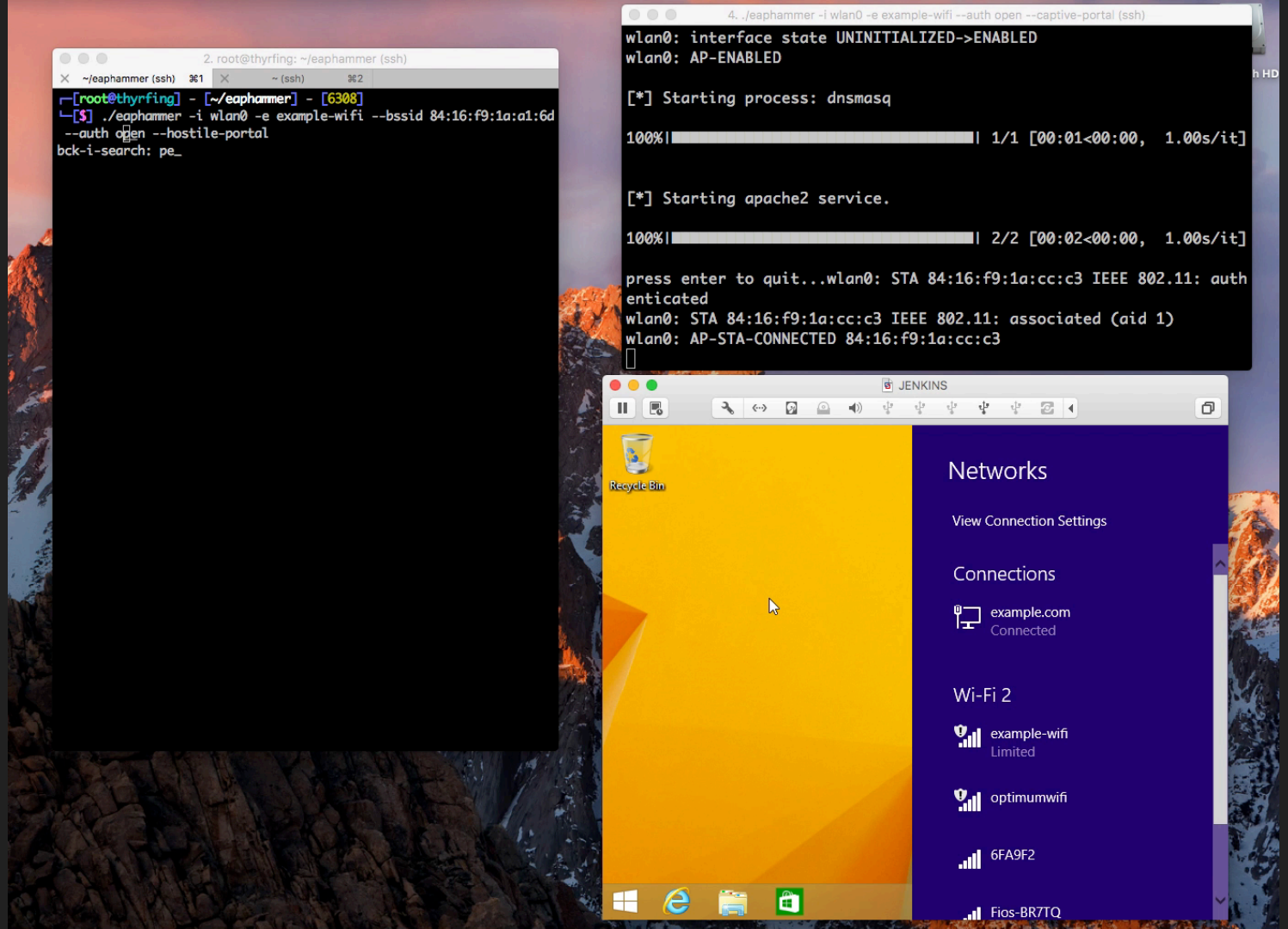
Victim


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



Attacker's Rogue AP

Captive Portal Redirects To SMB


2. root@thyrfing: ~/eaphammer (ssh)
~/eaphammer (ssh) ⌘1
~ (ssh) ⌘2
┌─[root@thyrfing] - [~/eaphammer] - [6308]
└─[$] ./eaphammer -i wlan0 -e example-wifi --bssid 84:16:f9:1a:a1:6d
--auth open --hostile-portal
bck-i-search: pe_
4. ./eaphammer -i wlan0 -e example-wifi --auth open --captive-portal (ssh)
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
[*] Starting process: dnsmasq
100%| | 1/1 [00:01<00:00, 1.00s/it]
[*] Starting apache2 service.
100%| | 2/2 [00:02<00:00, 1.00s/it]
press enter to quit...wlan0: STA 84:16:f9:1a:cc:c3 IEEE 802.11: auth
enticated
wlan0: STA 84:16:f9:1a:cc:c3 IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED 84:16:f9:1a:cc:c3
JENKINS
Recycle Bin
Networks
View Connection Settings
Connections
example.com
Connected
Wi-Fi 2
example-wifi
Limited
optimumwifi
6FA9F2
Fios-BR7TQ

WPA-EAP networks:

In most cases, this means EAP-TTLS or EAP-PEAP.

- Both use MS-CHAPv2 as the inner authentication method.
- Mutual authentication: the RADIUS server *must* prove knowledge of the supplicant's password for inner authentication to succeed [29]

WPA-EAP networks:

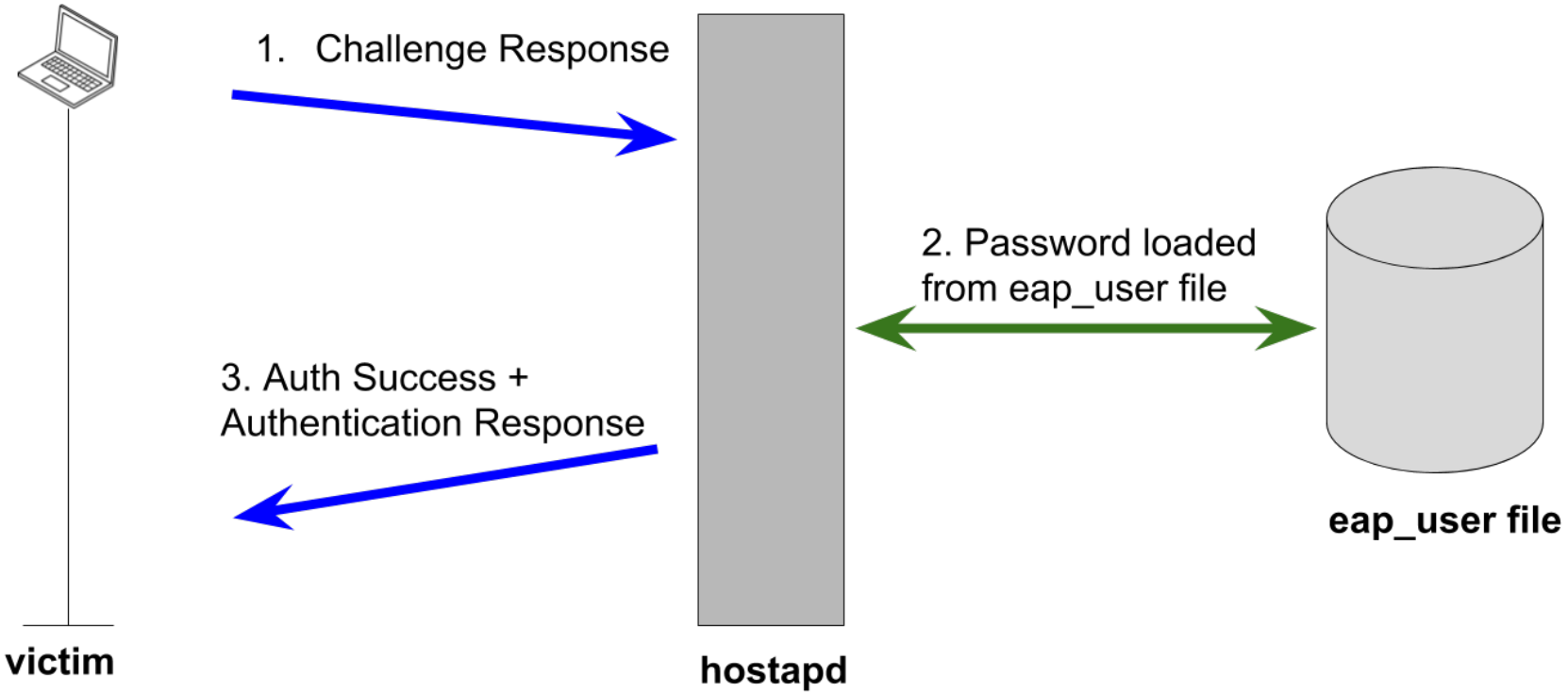
What this means:

