
# Wi-Fi Threats and Countermeaures

Gopinath KN (Gopi)

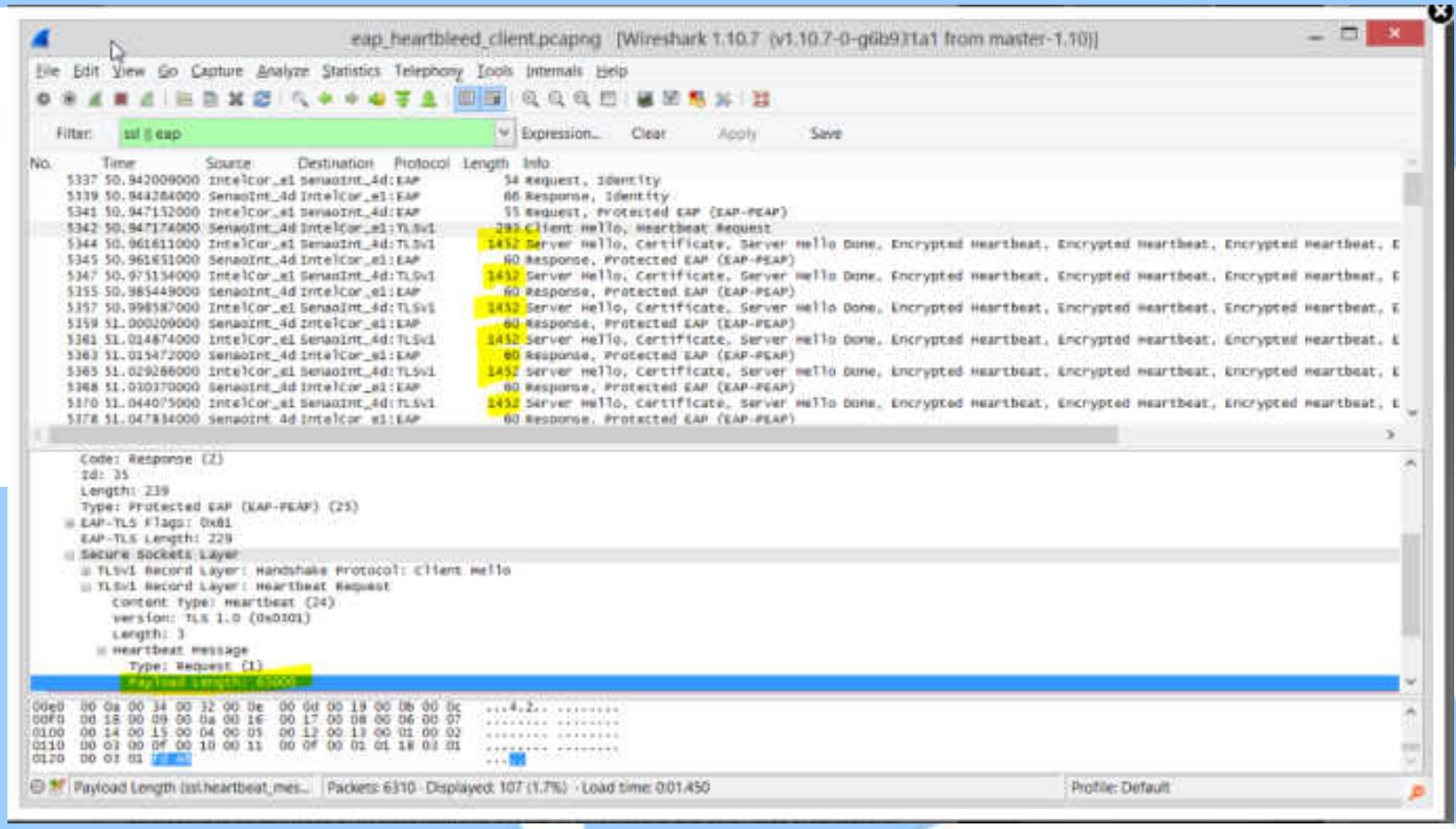
AirTight Networks
Secure Cloud-Managed Wi-Fi
http://airtightnetworks.com/

# Wi-Fi Security: Hot Off the Press, Jun 2014

## Cupid – a variant of OpenSSL Heartbleed bug in the Wi-Fi World

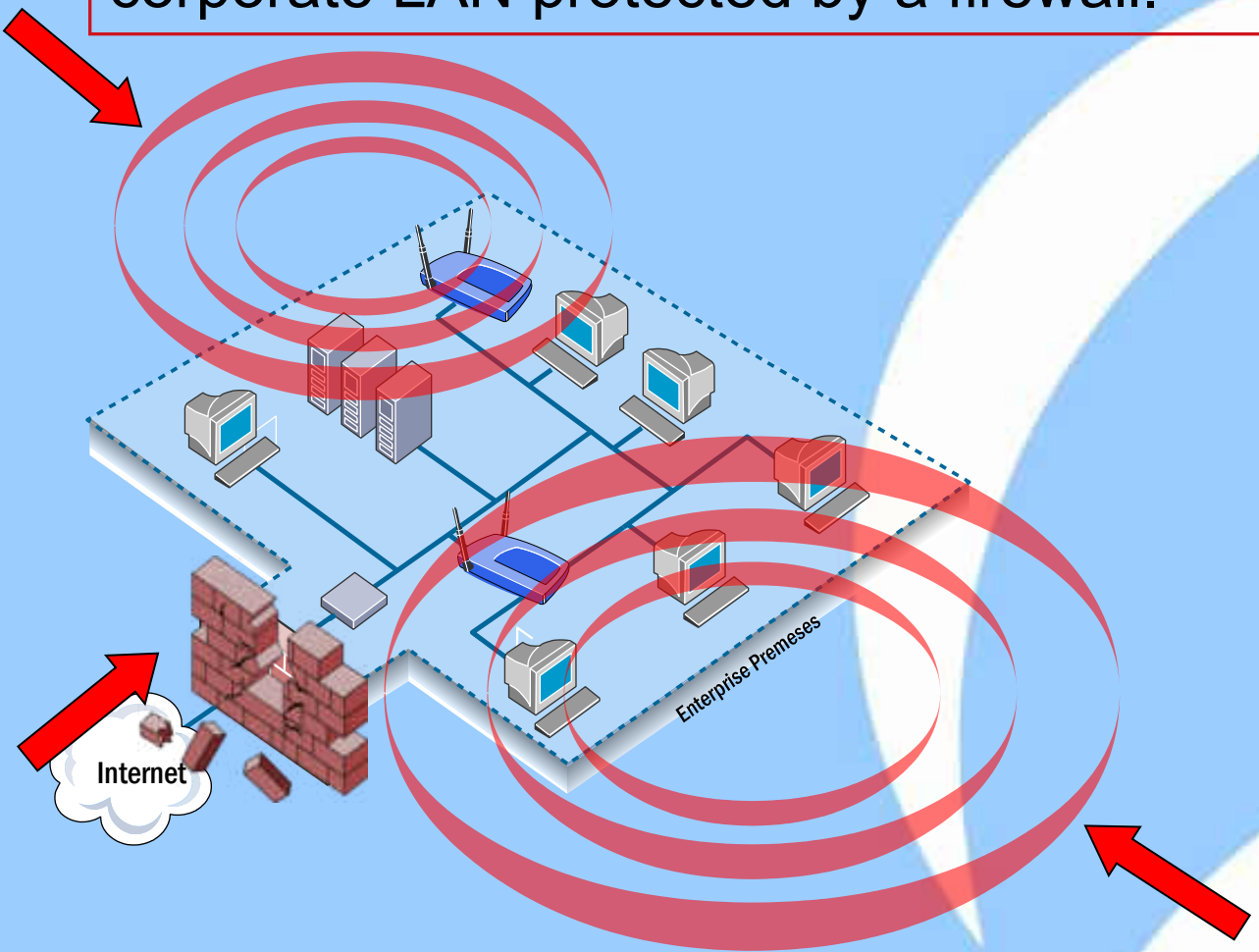
http://arstechnica.com/security/2014/06/meet-cupid-the-heartbleed-attack-spawns-evil-wi-fi-networks/

