

The Black Art of Wireless Post-Exploitation:

Bypassing Port-Based Access Controls Using
Indirect Wireless Pivots

DEF CON 25

Gabriel “solstice” Ryan

Disclaimers & Updates

These slides are an early version designed for pre-release prior to DEF CON 25.
All content will be updated by the time of the presentation at DEF CON 25 in July.
Final versions of all content will be available at:

<https://blog.gdssecurity.com>

<https://github.com/s0lst1c3/eaphammer>

net user author /domain

Gabriel Ryan

Security Engineer @ Gotham Digital Science

Appsec | Penetration Testing | Red Team | Research

[@s0lst1c3](#)

gryan@gdssecurity.com

labs@gdssecurity.com

New in this presentation:

Hostile Portal Attacks:

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

Indirect Wireless Pivots:

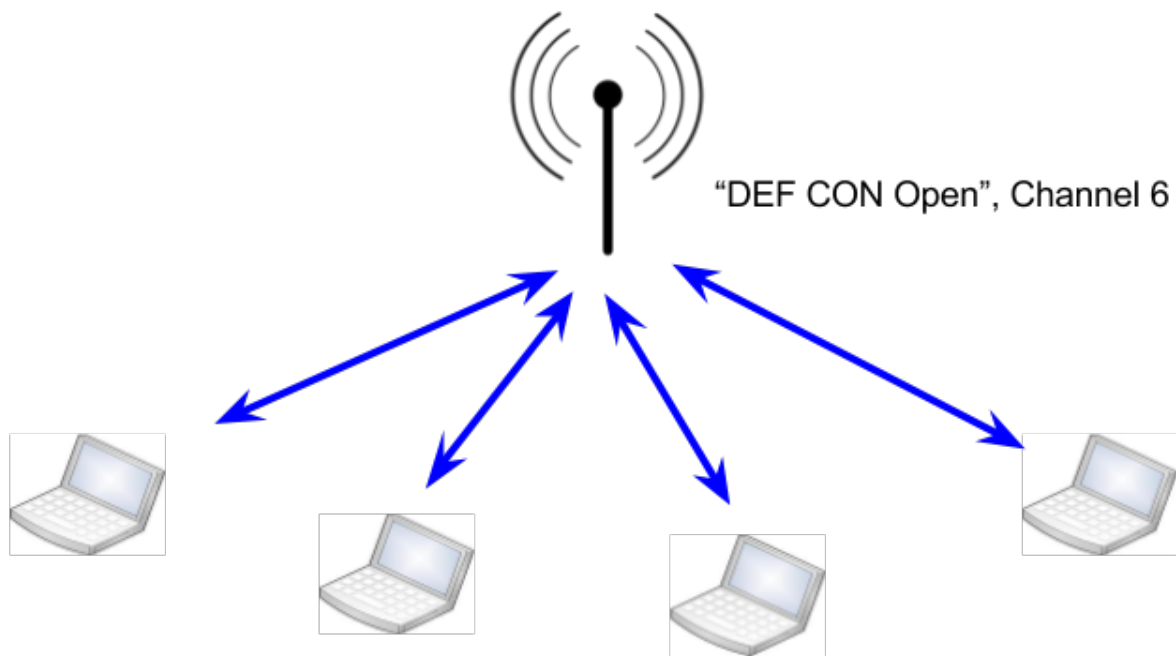
- Use Rogue AP attacks to ***bypass port-based access control mechanisms***

WPA2-EAP

Wireless Theory: Evil Twin Attacks

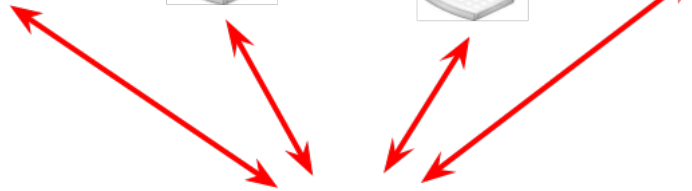
Rogue access point attacks:

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





"DEF CON Open", Channel 6



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 – asleep - Joshua Wright [2]

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

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

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

2017 – Lure10 Attacks – George Chatzisoifroniou [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

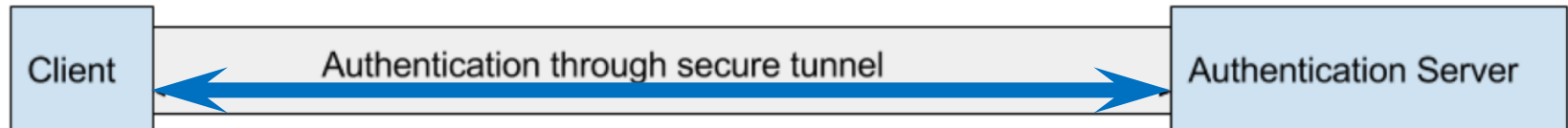
WPA2-EAP

Logically:

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



If client accepts certificate...

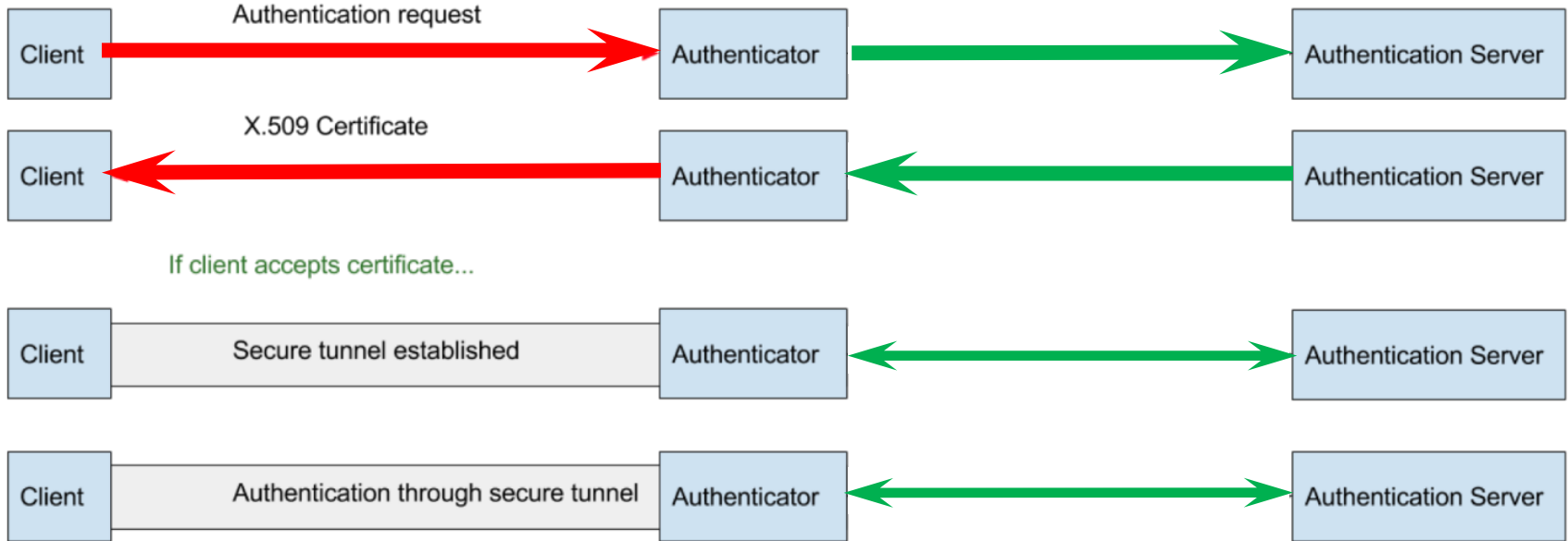


Wireless Theory: EAP

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])





 **Untrusted**

The attack:

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

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”:

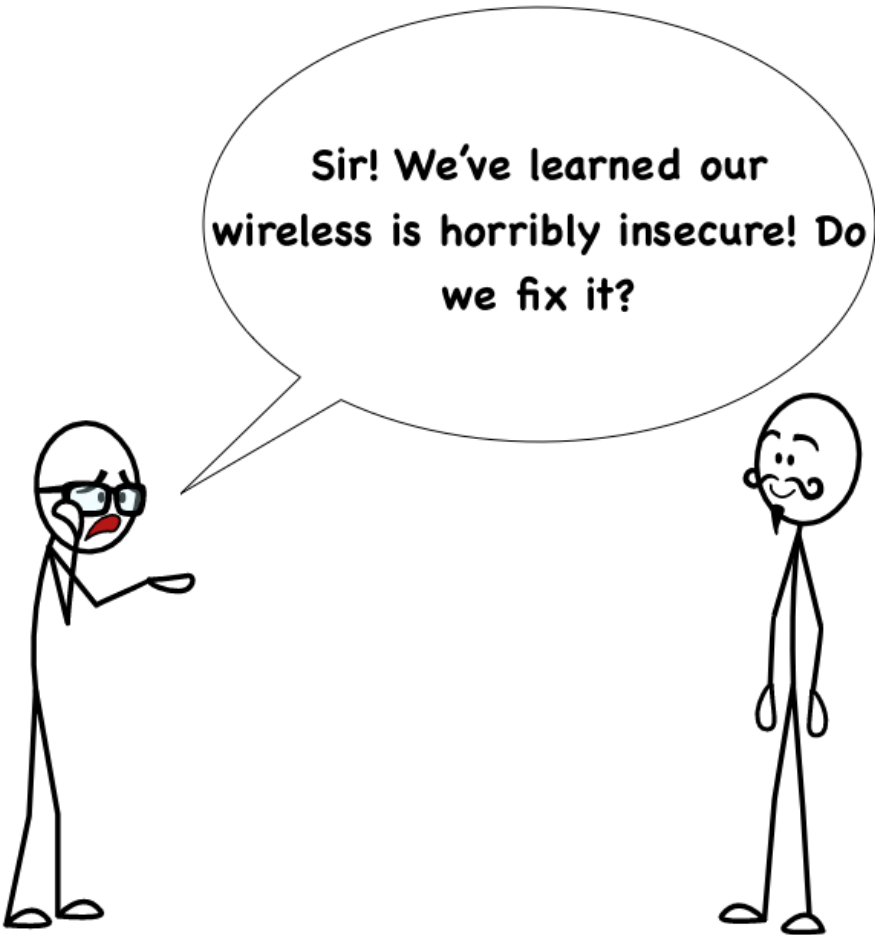
The current trend:

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

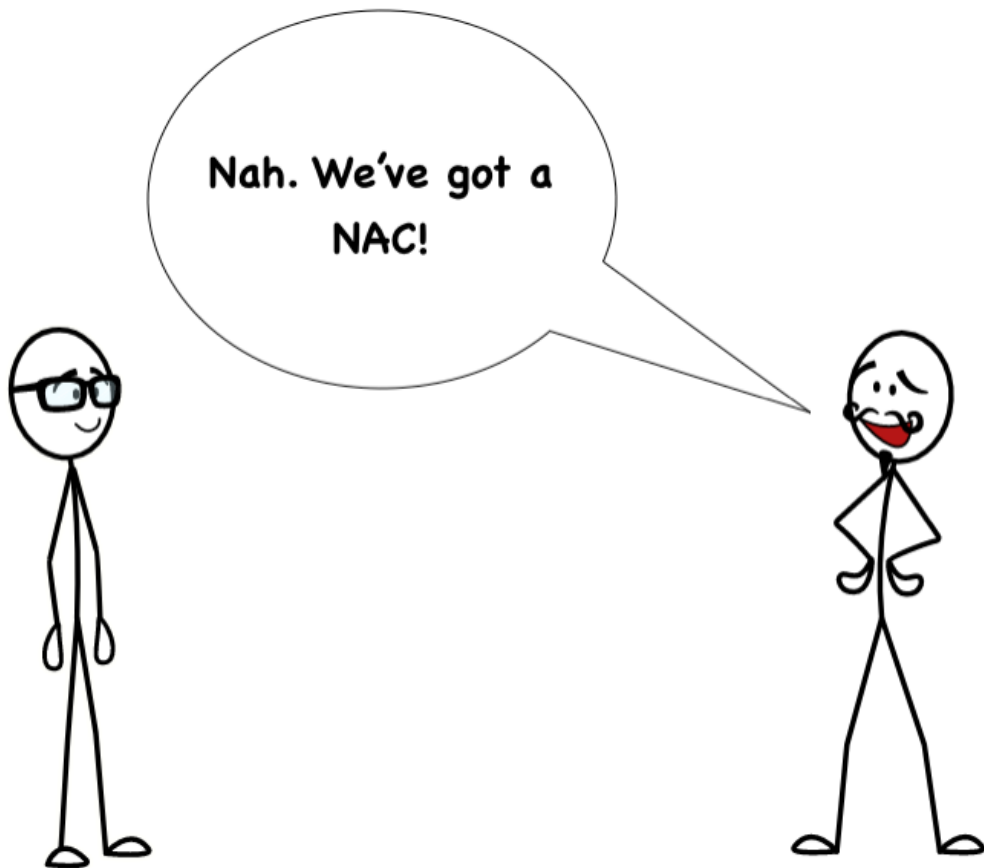
Containment vs. Prevention

Does this actually work?

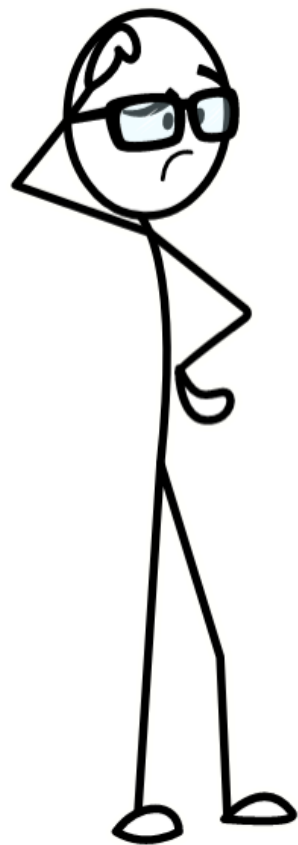
Classic WLAN Access Control Mechanisms



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?



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.

Background Info: SMB Relay Attacks

SMB Relay Attacks

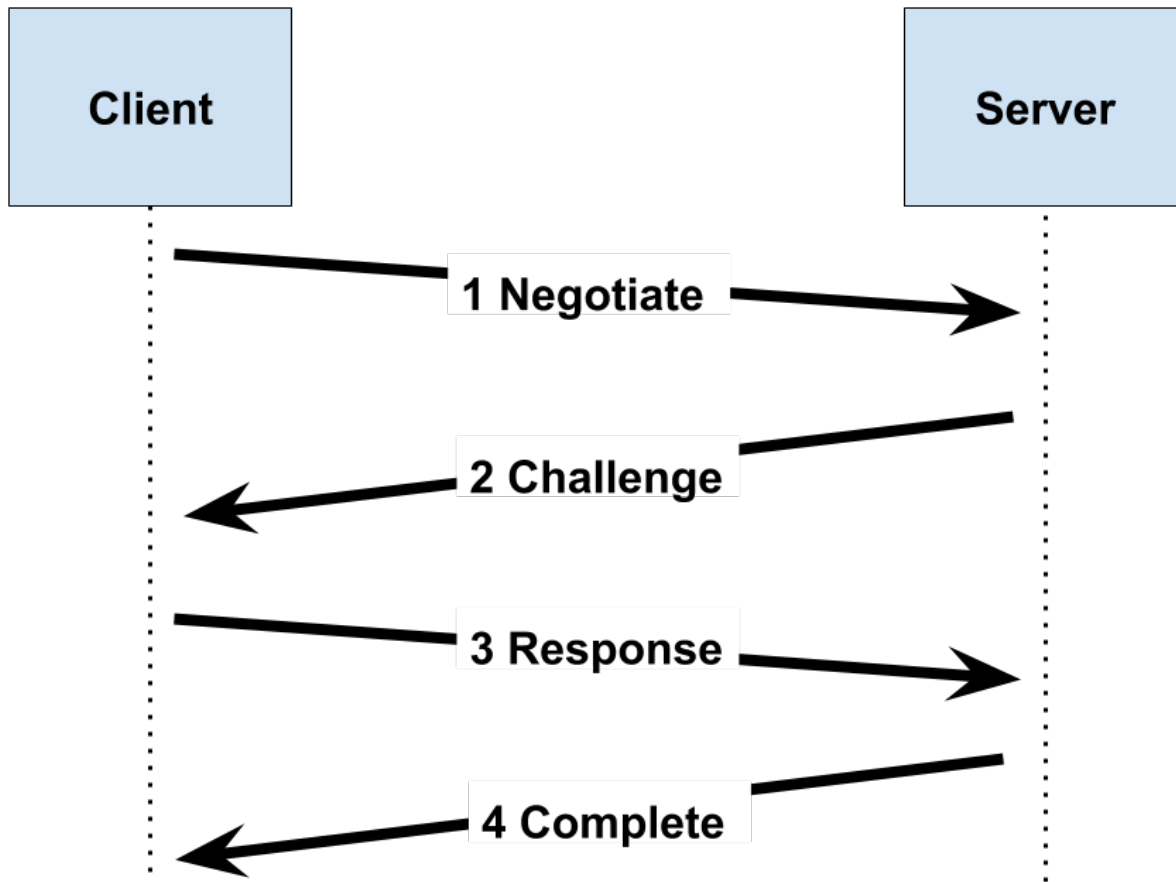
NTLM is a simple authentication protocol:

- challenge/response mechanism... remember EAP-PEAP? ;)

SMB Relay Attacks

NTLM authentication process:

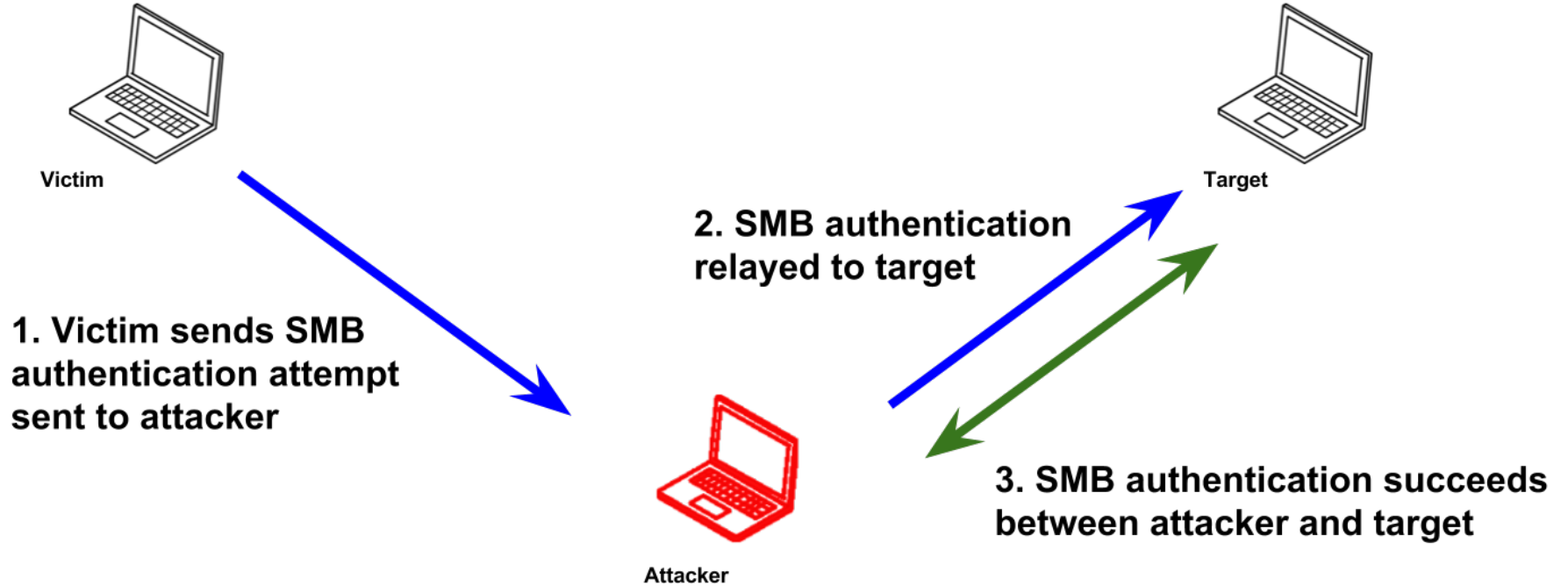
1. Client attempts to authenticate
2. Server issues challenge to client (string of characters)
3. Client encrypts the challenge using its password hash (uh oh)
4. Client sends encrypted challenge back to server as response
5. Server decrypts response using *same* password hash (oh no)
6. If decrypted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