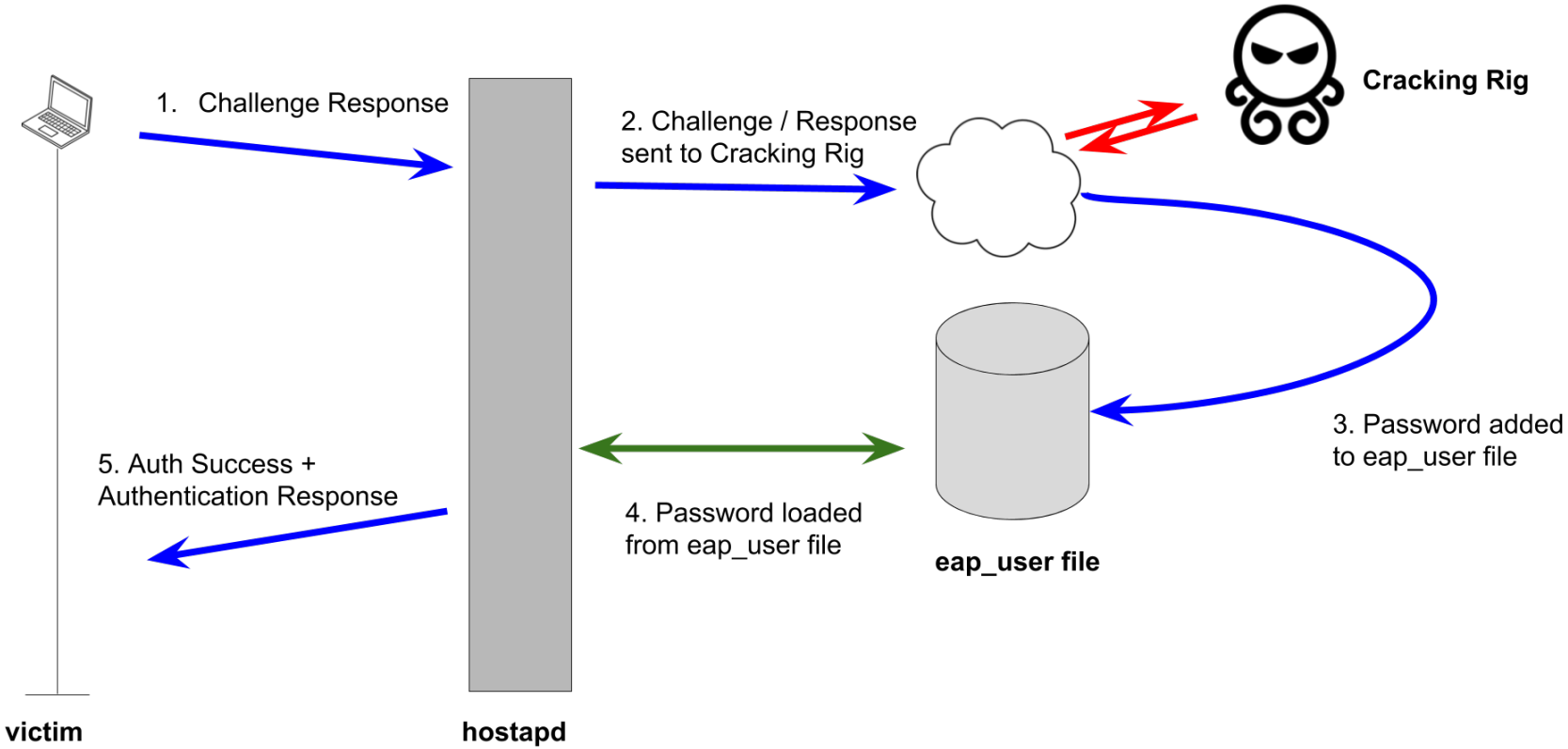
- Although the attacker can force the victim to authenticate with an evil twin to steal hashes, the attacker’s RADIUS server will fail the final the final stage of the authentication process and the client will not associate with the attacker [29].

Solution:

Crack credentials offline:

1. Weak RADIUS Passwords: Use auto crack ‘n add technique (Dominic White & Ian de Villiers in 2014)
2. Strong RADIUS Passwords: Crack offline, finish attack later

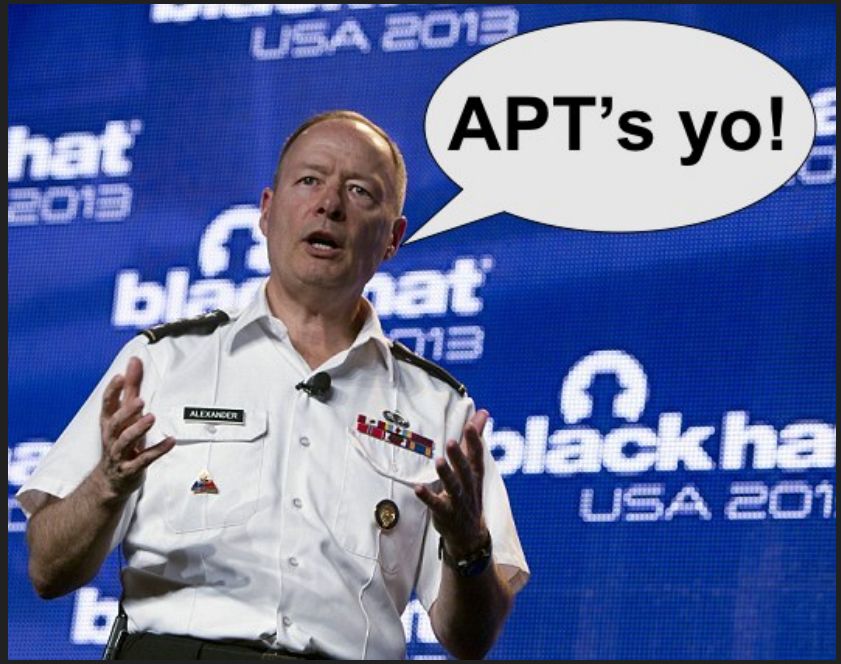

1. Challenge Response
2. Password loaded from eap_user file
3. Auth Success + Authentication Response
victim
hostapd
eap_user file