Meet "Cupid," the Heartbleed attack that spawns "evil" Wi-Fi networks



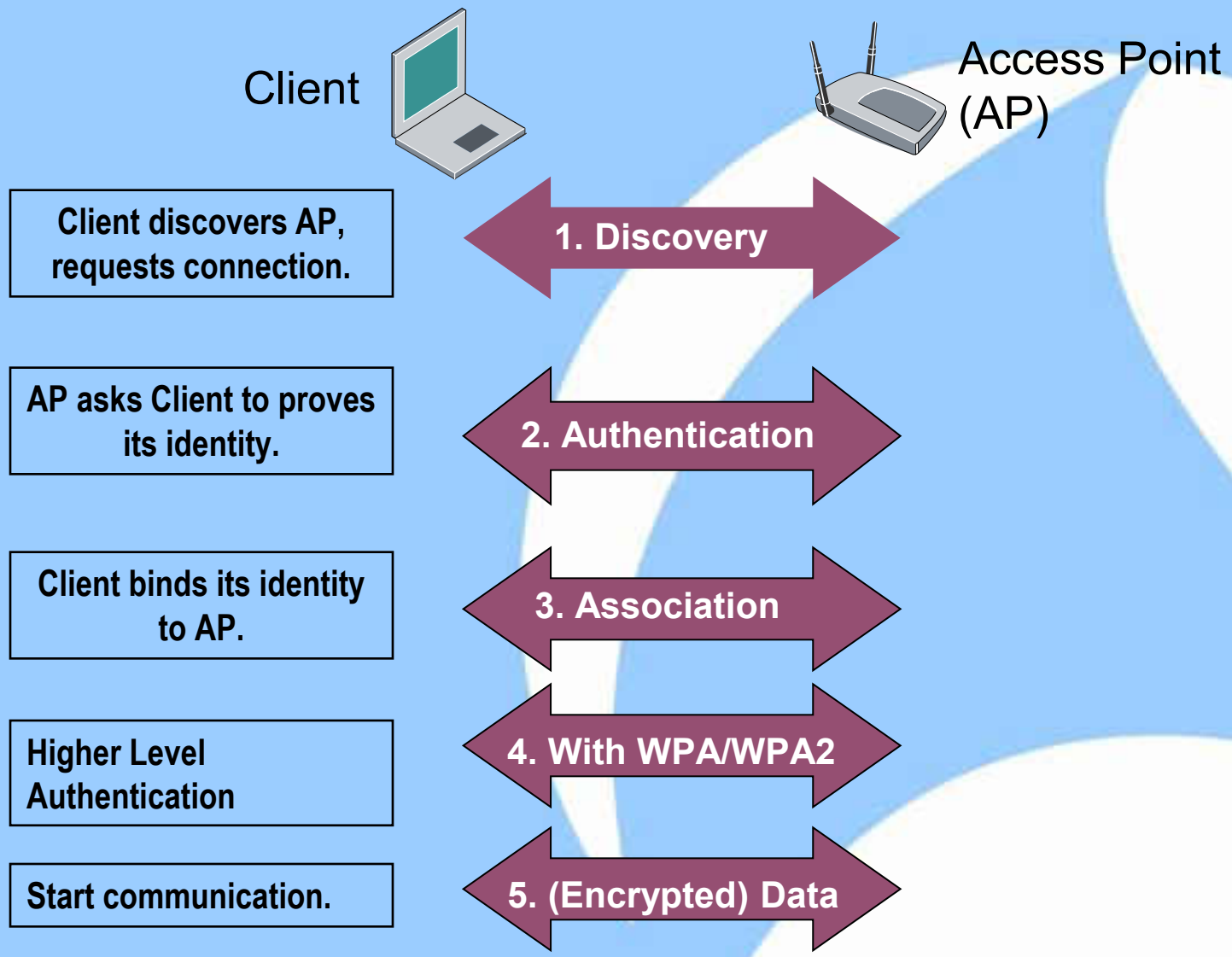
# Wireless LAN Security Trivia

**Myth:** My wireless LAN is secure as it is attached to the corporate LAN protected by a firewall.

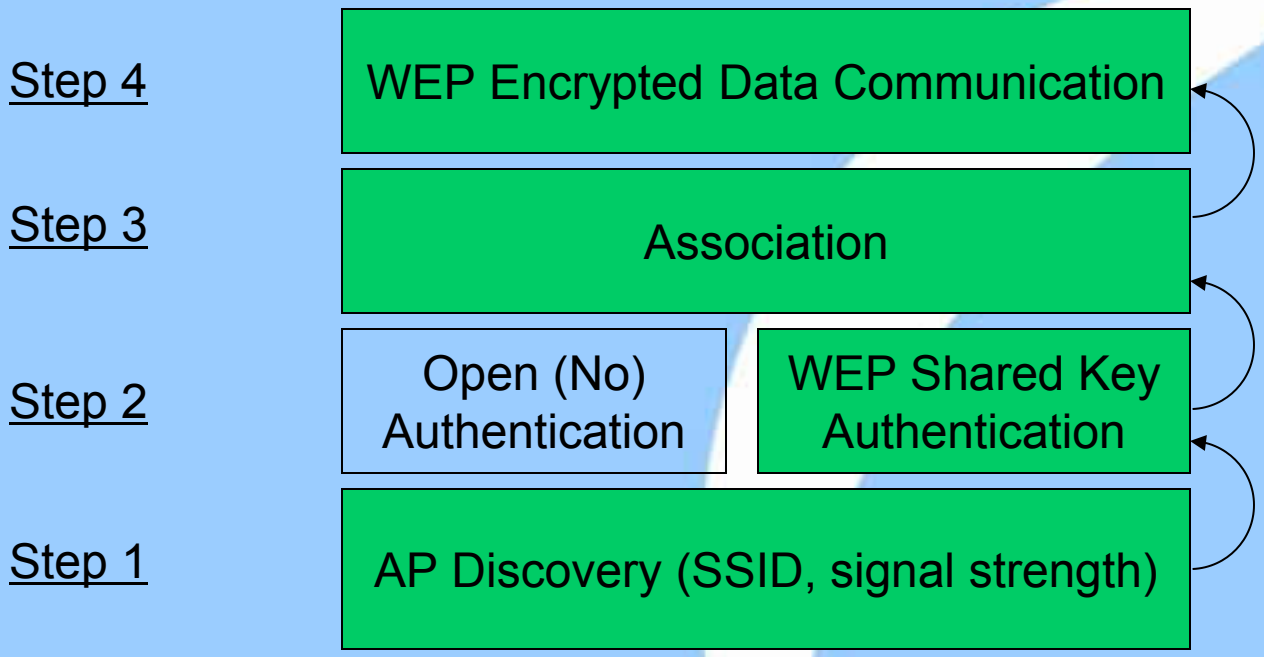


# Authorized WLAN Security

# Background: Stages of establishing a WiFi connection

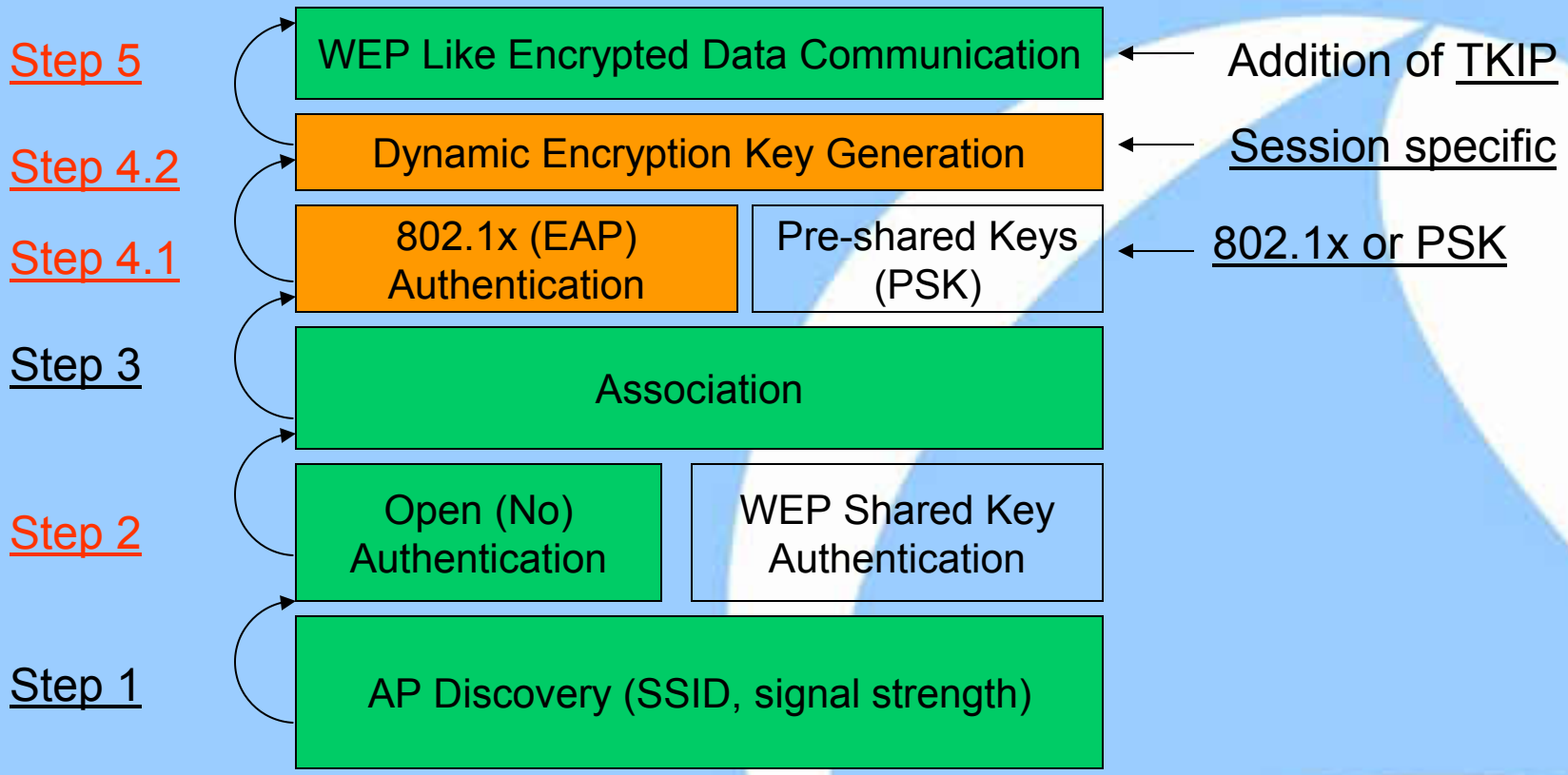


# Stages of establishing a WEP-encrypted WiFi connection



# WEP is broken. Let's move on!

# Stages in establishing a WPA-encrypted WiFi connection



# Pre-Shared Key (PSK) authentication & TKIP Encryption

- In PSK
  - Master keys are pre-configured in Client and AP
  - Encryption keys are derived using EAPOL 4-way handshake
  - Authentication Server is not needed
- TKIP
  - Band-aid on top of “WEP”

# PSK vulnerability

- In WPA the master key is used to generate transient session keys
- With PSK, all devices are configured with the same passphrase (or password) that serves as the master key
- Like any other password, the strength of the passphrase determines if it can be guessed using a dictionary attack
  - Once passphrase is guessed, an attacker can generate transient keys to decrypt all traffic
- WPA-PSK and WPA2-PSK (also known as WPA-Personal, WPA2-Personal) are vulnerable to dictionary attack