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



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

Demo time

No magic bullet

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

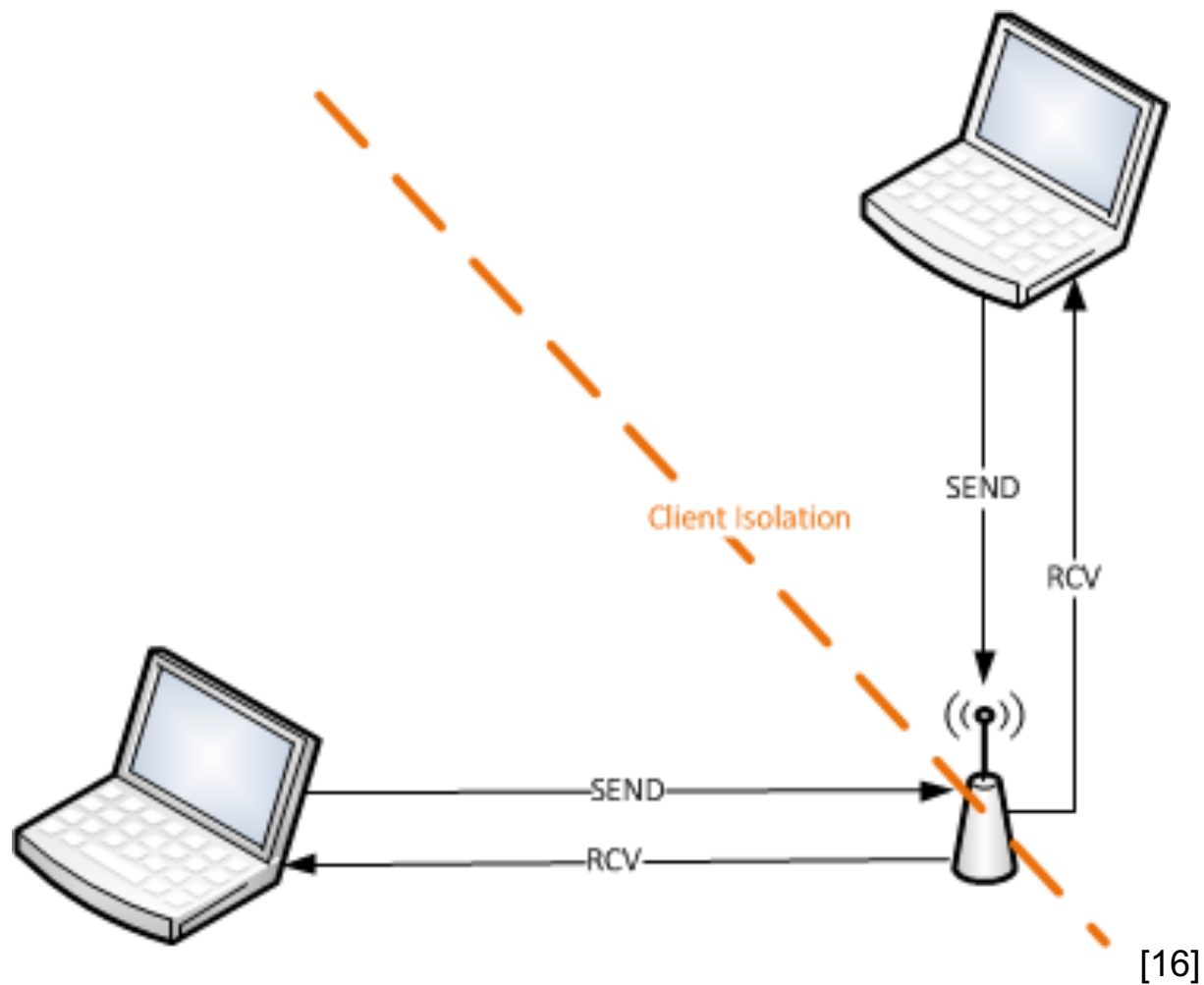
What about Client Isolation?

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]



The Problem:

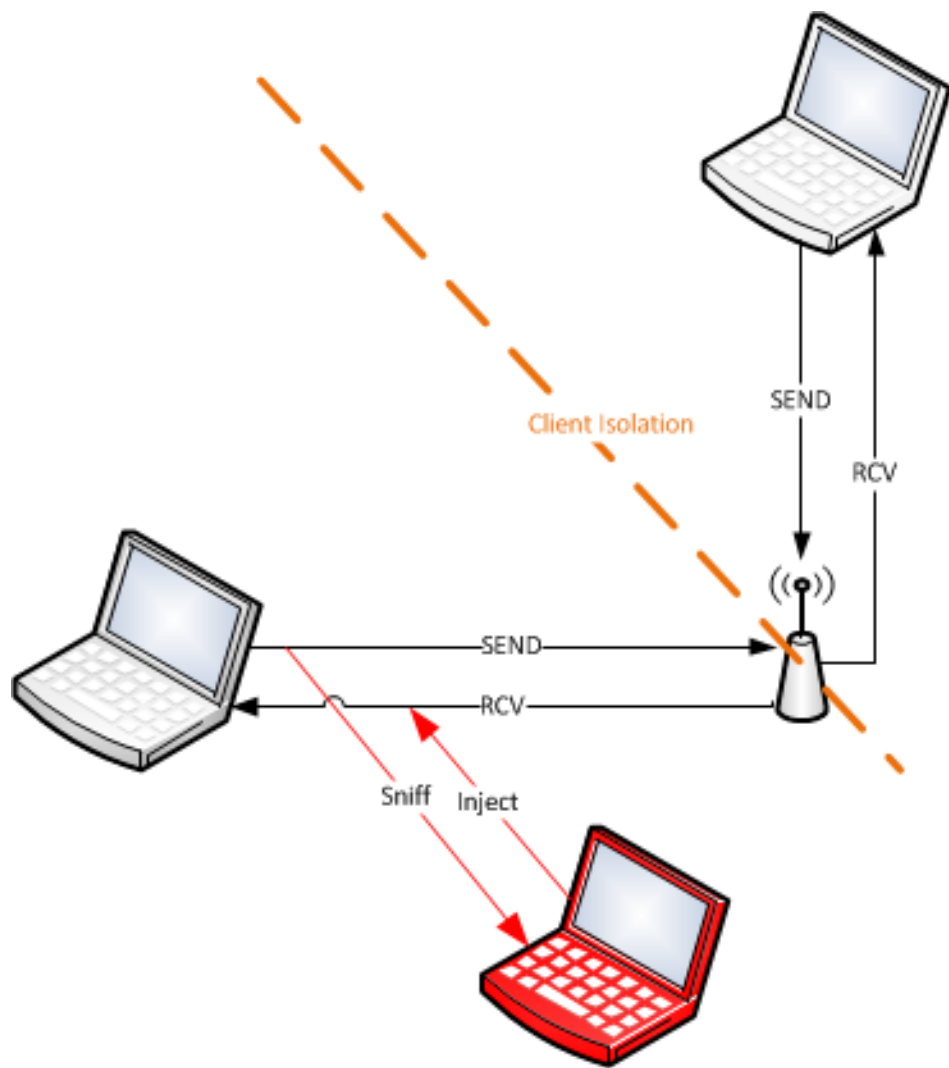
- Client isolation is a ***logical*** control, not a physical control
- The problem: “how do you prevent radio transceivers from communicating with one another?” [16]
- Cedric Blancher in 2005: You can’t. [17]

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]



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 time

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:

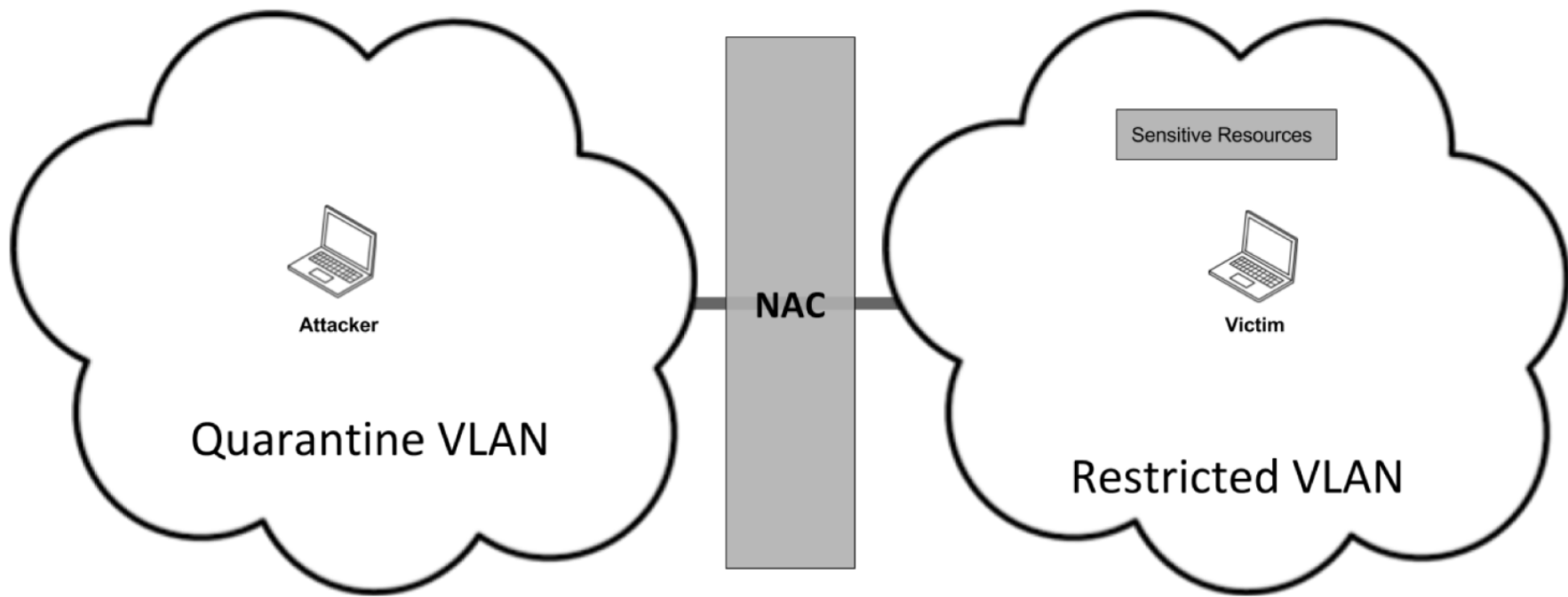
Violating access control policies causes the NAC to impose a *restriction*:

- 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



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

Demo time

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



HYATT
PLACE®

Eng

Choose Your Login Option

FIRST-TIME USERS

► Internet Plan

▼ In-House Guest

Room Number:

Last Name:

Submit



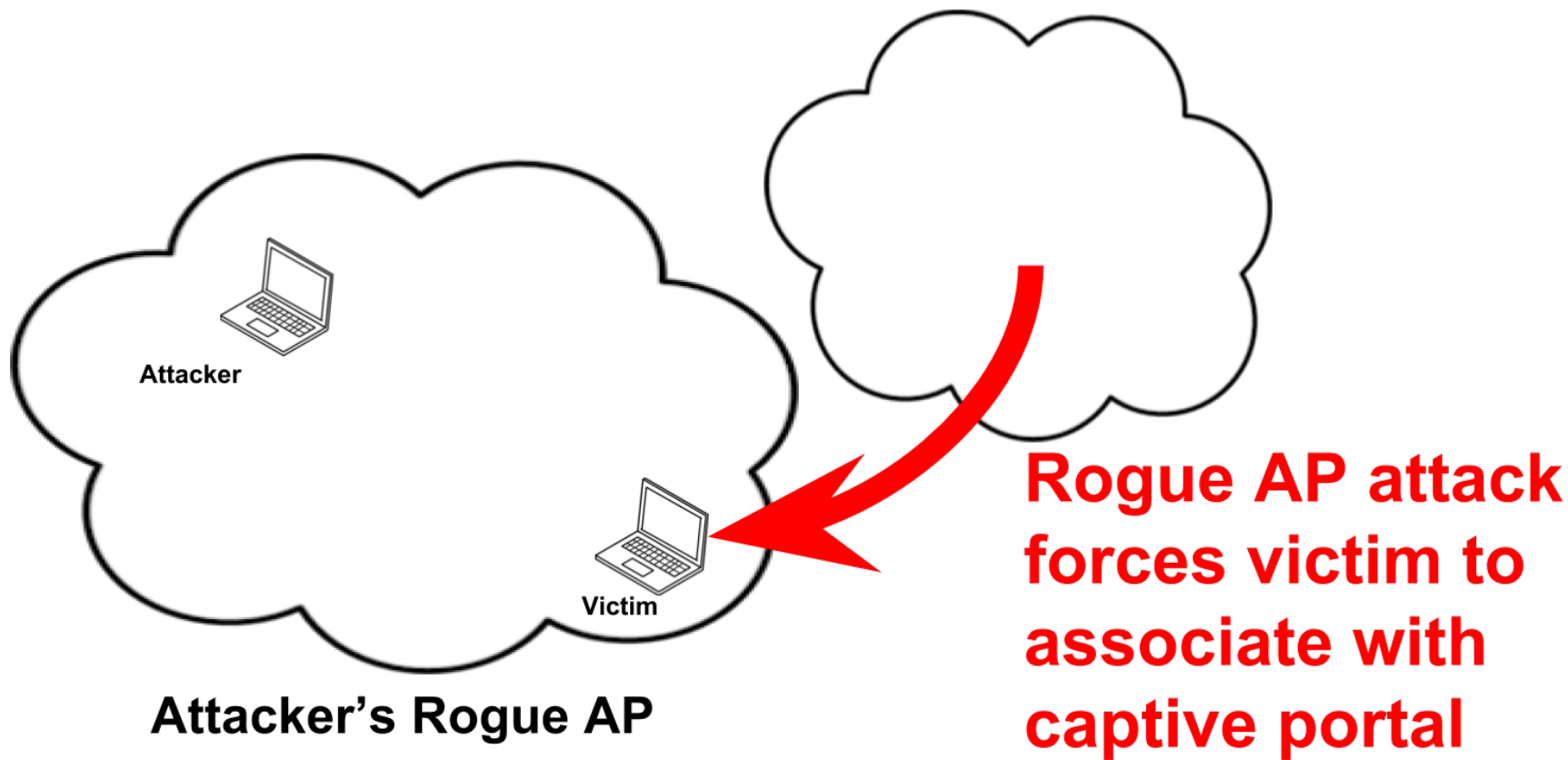
Captive Portal

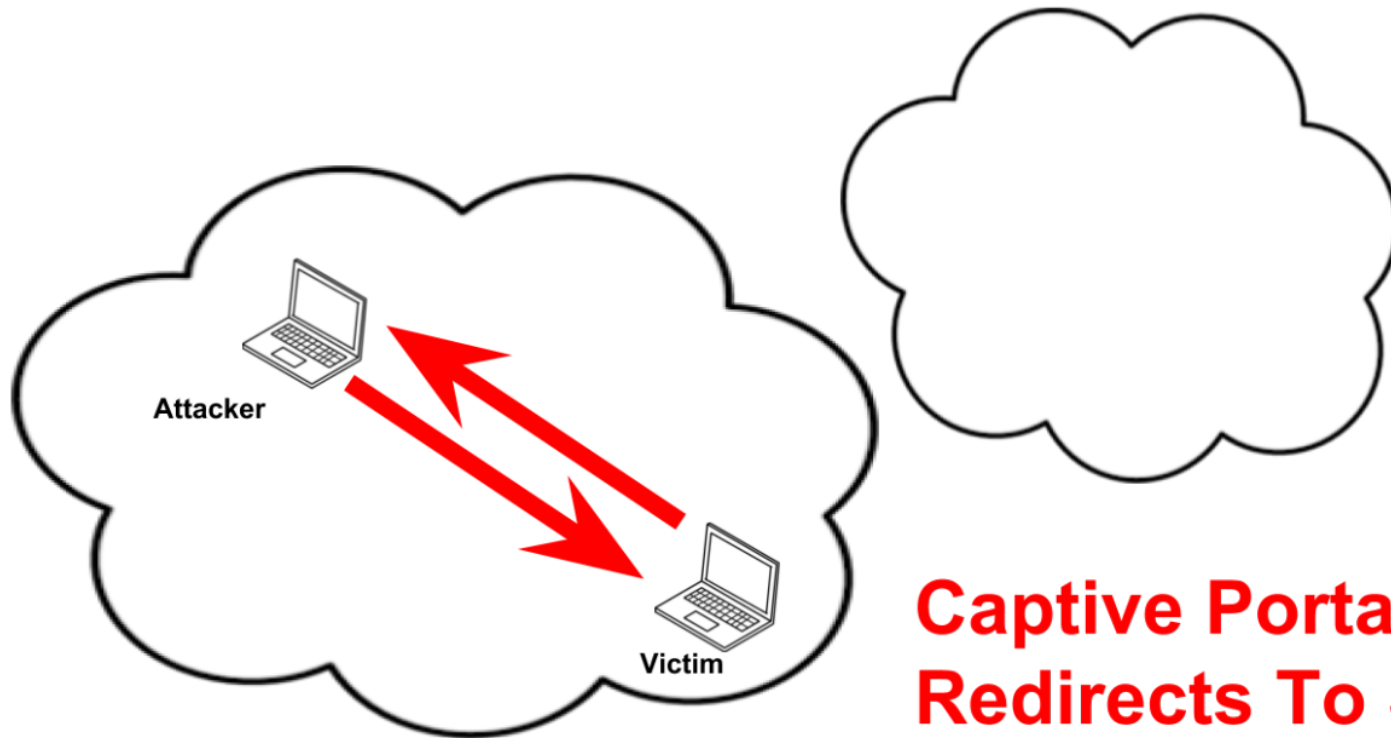
- All DNS queries resolved to captive portal
- All DNS traffic redirected to captive portal (optional)
- All HTTP traffic redirected to captive portal (optional)

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's Rogue AP

**Captive Portal
Redirects To SMB**

DEMO

What this gets you: lots and lots of NTLM hashes

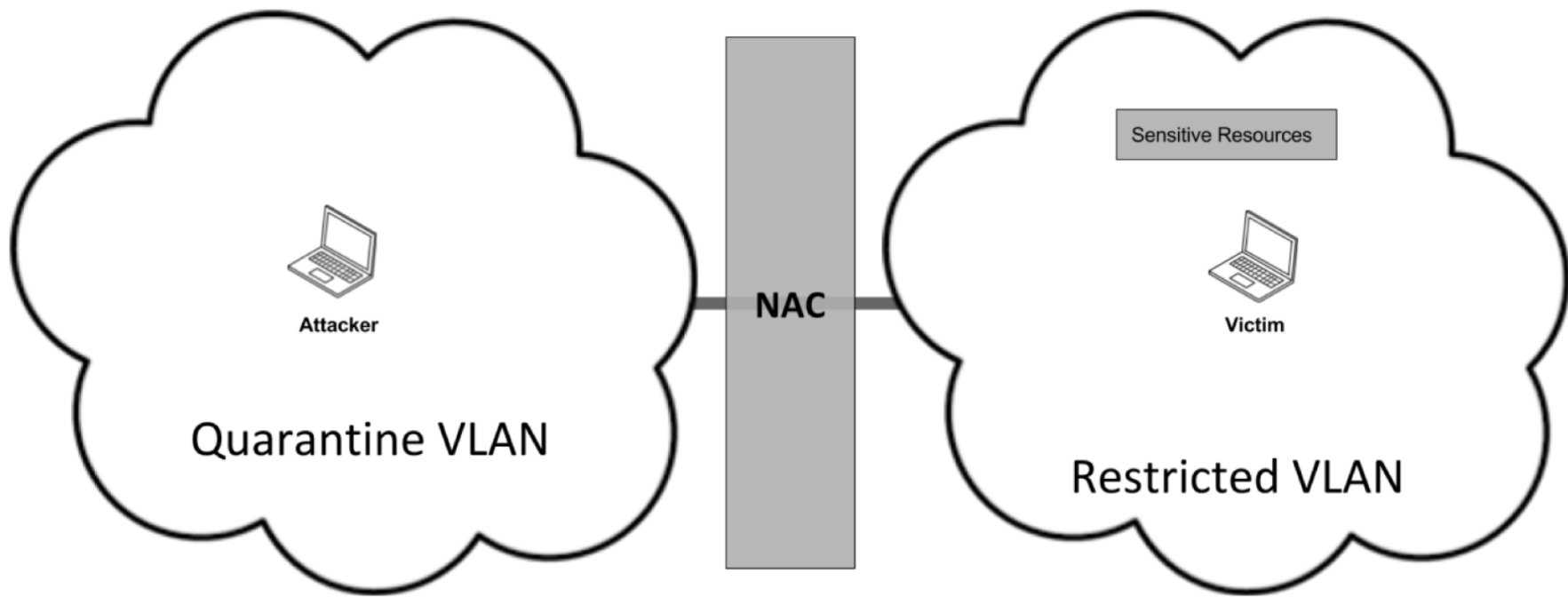
Similar results to LLMNR/NBT-NS poisoning, but with a few key advantages:

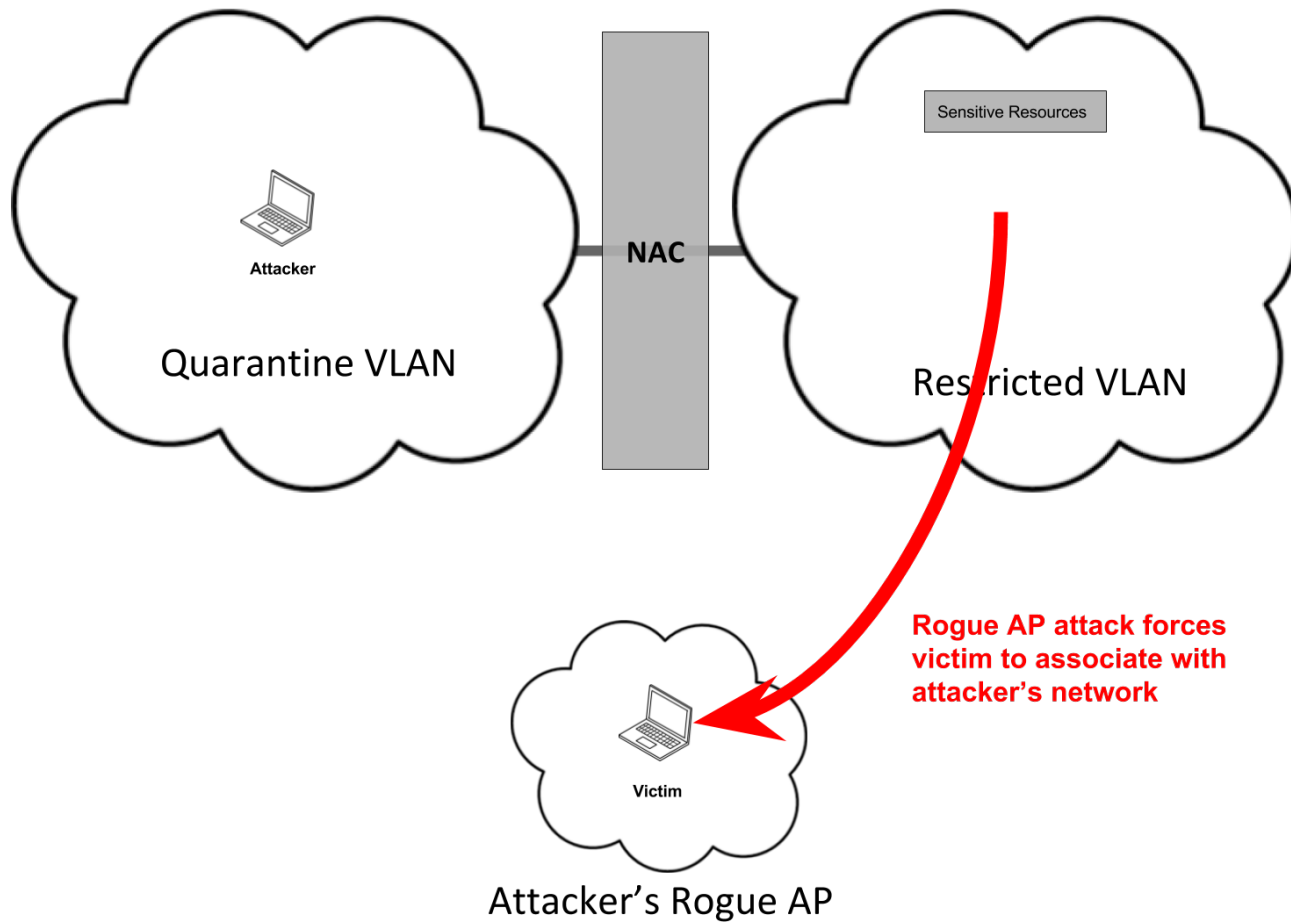
- No network access required
- Not limited to a local subnet (you get *everything* that is connected to wireless)
- Not a passive attack

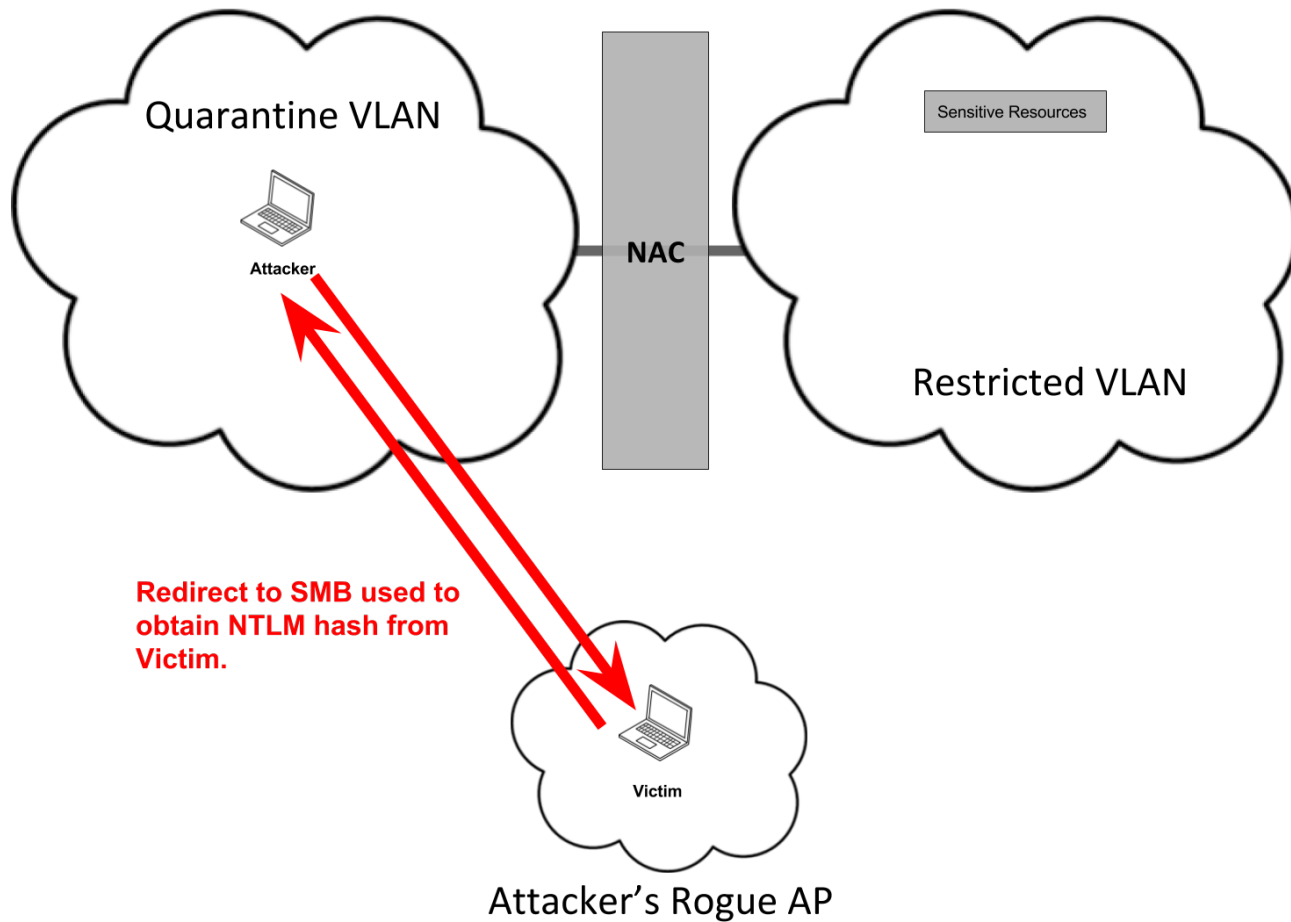
Back to our scenario...