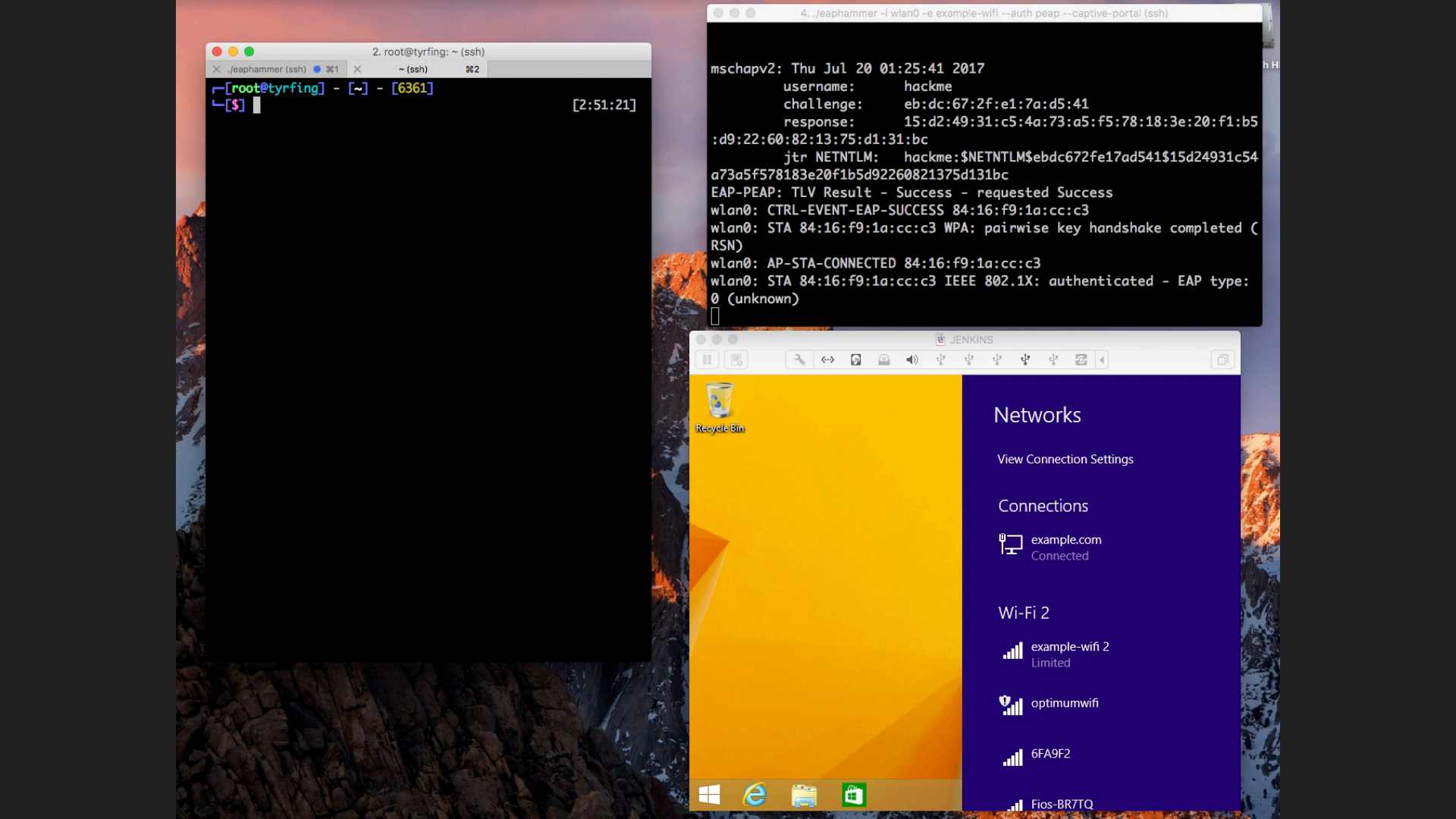

1. Challenge Response
2. Challenge / Response sent to Cracking Rig
Cracking Rig
3. Password added to eap_user file
eap_user file
4. Password loaded from eap_user file
5. Auth Success + Authentication Response
victim
hostapd

Second Option: Crack offline, Pwn later

No caveats other than time.

- Dictionary attack: lifecycle of the attack now takes place over the course of a week, rather than an hour.
- Divide and Conquer: 24 hours max when using FPGA based hardware, 100% success rate




2. root@tyrfing: ~ (ssh)
./eaphammer (ssh) ⌘1
~ (ssh) ⌘2
[root@tyrfing] - [~] - [6361]
[$]
[2:51:21]
4. ./eaphammer -i wlan0 -e example-wifi --auth peap --captive-portal (ssh)
mschapv2: Thu Jul 20 01:25:41 2017
username: hackme
challenge: eb:dc:67:2f:e1:7a:d5:41
response: 15:d2:49:31:c5:4a:73:a5:f5:78:18:3e:20:f1:b5
:d9:22:60:82:13:75:d1:31:bc
jtr NETNTLM: hackme:$NETNTLM$ebdc672fe17ad541$15d24931c54
a73a5f578183e20f1b5d92260821375d131bc
EAP-PEAP: TLV Result - Success - requested Success
wlan0: CTRL-EVENT-EAP-SUCCESS 84:16:f9:1a:cc:c3
wlan0: STA 84:16:f9:1a:cc:c3 WPA: pairwise key handshake completed (
RSN)
wlan0: AP-STA-CONNECTED 84:16:f9:1a:cc:c3
wlan0: STA 84:16:f9:1a:cc:c3 IEEE 802.1X: authenticated - EAP type:
0 (unknown)
JENKINS
Recycle Bin
Networks
View Connection Settings
Connections
example.com
Connected
Wi-Fi 2
example-wifi 2
Limited
optimumwifi
6FA9F2
Fios-BR7TQ

What this gets you: lots and lots of NTLM hashes

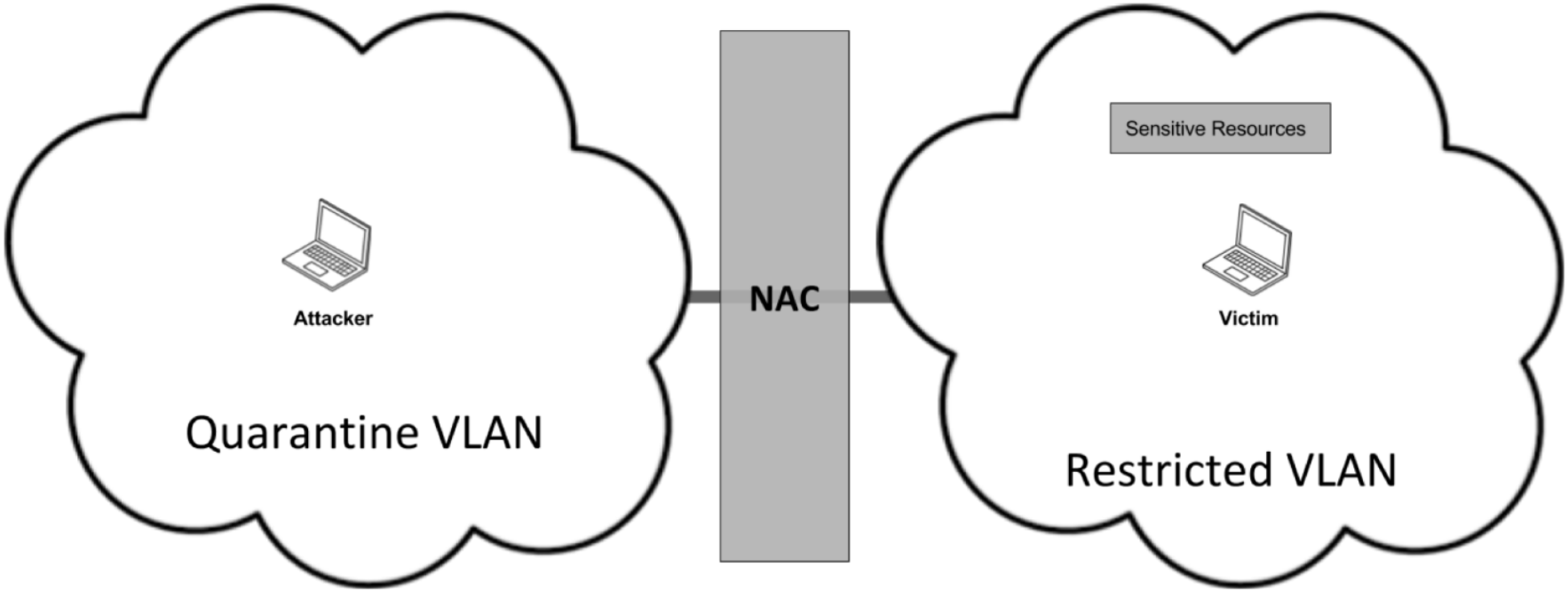
Similar results to LLMNR/NBT-NS poisoning, but with a few key advantages:

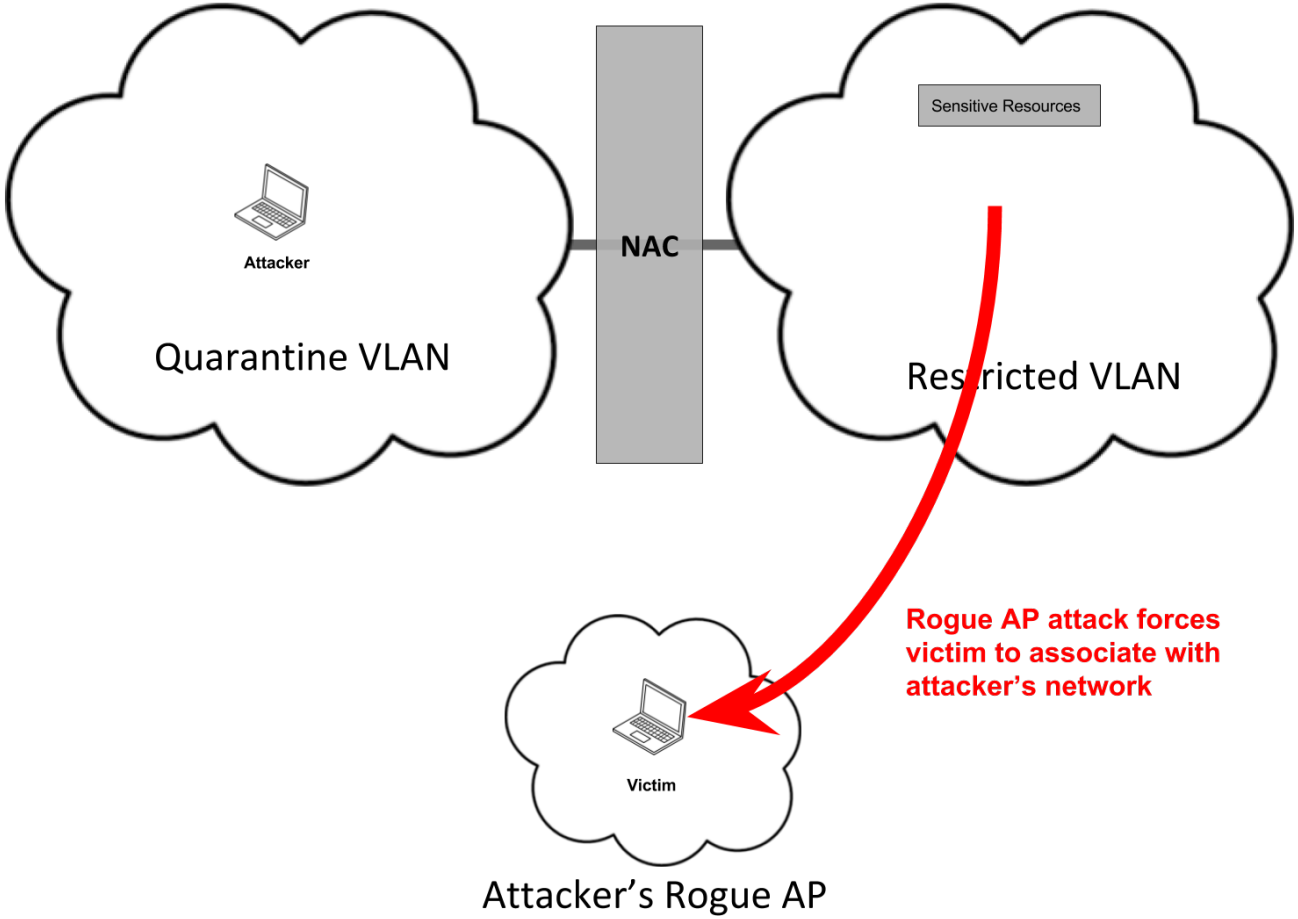
- No network access required
- Not limited to a local subnet (you get *everything* that is connected to wireless)
- Not a passive attack