# Cloud Service for WiFi Cracking


If using WPA/WPA2 - PSK

Use a password with **at least eight** characters long and mix of **alphanumeric and special characters**

# TKIP was considered safe enough

- RSA Security White Paper, “The Wireless Security Survey of New York City”, October 2008 says:

> **“While WPA1 was designed as a temporary replacement for WEP until WPA2 arrived, it would be incorrect to state that its security level is inferior to that of WPA2: Over the years of practical use, no exploitable WPA1-specific vulnerabilities have been discovered that are not present within WPA2.”**

- According to Payment Card Industry (PCI) Data Security Standard, version 1.2, October 2008:

**Upgrade to WPA from WEP suffices to achieve PCI compliance.**

# TKIP vulnerability exposed for the first time

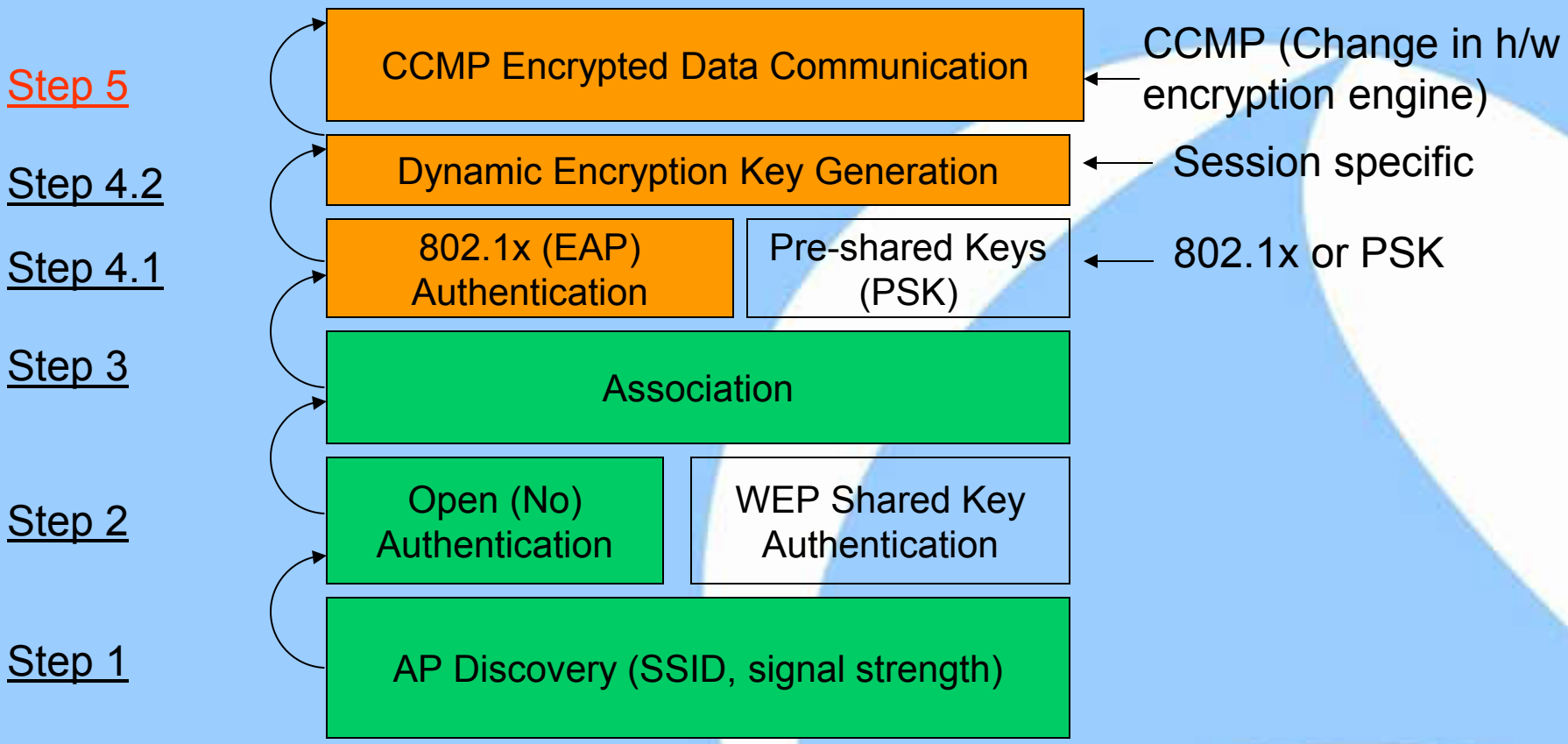
Erik Tews and Martin Beck Demonstrated at PacSec, Japan, Nov 2008

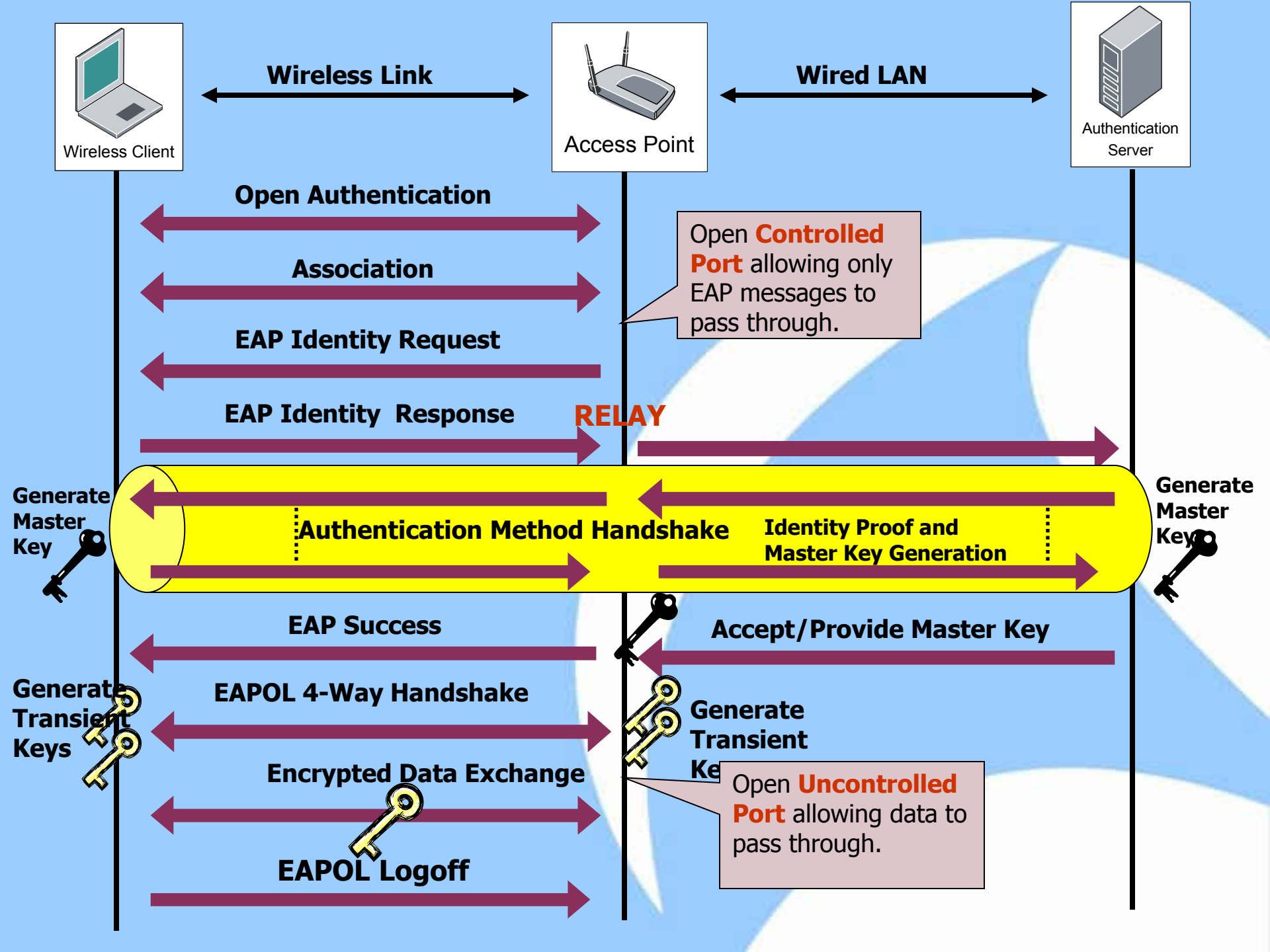
- For further technical details refer to:
  - Tkiptun-ng documentation: http://www.aircrack-ng.org/doku.php?id=tkiptun-ng
  - AirTight Knowledge Center