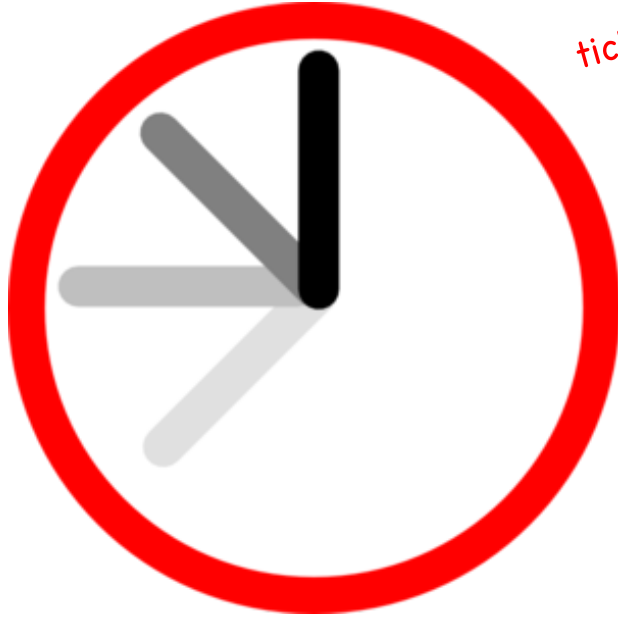
Indirect Wireless Pivots

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



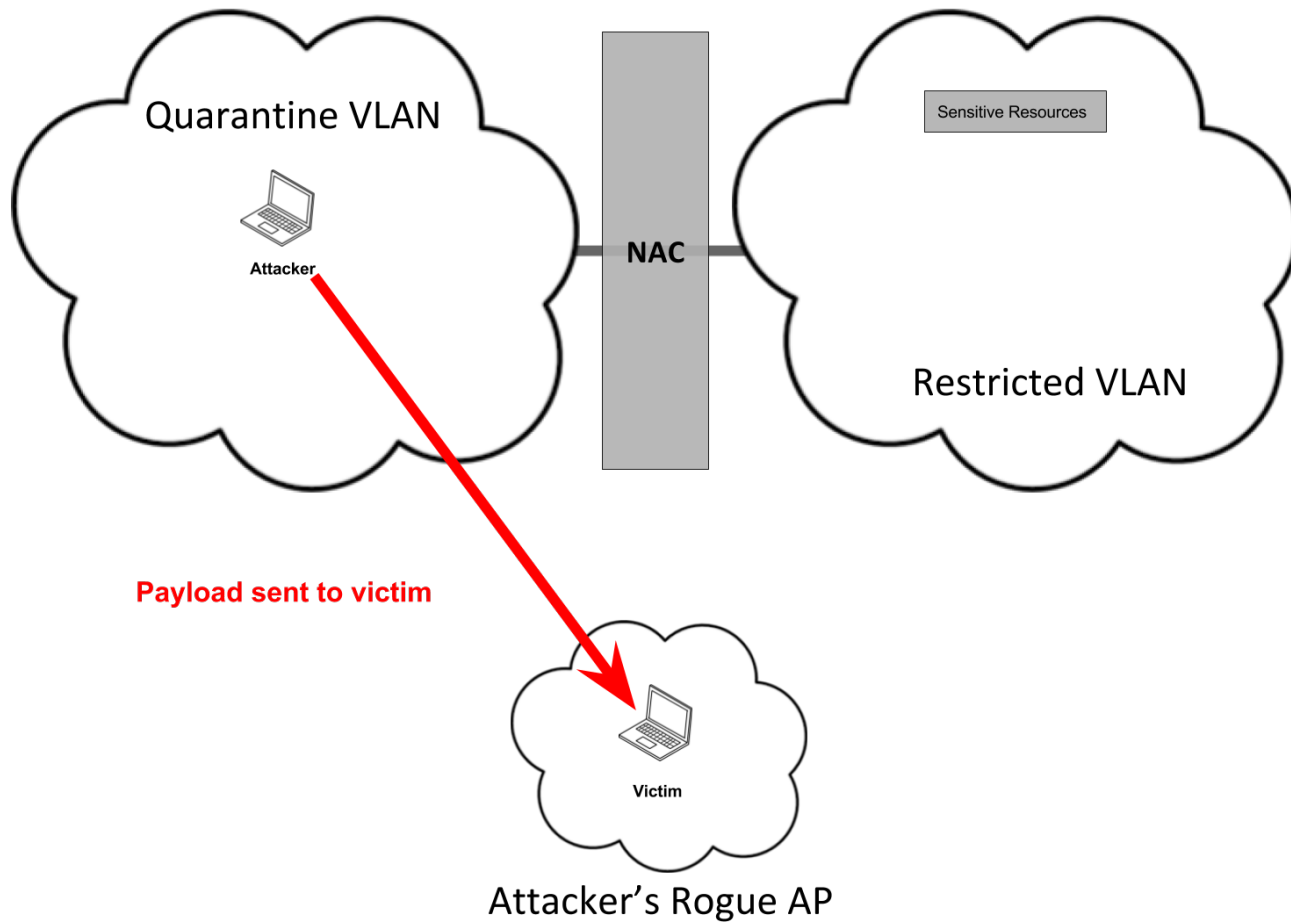


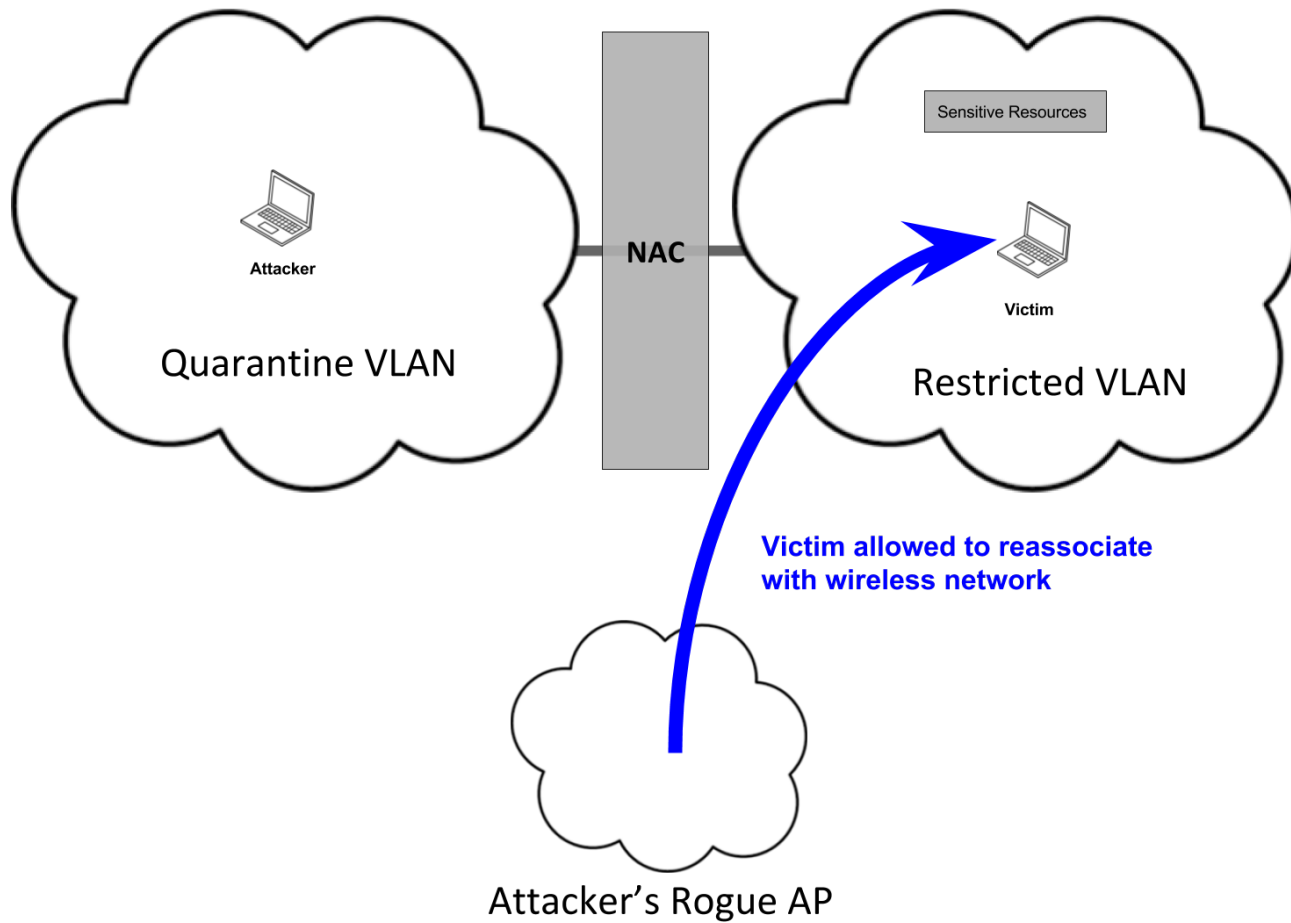


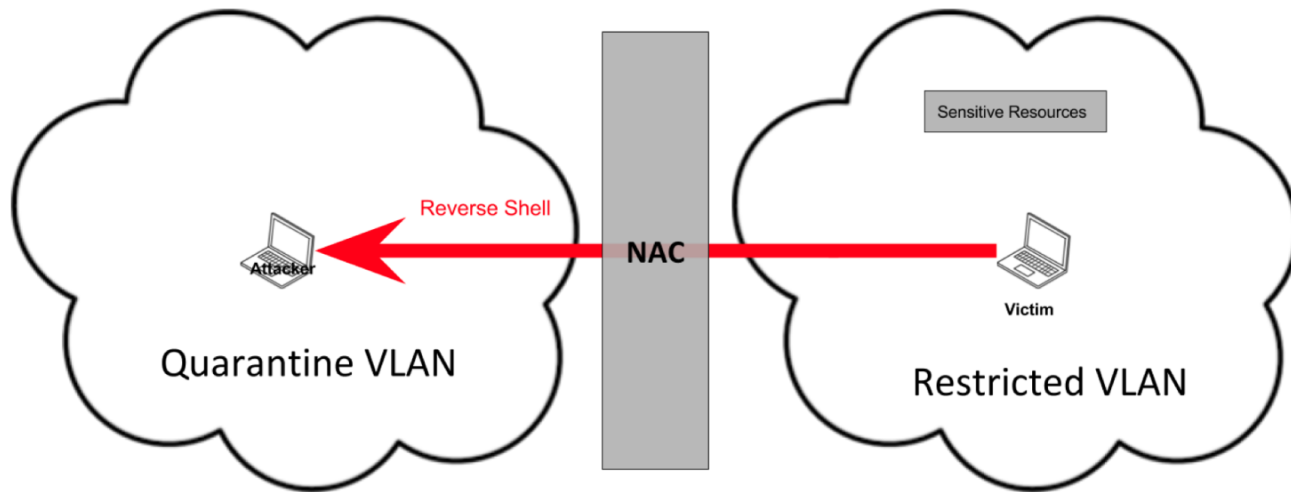


tick tick tick tick...

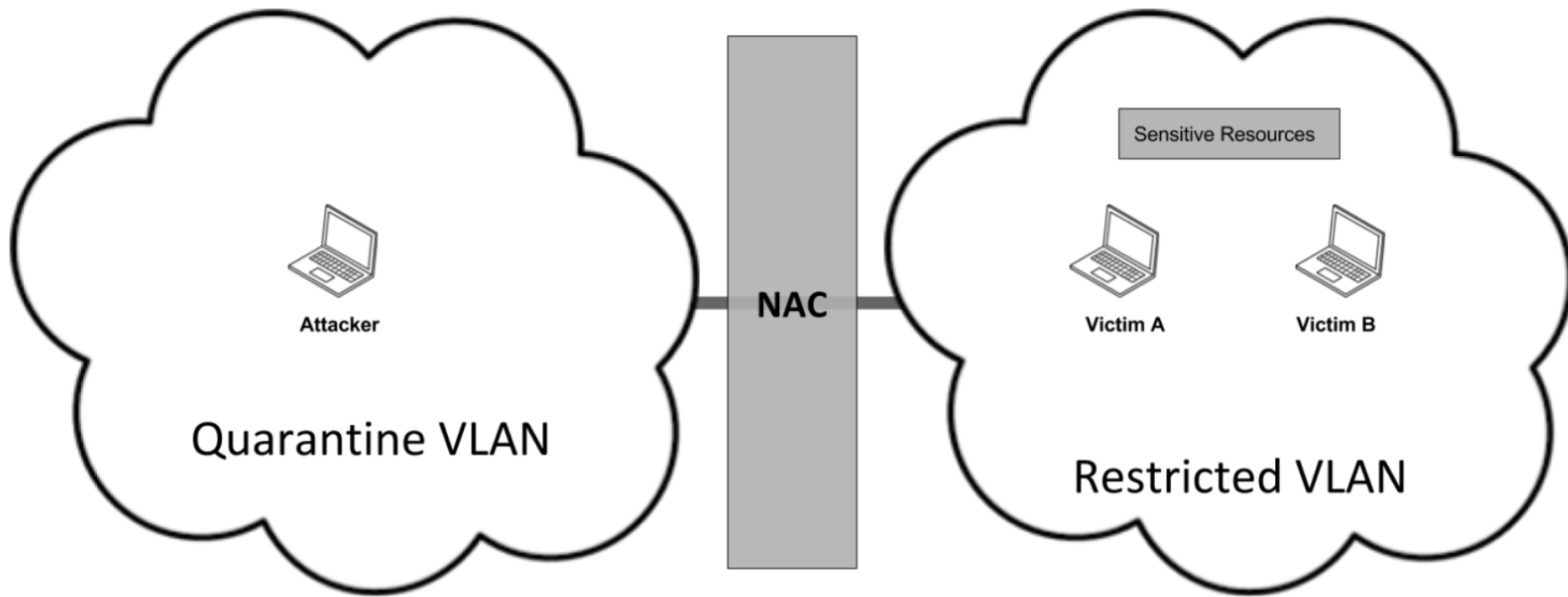
Hashes cracked offline...