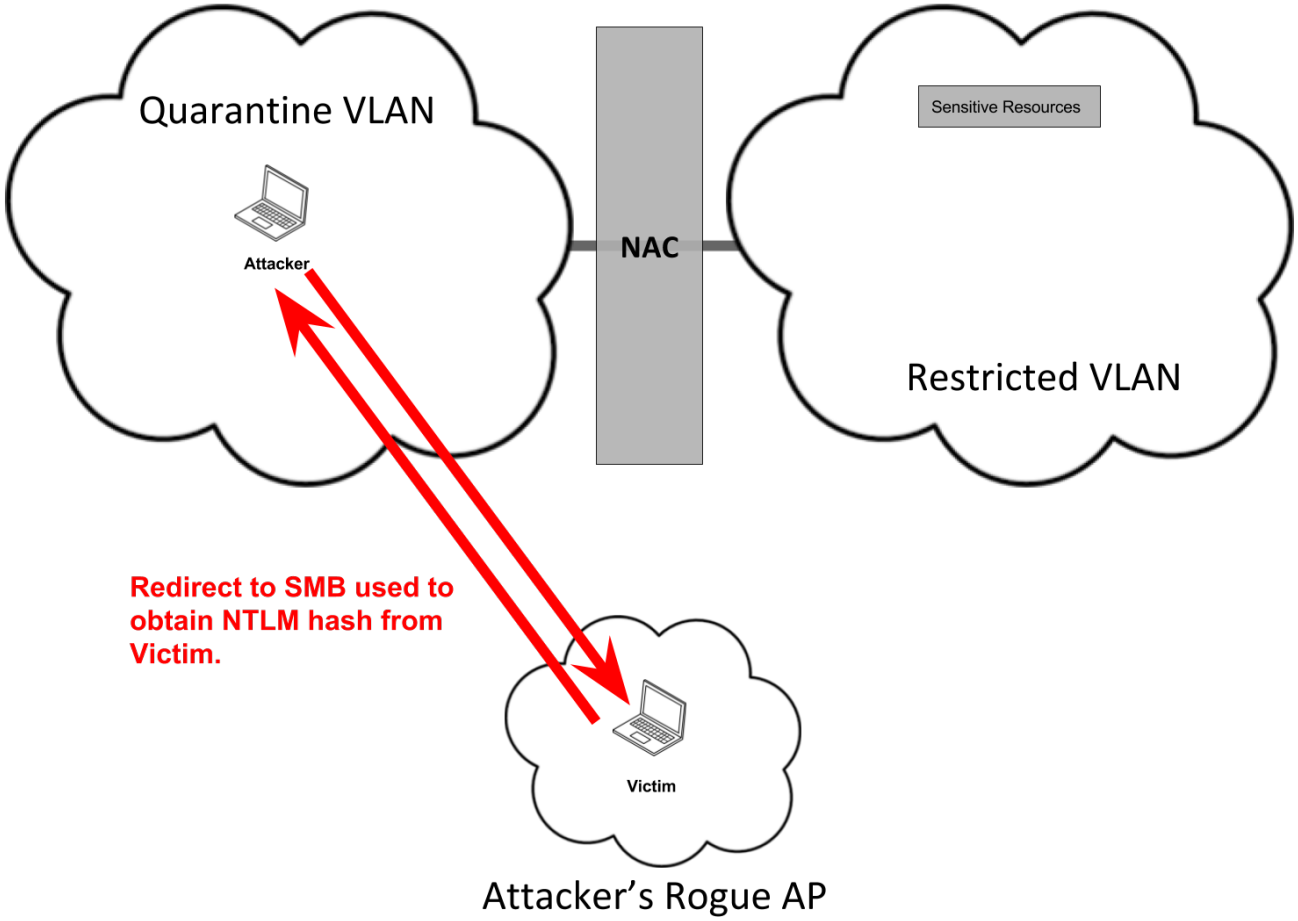
Back to our scenario...

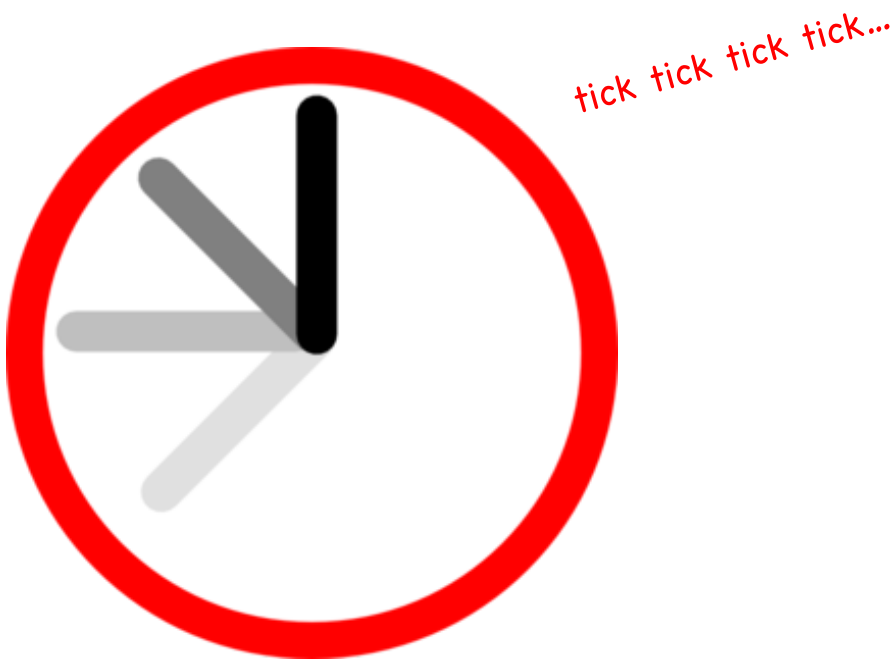
Indirect Wireless Pivots

Use Rogue Access Point attacks to ***bypass port-based access control mechanisms***