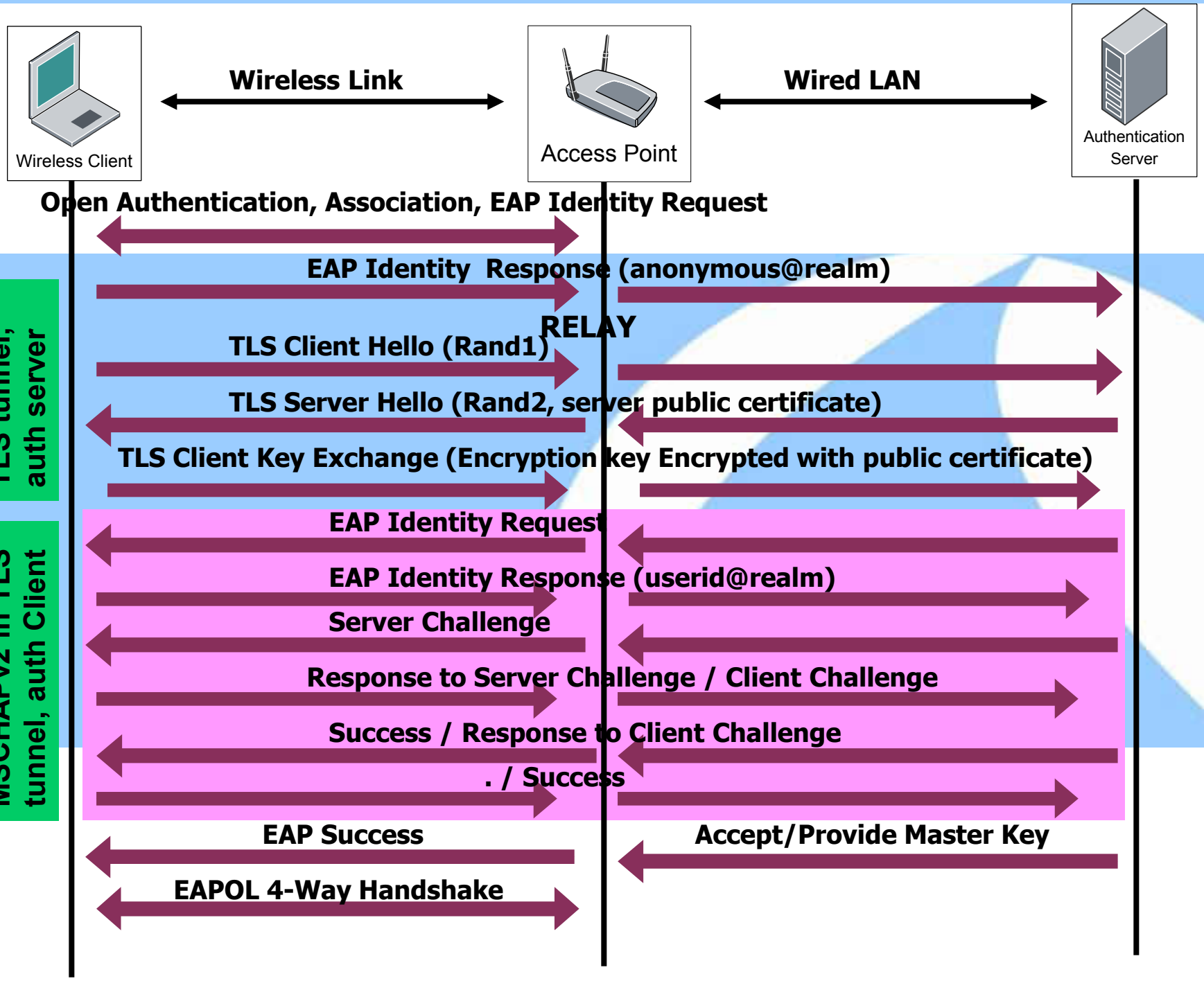
  http://www.airtightnetworks.com/home/resources/knowledge-center/wpa-wpa2-tkip-attack.html

**Wi-Fi Alliance disallows the use of TKIP in high speed networks (e.g., 802.11n, 802.11ac)**

# Stages in establishing a WPA2 (802.11i) encrypted WiFi connection




Wireless Client
Wireless Link
Access Point
Wired LAN
Authentication Server
Open Authentication
Association
EAP Identity Request
EAP Identity Response
RELAY
Open Controlled Port allowing only EAP messages to pass through.
Generate Master Key
Authentication Method Handshake
Identity Proof and Master Key Generation
Generate Master Key
EAP Success
Accept/Provide Master Key
Generate Transient Keys
EAPOL 4-Way Handshake
Generate Transient Ke
Encrypted Data Exchange
Open Uncontrolled Port allowing data to pass through.
EAPOL Logoff