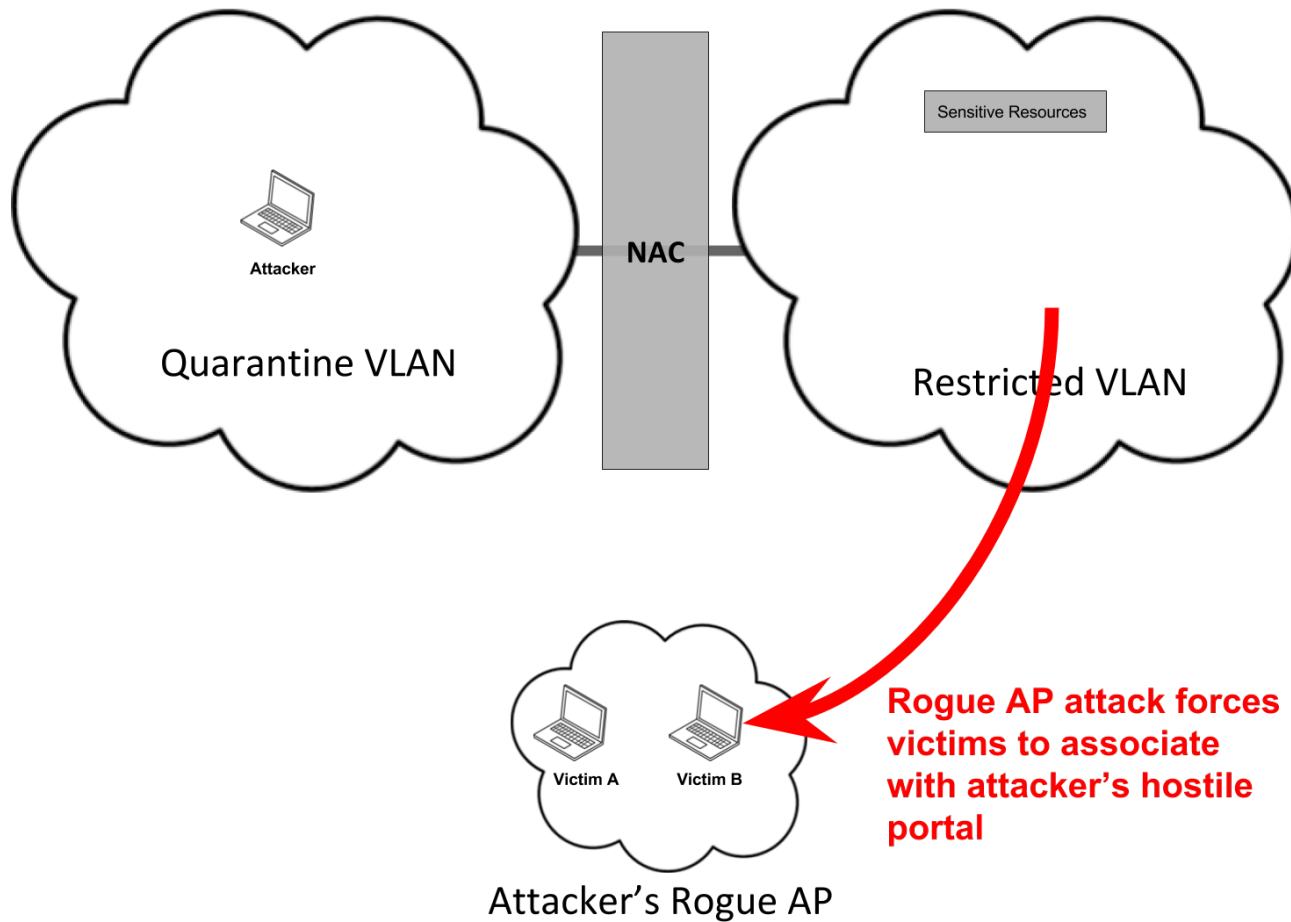


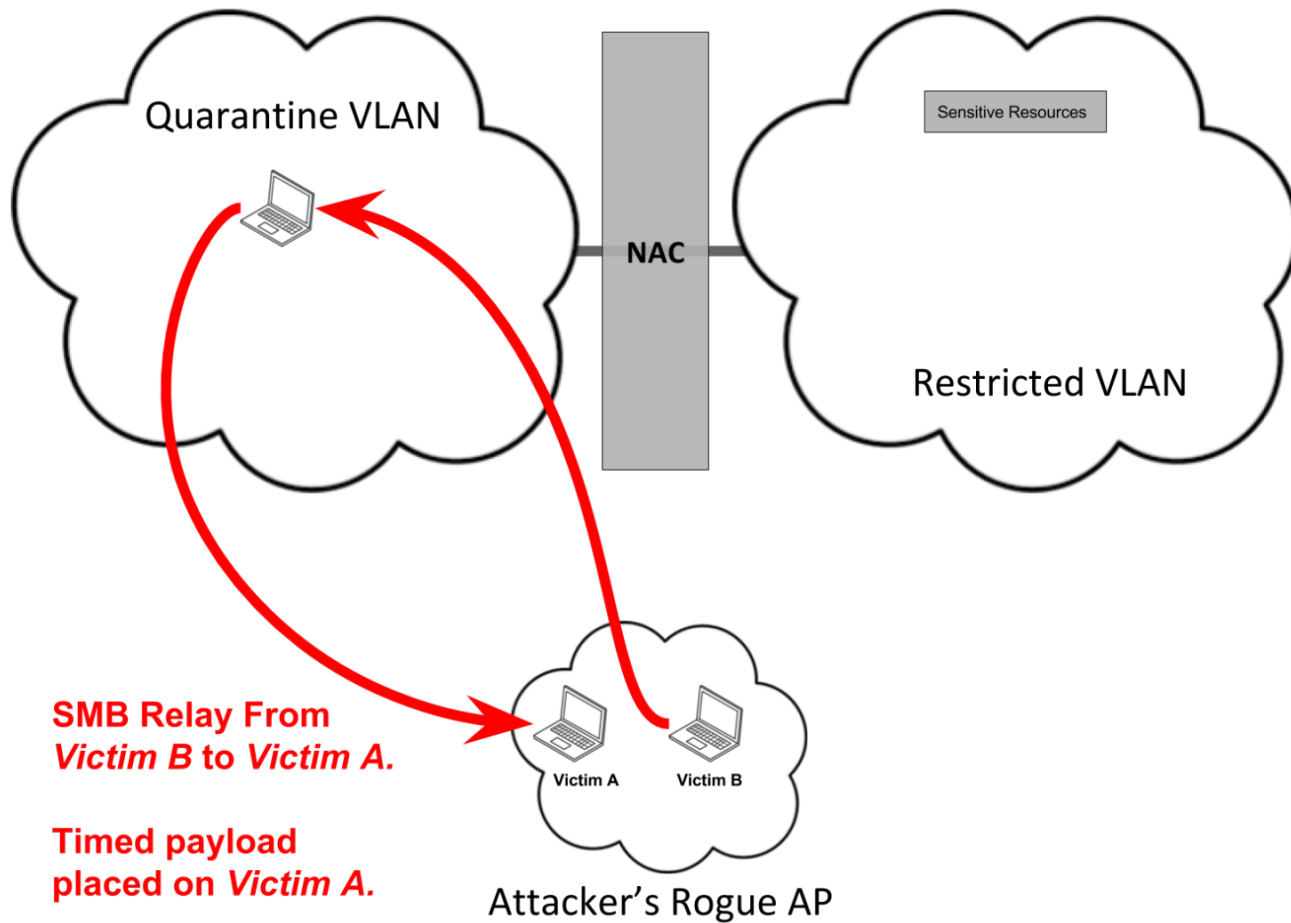


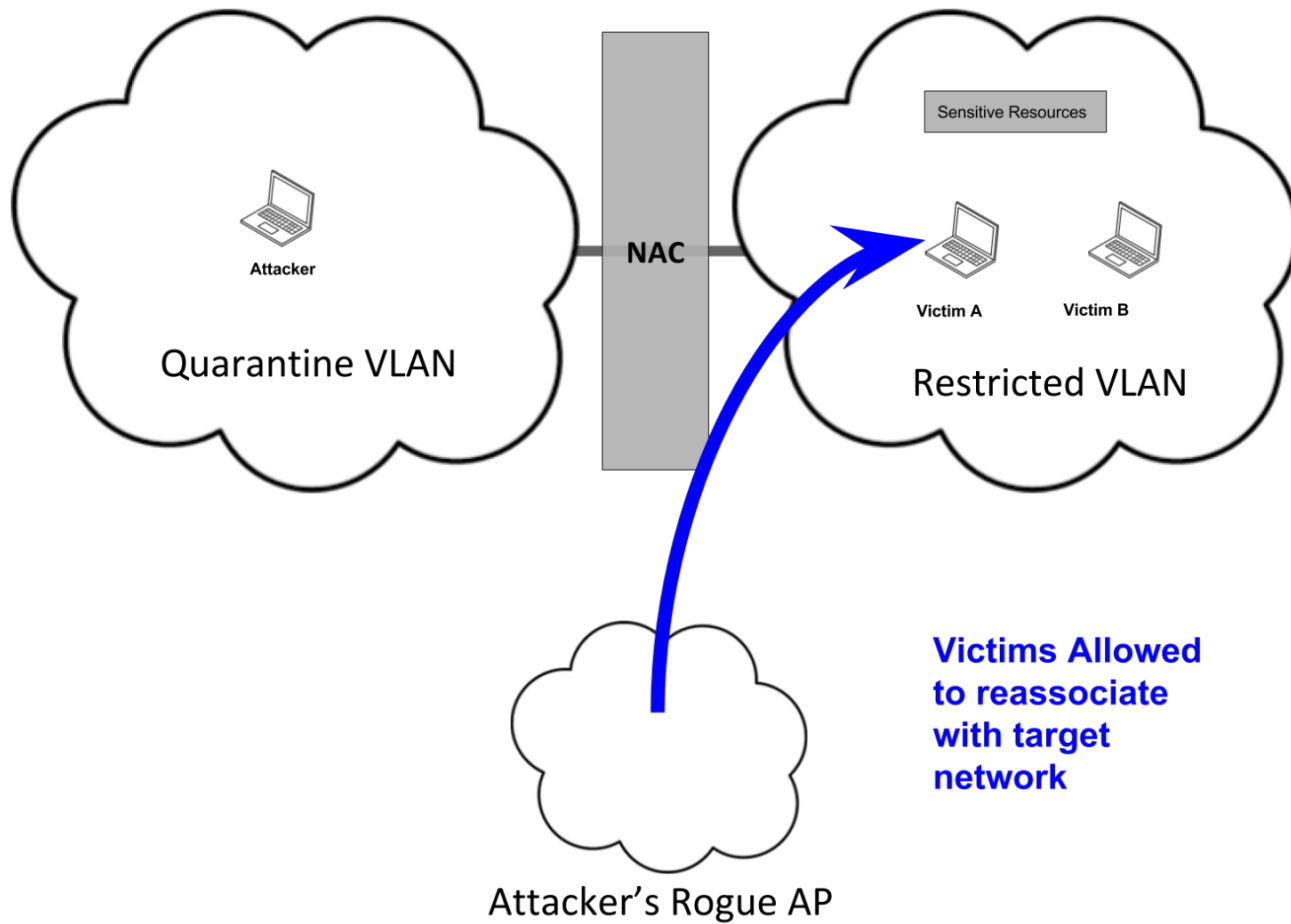


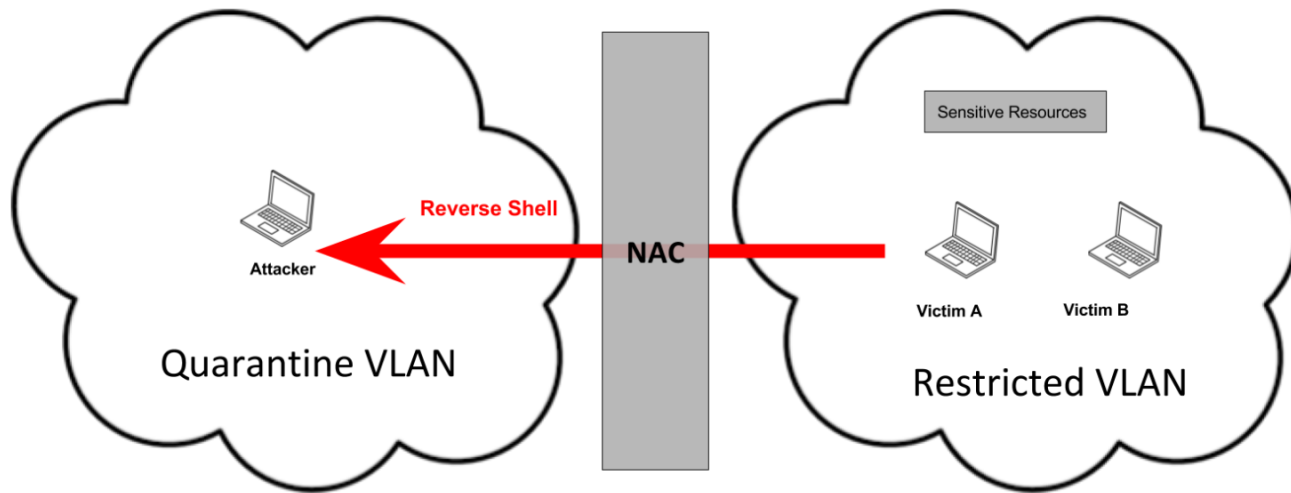
Better approach: SMB Relay











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 !!!!)