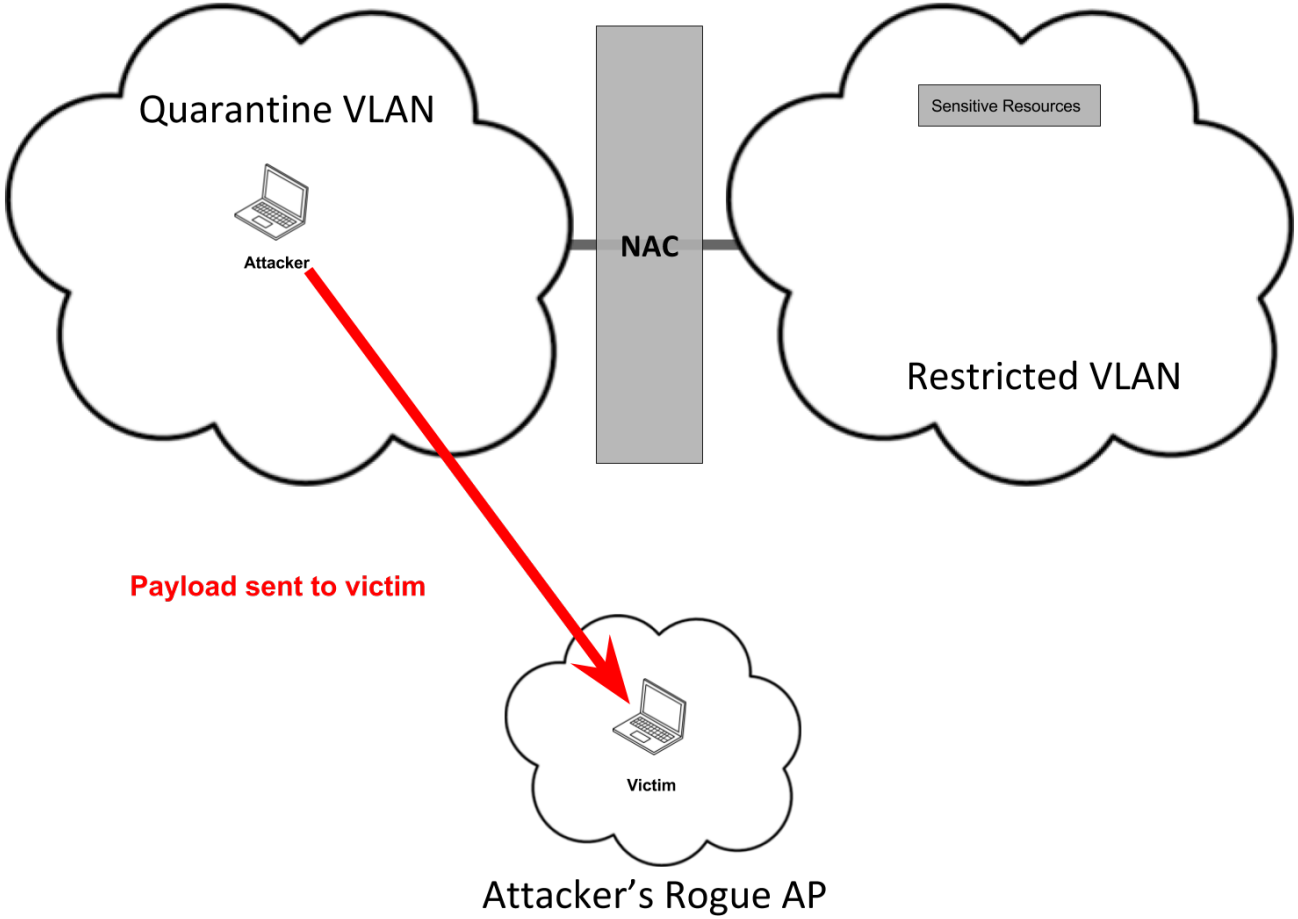

Attacker
NAC
Sensitive Resources
Victim
Quarantine VLAN
Restricted VLAN

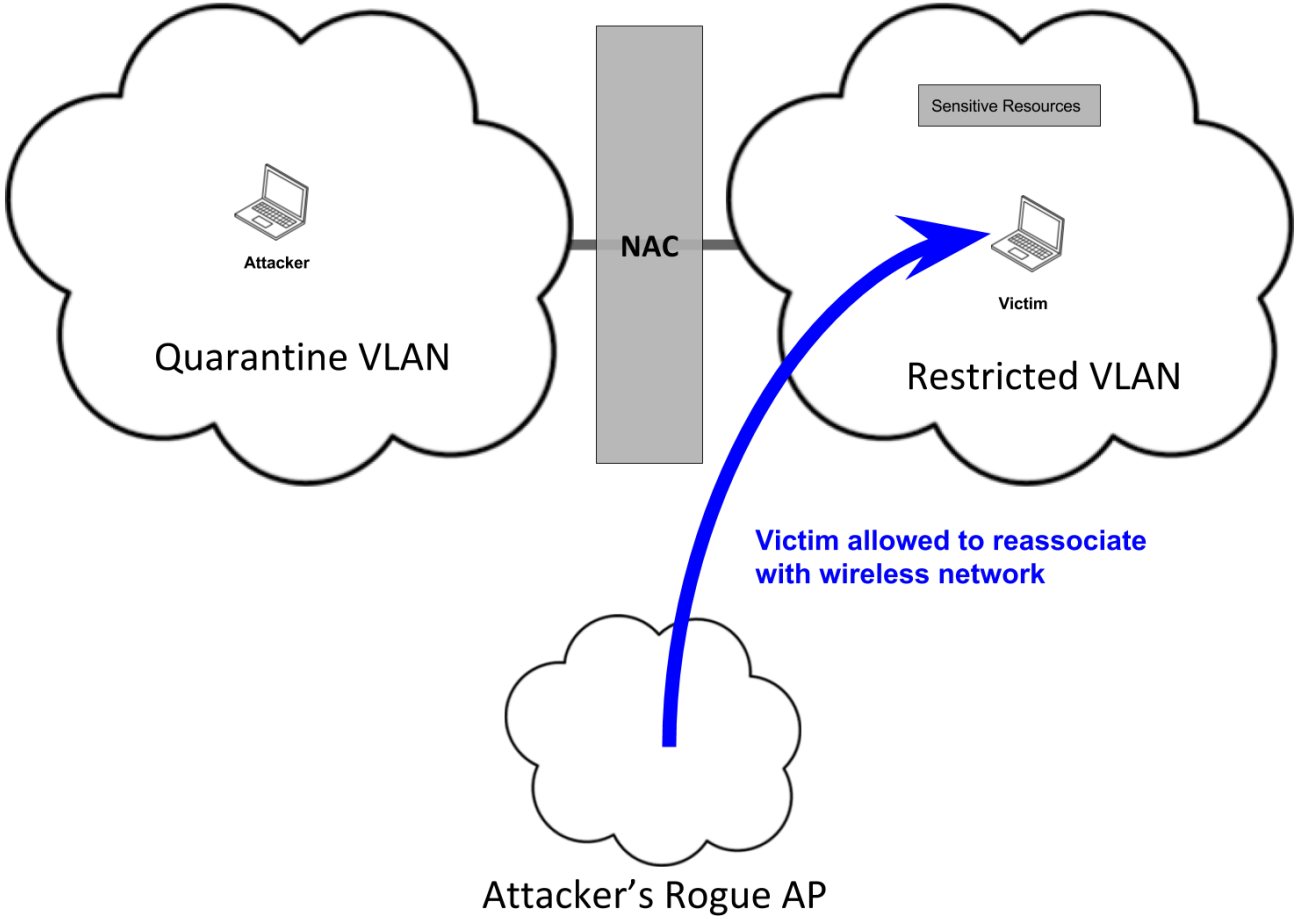

Attacker
Quarantine VLAN
NAC
Sensitive Resources
Restricted VLAN
Rogue AP attack forces victim to associate with attacker's network
Victim
Attacker's Rogue AP