Wireless Client
Wireless Link
Access Point
Wired LAN
Authentication Server
Open Authentication, Association, EAP Identity Request
EAP Identity Response (anonymous@realm)
RELAY
TLS Client Hello (Rand1)
TLS Server Hello (Rand2, server public certificate)
TLS Client Key Exchange (Encryption key Encrypted with public certificate)
Phase 1: Est. TLS tunnel, auth server
EAP Identity Request
EAP Identity Response (userid@realm)
Server Challenge
Response to Server Challenge / Client Challenge
Success / Response to Client Challenge
. / Success
Phase 2: MSCHAPv2 in TLS tunnel, auth Client
EAP Success
Accept/Provide Master Key
EAPOL 4-Way Handshake

# 802.1x example: Protected Extensible Authentication Protocol (PEAP)

- PEAP is a popular authentication method supported over 802.1x
  - Supported in Windows XP, Windows Vista, Linux
- PEAP operates in 2 phases
  - Phase 1: Client authenticates the Authentication Server using TLS server certificate; builds an encrypted tunnel between Client and Authentication server
  - Phase 2: Another authentication method such as MSCHAPv2 (a two-way challenge and response password based authentication method) can be executed within this tunnel
- Word of caution: PEAP is not full-proof; depends on the configuration

**More details:** https://wiki.bc.net/atl-conf/download/attachments/12615756/PEAP_Shmoocon2008_Wright_Antoniewicz.pdf

# Summary: wireless authentication and encryption

- WEP is fundamentally broken and it cannot be fixed
  - A variety of vulnerabilities and freely available attack tools
- PSK (WPA/WPA2) is vulnerable to dictionary attacks
  - Not for enterprise class security
  - Use strong passphrase
- TKIP vulnerable
  - Not a key cracking exploit
  - Can be used (in conjunction with QoS) to inject packets
- WPA2 with AES encryption and 802.1x authentication provides best known security (with proper configuration of course!)

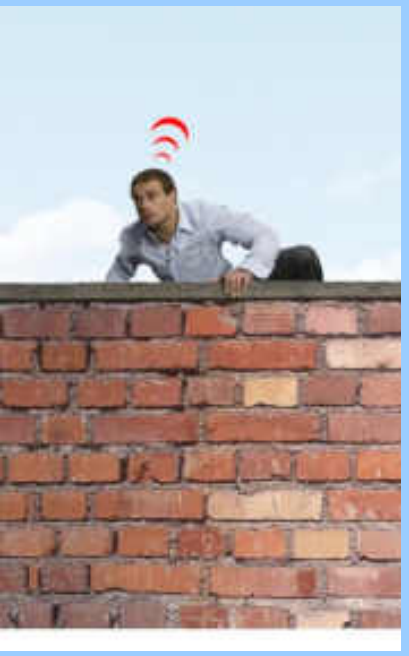
# So, Is WPA2/802.11i Sufficient for Overall enterprise WLAN security?