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:

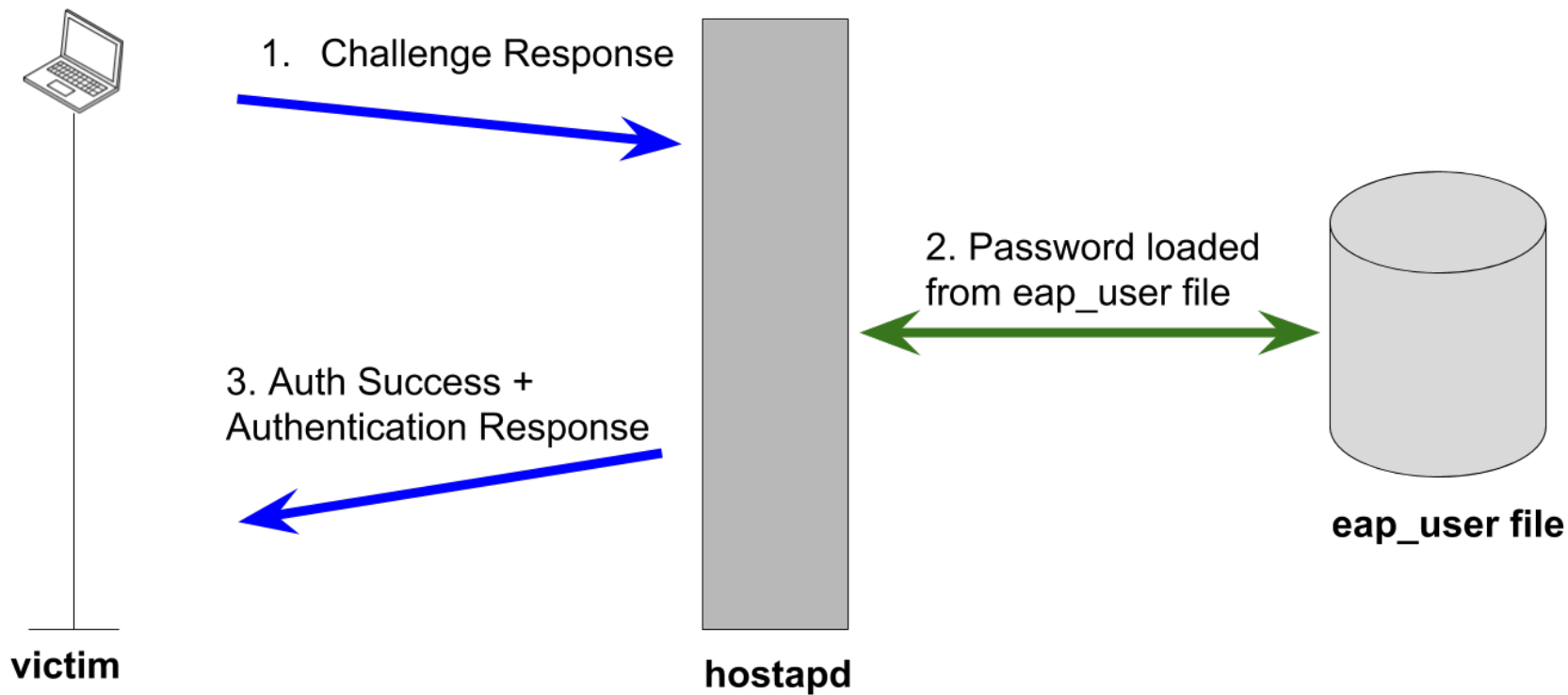
What this means:

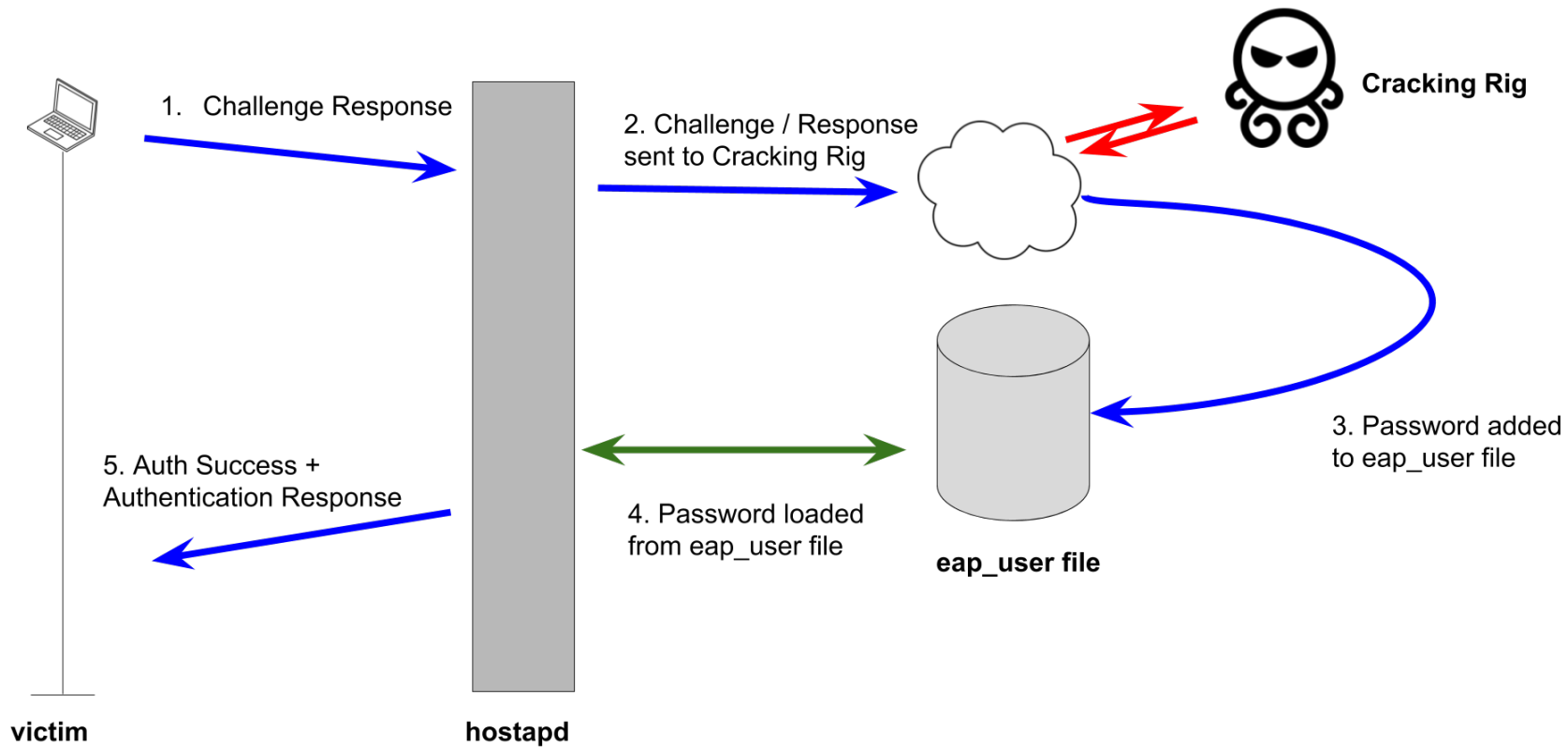
- Although the attacker can force the victim to authenticate with an evil twin to steal hashes, the attacker's RADIUS server will fail the final the final stage of the authentication process and the client will not associate with the attacker [29].

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





Auto Crack 'N Add (Dominic White & Ian de Villiers)

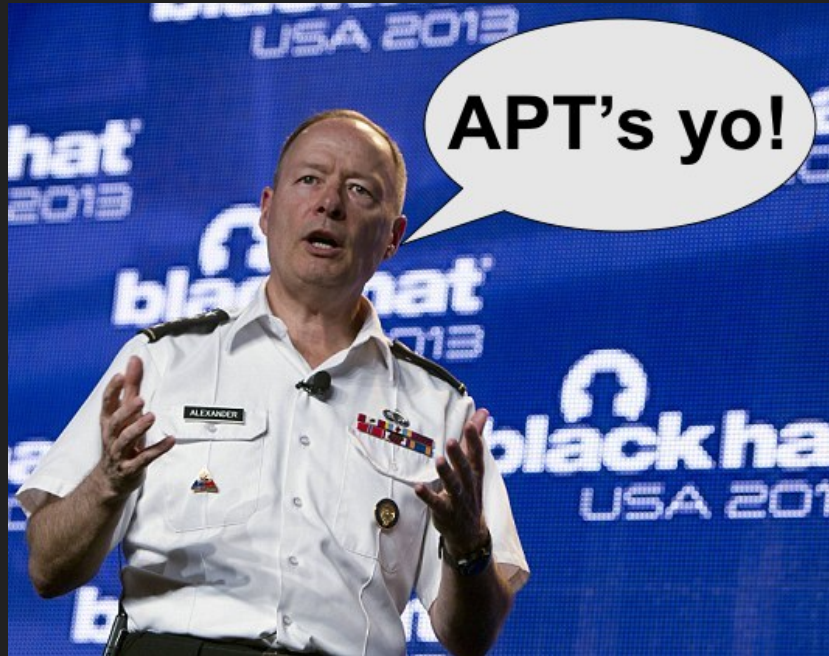
Second Option: Crack offline, Pwn later

No caveats other than time.

- Lifecycle of the attack now takes place over the course of a week, rather than an hour.

Still effective:

- Time-frame actually isn't bad from the perspective of a sophisticated attacker, and real attackers aren't time boxed.
- Dead-simple attack to perform



DEMO

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.

github.com/s0lst1c3/eaphammer

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_Shmocon2008_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-NJyZfYMtj72S6WDIQPyl7vgQYy14fu-5t3mssAfFhmQo_bl6OYyqK5dENUGHee-40daHWqAem3m2dWJd6jNeuP9ZSnaezoRkarq_s8J92z3SJMEXdxdAUkF1nMzRoaCPeG2anVCQ1tSxB8Uupviji6Pom1xr10aRuISitMk7bfMmAQ00VBESXW7IWkM1veZMINA24NpcKkmcdvF3u_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](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](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/>