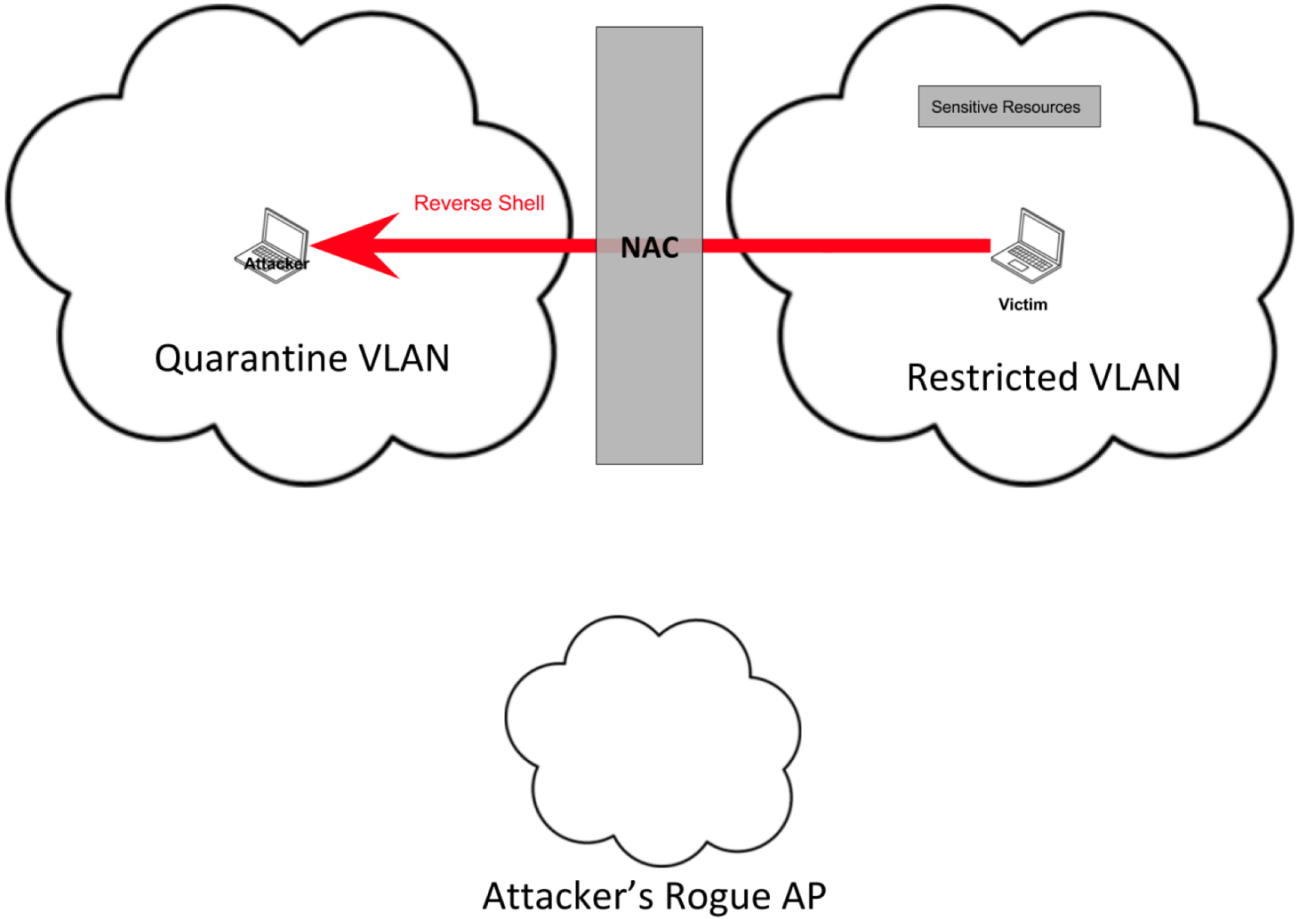

Quarantine VLAN
Attacker
NAC
Sensitive Resources
Restricted VLAN
Redirect to SMB used to obtain NTLM hash from Victim.
Victim
Attacker's Rogue AP



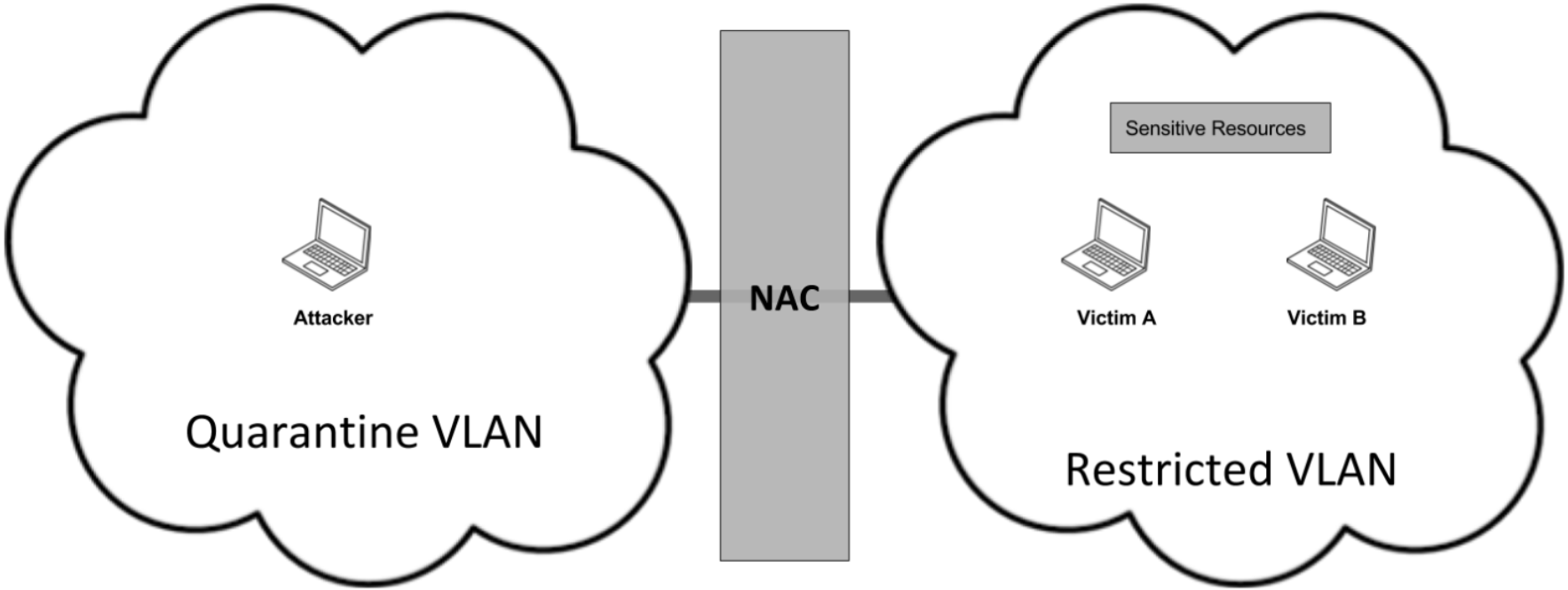
Hashes cracked offline…

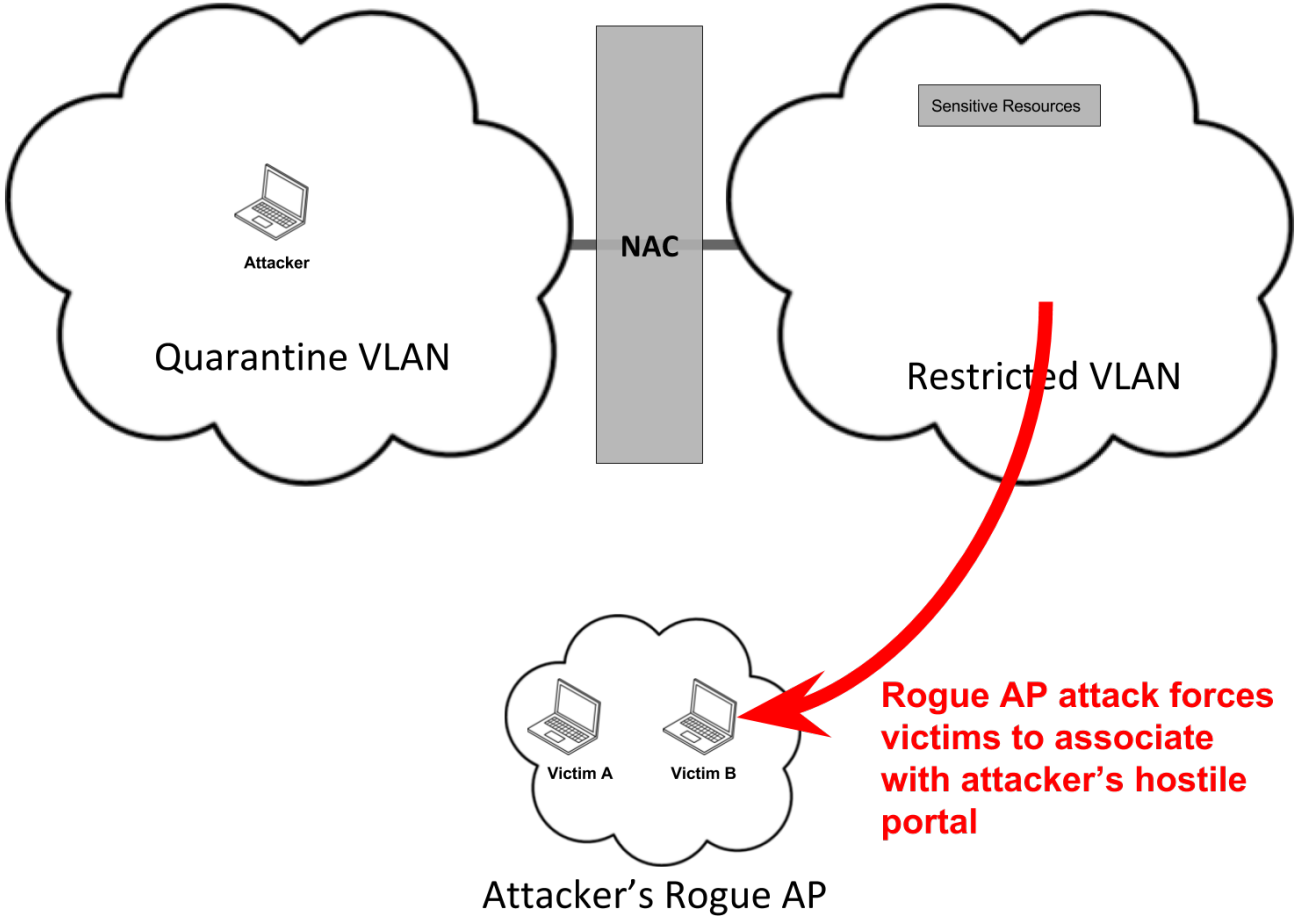

Quarantine VLAN
Attacker
NAC
Sensitive Resources
Restricted VLAN
Payload sent to victim
Victim
Attacker's Rogue AP