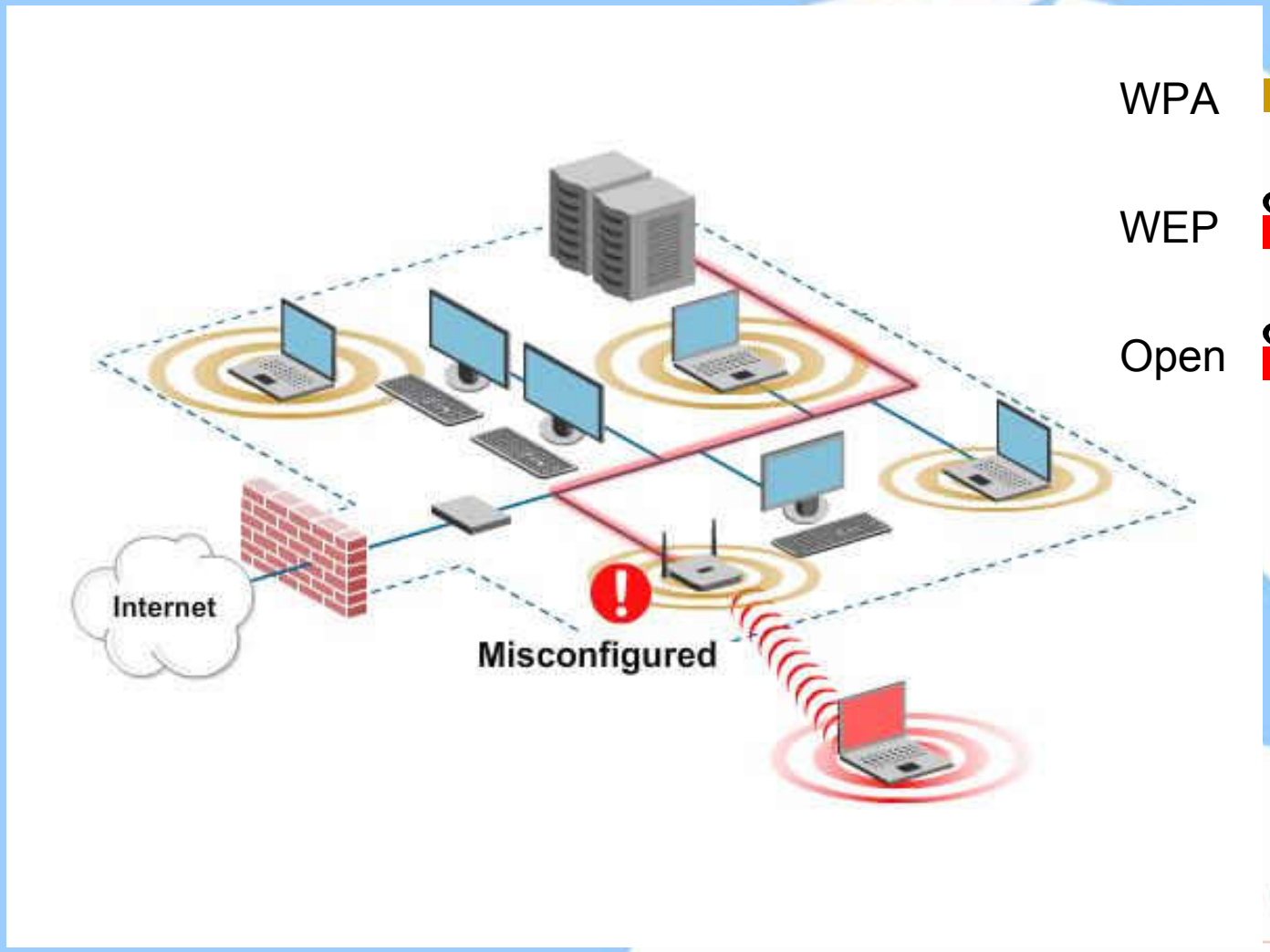
# Video

# Threats Due To Unauthorized Wi-Fi Communication

# Enterprise Security Perimeter Bypass: Five Common Scenarios

# Scenario #1: Misconfigured Devices





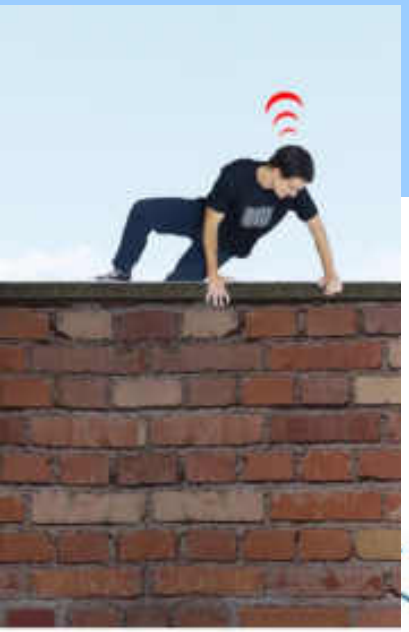
# Misconfigured AP

| | | | | | | |
|---|---|---|---|---|---|---|
| 3528 | 17.58836700 | Htc_14:8b:9b | WibhuTec_90:03:50 | 802.11 | 48 | Authentication, SN=2319, FN=0, Flags=....R... |
| 3530 | 17.58963800 | WibhuTec_90:03:50 | Htc_14:8b:9b | 802.11 | 48 | Authentication, SN=256, FN=0, Flags=........ |
| 3531 | 17.59019100 | WibhuTec_90:03:50 | Htc_14:8b:9b | 802.11 | 48 | Authentication, SN=256, FN=0, Flags=....R... |
| 3533 | 17.59181200 | Htc_14:8b:9b | WibhuTec_90:03:50 | 802.11 | 132 | Association Request, SN=2320, FN=0, Flags=........, SSID=test_ssid |
| 3535 | 17.59947600 | WibhuTec_90:03:50 | Htc_14:8b:9b | EAPOL | 151 | Key (Message 1 of 4) |
| 3581 | 17.66412000 | Htc_14:8b:9b | WibhuTec_90:03:50 | EAPOL | 173 | Key (Message 2 of 4) |
| 3584 | 17.66736800 | WibhuTec_90:03:50 | Htc_14:8b:9b | EAPOL | 207 | Key (Message 3 of 4) |
| 3587 | 17.67420100 | Htc_14:8b:9b | WibhuTec_90:03:50 | EAPOL | 151 | Key (Message 4 of 4) |
| 15606 | 35.38862000 | IntelCor_d0:29:a4 | WibhuTec_a0:24:62 | 802.11 | 44 | Deauthentication, SN=2763, FN=0, Flags=....R... |
| 16957 | 37.12531000 | IntelCor_35:f6:7e | WibhuTec_a1:b5:60 | 802.11 | 48 | Authentication, SN=336, FN=0, Flags=........ |
| 16959 | 37.12669000 | WibhuTec_a1:b5:60 | IntelCor_35:f6:7e | 802.11 | 48 | Authentication, SN=256, FN=0, Flags=........ |
| 16961 | 37.12765300 | IntelCor_35:f6:7e | WibhuTec_a1:b5:60 | 802.11 | 156 | Association Request, SN=337, FN=0, Flags=........, SSID=Social_Spectrum |
| 16968 | 37.14180200 | WibhuTec_a1:b5:60 | IntelCor_35:f6:7e | EAPOL | 173 | Key (Message 1 of 4) |
| 16970 | 37.14267500 | IntelCor_35:f6:7e | WibhuTec_a1:b5:60 | EAPOL | 191 | Key (Message 2 of 4) |
| 16971 | 37.14636200 | WibhuTec_a1:b5:60 | IntelCor_35:f6:7e | EAPOL | 207 | Key (Message 3 of 4) |
| 16973 | 37.14719400 | IntelCor_35:f6:7e | WibhuTec_a1:b5:60 | EAPOL | 151 | Key (Message 4 of 4) |
| 28491 | 53.98025700 | WibhuTec_90:03:51 | Htc_14:8b:9b | 802.11 | 44 | Deauthentication, SN=256, FN=0, Flags=........ |
| 28771 | 55.10168300 | WibhuTec_90:03:51 | Htc_14:8b:9b | 802.11 | 44 | Deauthentication, SN=256, FN=0, Flags=........ |
| 28928 | 55.65601300 | WibhuTec_90:03:51 | Htc_14:8b:9b | 802.11 | 44 | Deauthentication, SN=256, FN=0, Flags=........ |
| 29174 | 56.59162800 | WibhuTec_90:03:51 | Htc_14:8b:9b | 802.11 | 44 | Deauthentication, SN=256, FN=0, Flags=........ |
| 29195 | 56.65034100 | Htc_14:8b:9b | WibhuTec_90:03:50 | 802.11 | 48 | Authentication, SN=2488, FN=0, Flags=........ |
| 33049 | 72.24956500 | IntelCor_04:4e:3f | WibhuTec_90:03:50 | 802.11 | 48 | Authentication, SN=1360, FN=0, Flags=........ |
| 33051 | 72.25083500 | WibhuTec_90:03:50 | IntelCor_04:4e:3f | 802.11 | 48 | Authentication, SN=256, FN=0, Flags=........ |
| 33055 | 72.25872900 | IntelCor_04:4e:3f | WibhuTec_90:03:50 | 802.11 | 116 | Reassociation Request, SN=1361, FN=0, Flags=........, SSID=test_ssid |
| 33057 | 72.26108500 | WibhuTec_90:03:50 | IntelCor_04:4e:3f | 802.11 | 152 | Reassociation Response, SN=257, FN=0, Flags=........ |
| 56513 | 106.3418240 | WibhuTec_a0:27:a0 | IntelCor_16:45:3b | 802.11 | 48 | Authentication, SN=256, FN=0, Flags=........ |
| 56650 | 106.4667880 | IntelCor_16:45:3b | WibhuTec_a0:27:a0 | 802.11 | 87 | Association Request, SN=98, FN=0, Flags=........, SSID=sampl1 |
| 64977 | 117.4266720 | WibhuTec_90:03:50 | Htc_14:8b:9b | 802.11 | 48 | Authentication, SN=256, FN=0, Flags=........ |
| 64979 | 117.4281810 | Htc_14:8b:9b | WibhuTec_90:03:50 | 802.11 | 110 | Association Request, SN=2968, FN=0, Flags=........, SSID=test_ssid |
| 64980 | 117.4310590 | WibhuTec_90:03:50 | Htc_14:8b:9b | 802.11 | 152 | Association Response, SN=257, FN=0, Flags=........ |
| 64981 | 117.4324590 | WibhuTec_90:03:50 | Htc_14:8b:9b | 802.11 | 152 | Association Response, SN=257, FN=0, Flags=....R... |