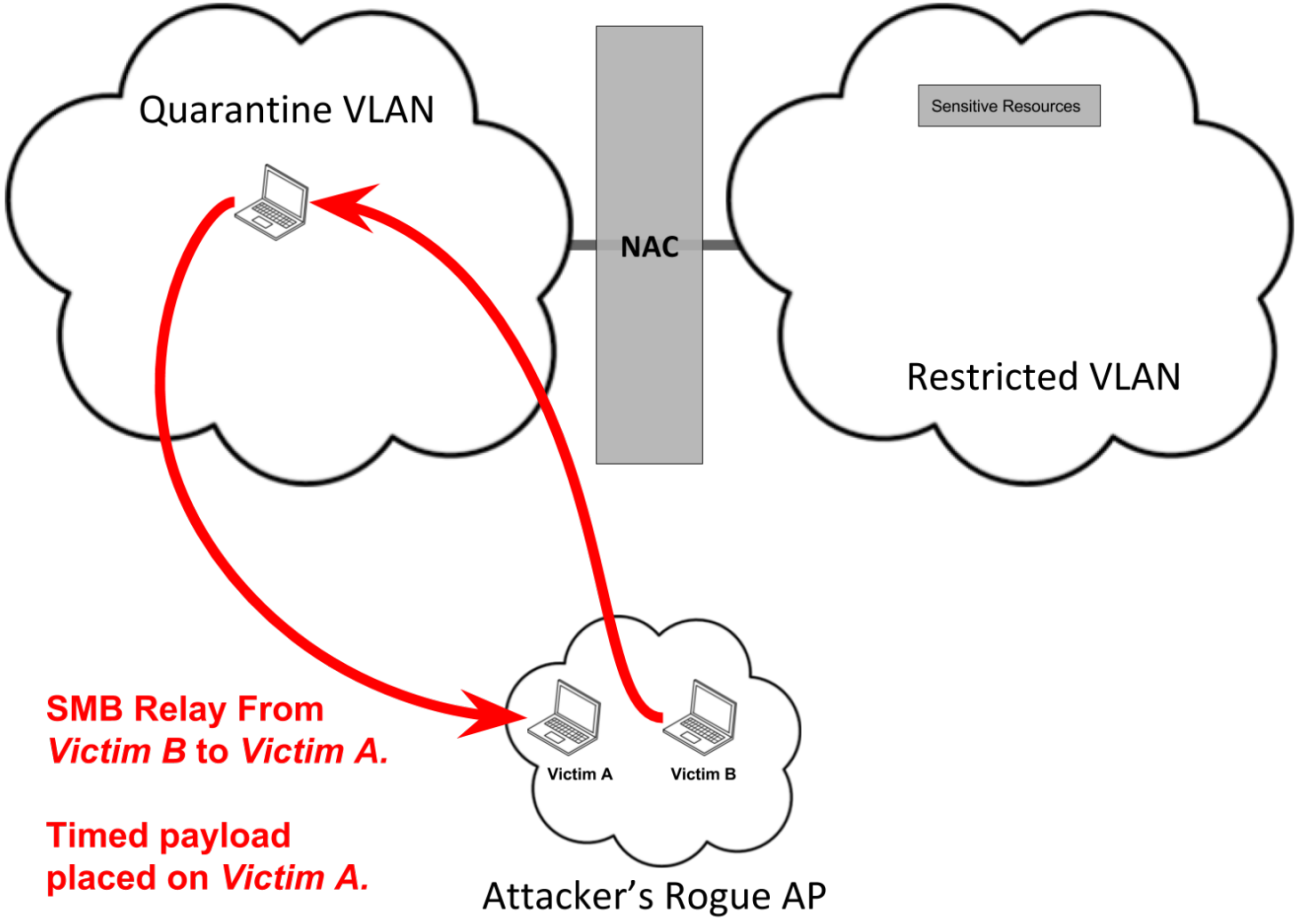

Attacker
NAC
Sensitive Resources
Victim
Quarantine VLAN
Restricted VLAN
Victim allowed to reassociate
with wireless network
Attacker's Rogue AP

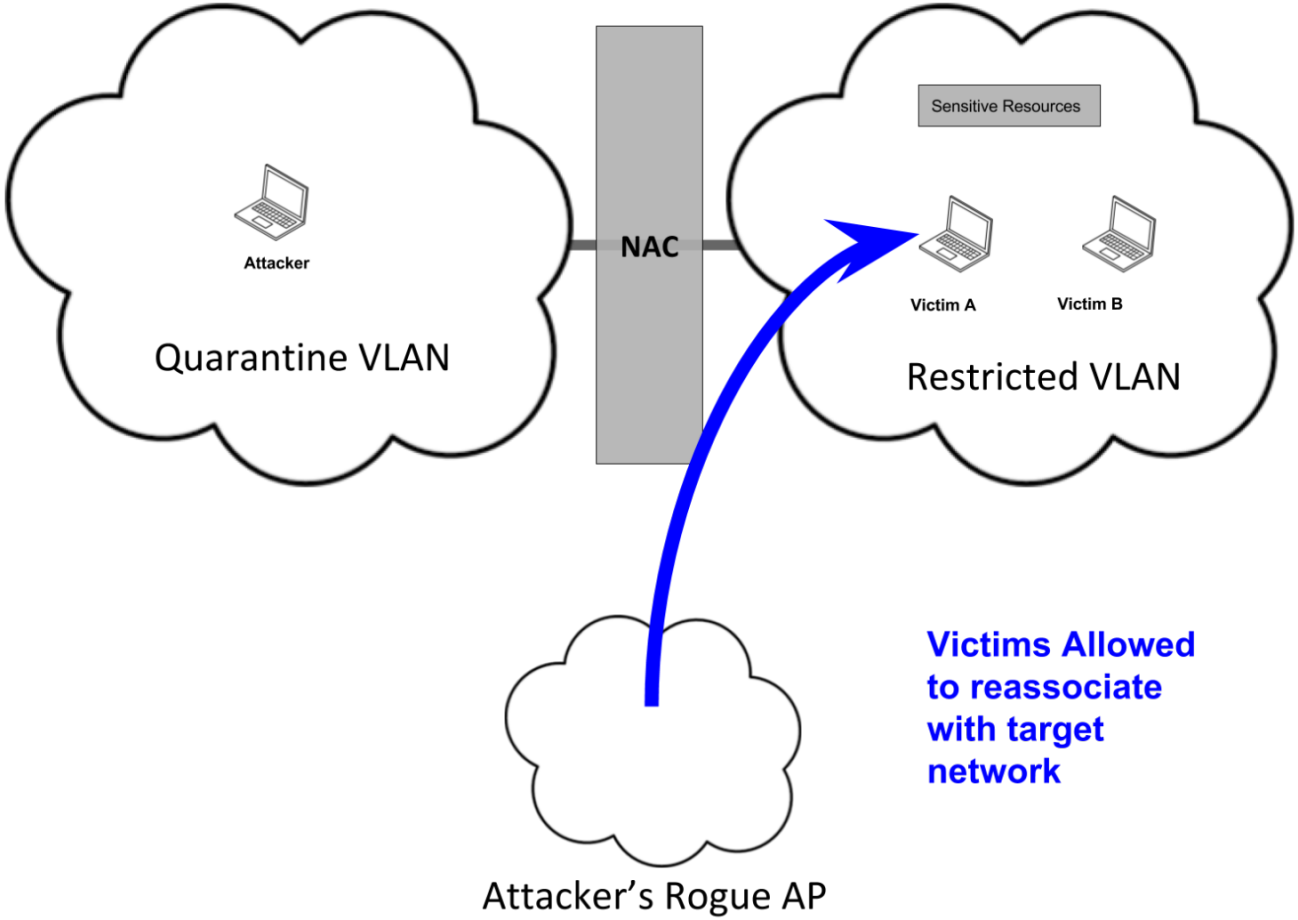

Sensitive Resources
Reverse Shell
NAC
Attacker
Victim
Quarantine VLAN
Restricted VLAN
Attacker's Rogue AP

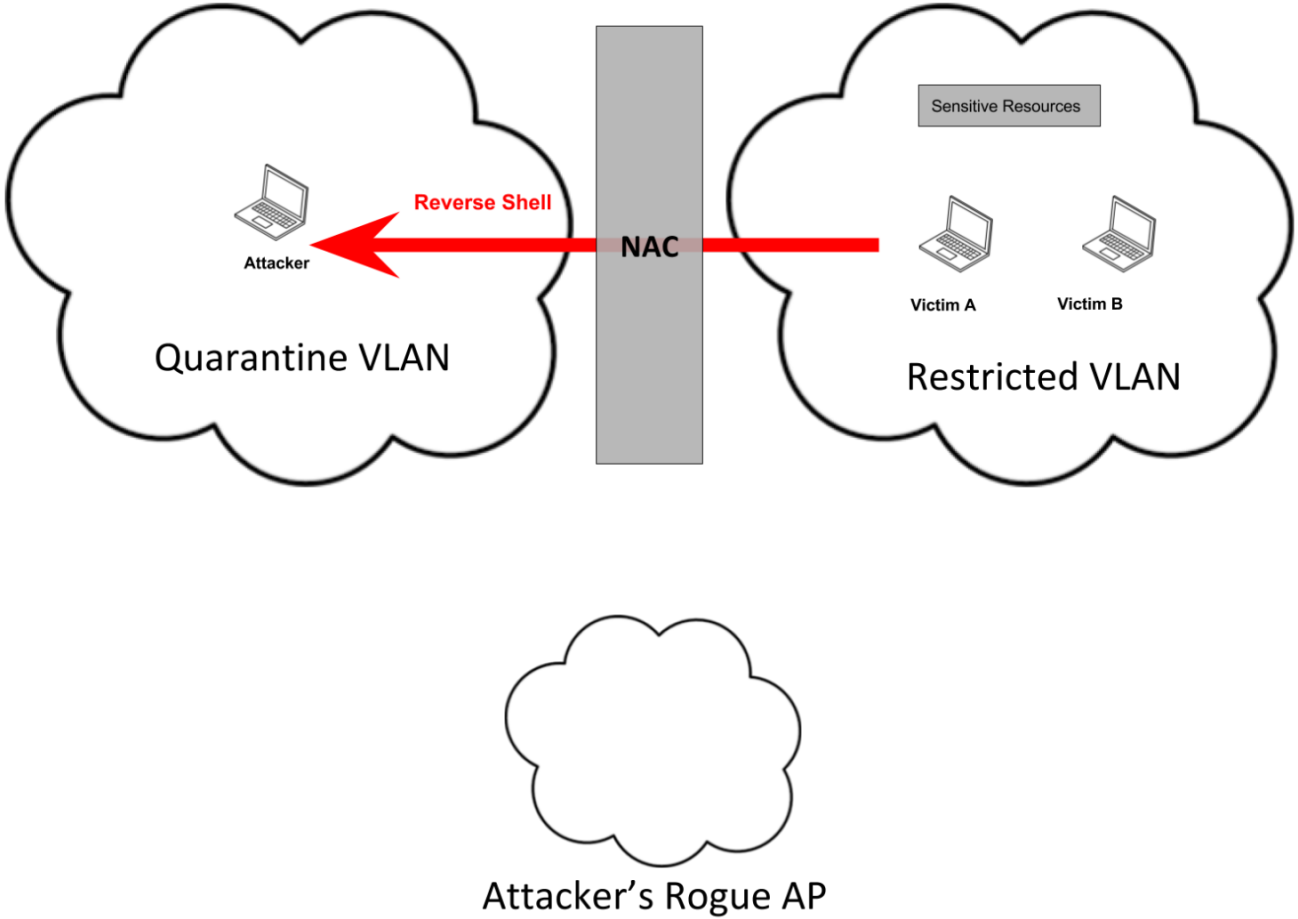
Better approach: SMB Relay


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
Rogue AP attack 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

DEMO

Indirect Wireless Pivots:

Equivalent technique in a wired network:

- Unplugging an authorized device from the wall and connecting it to a hostile network on which it can be attacked.

Indirect Wireless Pivots:

- Port-based access controls rely on the assumption that the physical layer can be trusted
- In a wireless network, WPA2-EAP is the means through which the integrity of the physical layer is protected
- When weak forms of WPA2-EAP are used, the attacker can freely control the physical layer using rogue access point attacks, rendering port-based NAC mechanisms useless

Indirect Wireless Pivots:

- Demonstrates that port-based NAC mechanisms do not effectively mitigate the risk presented by weak WPA2-EAP implementations

Indirect Wireless Pivots:

- Demonstrates that adding port-based NAC mechanisms to a wireless network does not make the use of EAP-TTLS and EAP-PEAP any less inappropriate if the network in question is used to grant access to sensitive information
- I.e. PCI or HIPAA data (compliant != secure !!!!)

A Case For EAP-TLS:

It’s not as bad as it used to be.

- Use Group Policy to configure 802.1x clients [26]

Best option:

- Use a private CA
- Leverage Active Directory to deploy EAP-TLS
- Distribute the server cert to clients using a solid MDM or BYOD onboarding solution [27]

A Case For EAP-TLS:

You can even use Let’s Encrypt:

- Note: even the folks at Let’s Encrypt state that this is far from the best option out there [27]

Closing thoughts:

- Just because wireless and wired networks operate similarly at the logical level, does not mean that they work the same way at the physical level
- As a community, we should question whether it is truly a sound business decision to neglect EAP-TLS in favor of a more reactive approach that focuses on access control and threat containment.
- The needs for convenience and security are often at odds with one another. Maintain a healthy skepticism towards proposed solutions that promise both.

Tool Release:
github.com/s0lst1c3/eaphammer

Whitepaper:
blog.gdssecurity.com/labs/2017/8/31/whitepaper-the-black-art-of-wireless-post-exploitation-bypas.html

References:

[1] http://dl.acm.org/citation.cfm?id=1360099

[2] http://asleap.sourceforge.net/asleap-defcon.pdf

[3] http://theta44.org/karma/aawns.pdf

[4] http://www.willhackforsushi.com/presentations/PEAP_Shmoocon2008_Wright_Antoniewicz.pdf

[5] https://defcon.org/images/defcon-22/dc-22-presentations/White-deVilliers/DEFCON-22-Dominic-White-Ian-de-Villiers-Manna-from-Heaven-Detailed-UPDATED.pdf

References:

[6] https://tools.ietf.org/html/rfc3579

[7] https://tools.ietf.org/html/rfc4017

[8] https://tools.ietf.org/html/rfc5281

[9] http://www.willhackforsushi.com/?page_id=67

[10] https://tools.ietf.org/html/rfc5216

References:

[11] https://4310b1a9-a-93739578-s-sites.googlegroups.com/a/riosec.com/home/articles/Open-Secure-Wireless/Open-Secure-Wireless.pdf?attachauth=ANoY7cp3gqgS8JIZY9jdvVoc0DQu7i16aoRTm6icHP-NJyZfYMtj72S6WDlQPyl7vgQYy14fu-5t3mssAfFhmQo_bl6OYyqK5dENUGHee-40daHWqAem3m2dWJd6jNeuP9ZSnaezoRkarq_s8J92z3SJMEXdxdAUkF1nMzRoaCPeG2anVCQ1tSxB8Uupviji6Pom1xr10aRuISitMk7bfMmAQ00VBESXW7lWkM1veZMlNA24NpcKkmcdvF3u_R21u_b_pkEAGIJ0&attredirects=0

References:

[12] https://www.blackhat.com/presentations/bh-dc-07/Arkin/Presentation/bh-dc-07-Arkin-ppt-up.pdf

[13] https://www.sans.org/reading-room/whitepapers/analyst/securing-personal-mobile-device-next-gen-network-access-controls-35627

[14] [VENDOR REDACTED]

[15] https://blogs.technet.microsoft.com/josebda/2010/12/01/the-basics-of-smb-signing-covering-both-smb1-and-smb2/

References:

[16] https://blog.gdssecurity.com/labs/2013/2/5/resurrecting-wifitap.html

[17] http://sid.rstack.org/static/articles/w/i/f/Wifitap_README_202c.html

[18] https://www.aircrack-ng.org/doku.php?id=airtun-ng

[19] https://www.aircrack-ng.org/doku.php?id=tkiptun-ng

[20] http://www.ietf.org/rfc/rfc1001.txt

[21] http://www.rfc-editor.org/rfc/rfc1002.txt

References:

[22] https://msdn.microsoft.com/en-us/library/dd240328.aspx

[23] https://www.trustwave.com/Resources/SpiderLabs-Blog/Introducing-Responder-1-0/

[24] https://www.cylance.com/redirect-to-smb

[25] https://technet.microsoft.com/en-us/library/dd283093(v=ws.10).aspx

[26] https://msdn.microsoft.com/en-us/library/dd759173(v=ws.11).aspx

[27] https://framebyframewifi.net/2017/01/29/use-lets-encrypt-certificates-with-freeradius/

References:

[28] https://docs.microsoft.com/en-us/windows/configuration/manage-wifi-sense-in-enterprise

[29] https://technet.microsoft.com/en-us/library/cc957983.aspx

[30] https://www.helpnetsecurity.com/2017/04/26/lure10-exploiting-wi-fi-sense/

[31]http://web.archive.org/web/20160203043946/https://www.cloudcracker.com/blog/2012/0 7/29/cracking-ms-chap-v2/

[32] http://crack.sh/bsideslv2017.pdf

[33] https://crack.sh/