WPA2

Open

# Scenario #2: Rogue Access Point



# What are different types of Rogue APs

- Various permutations and combinations of
  - Bridging APs (on subnets coinciding with or different from wired interface address)
  - Router (NAT) APs (with and without MAC cloning)
  - APs with encrypted wireless links
  - APs with open wireless links
  - Soft APs (natively configured on wireless client or which use external devices such as USB sticks)

# Windows 7 Virtual AP
# Evolution of Wi-Fi support on laptops

### Traditional Wi-Fi

Operate as client/ad-hoc

### First Gen “Soft AP”

Convert laptop into AP

But, single function: Can operate either as AP OR client/ad-hoc

### Windows 7 Virtual WiFi – The Next Gen Soft AP

Can operate as Soft AP and Client/Ad-hoc simultaneously

# Windows 7 Soft AP: A User's Delight

- No new hardware/software needed
- Connect to two different wireless networks with a single card
- One virtual interface acts as a client
- Easy to configure the other interface as an AP or a client
- Configure other virtual interface in AP mode to
  - Form a personal wireless network with PDAs and other devices
  - Share Internet
  - Extend the range of an AP by introducing a hop

# Scenario #3: Uncontrolled Clients

**BYOD**

**Authorized Client Extrusions**

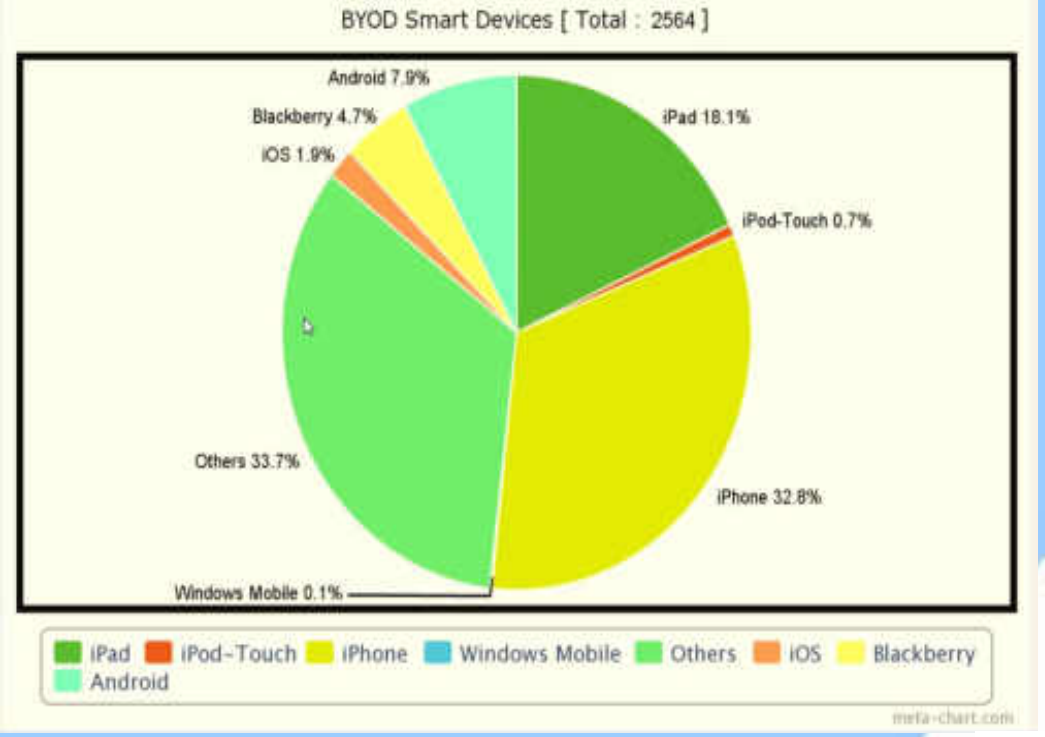
# BYOD


## A Wireless Tsunami of Devices

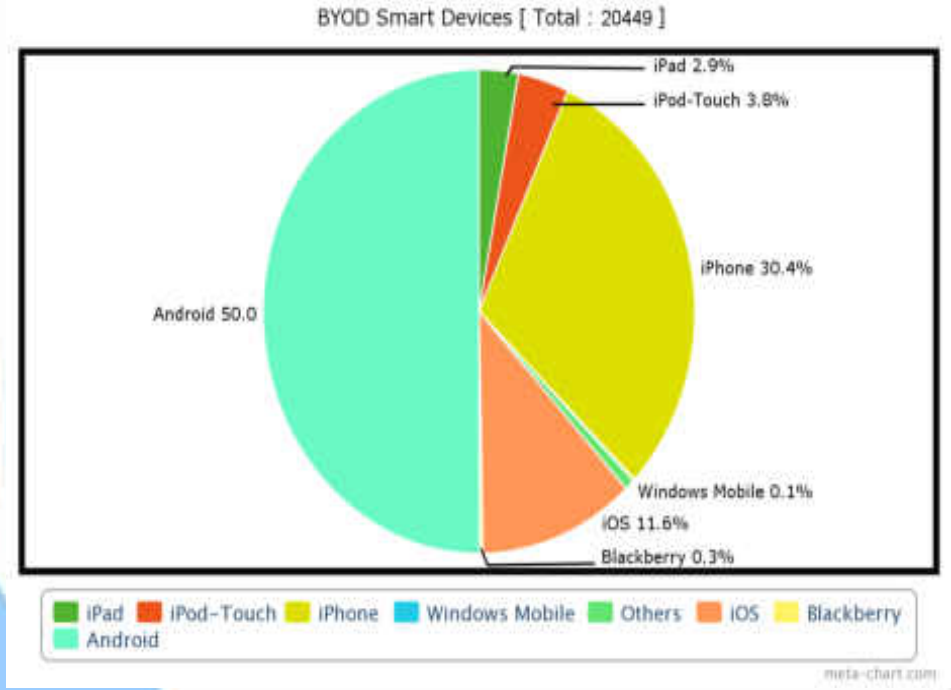
# Managing the “Unmanaged”

WPA2/802.1x cannot prevent unauthorized devices from accessing the enterprise network

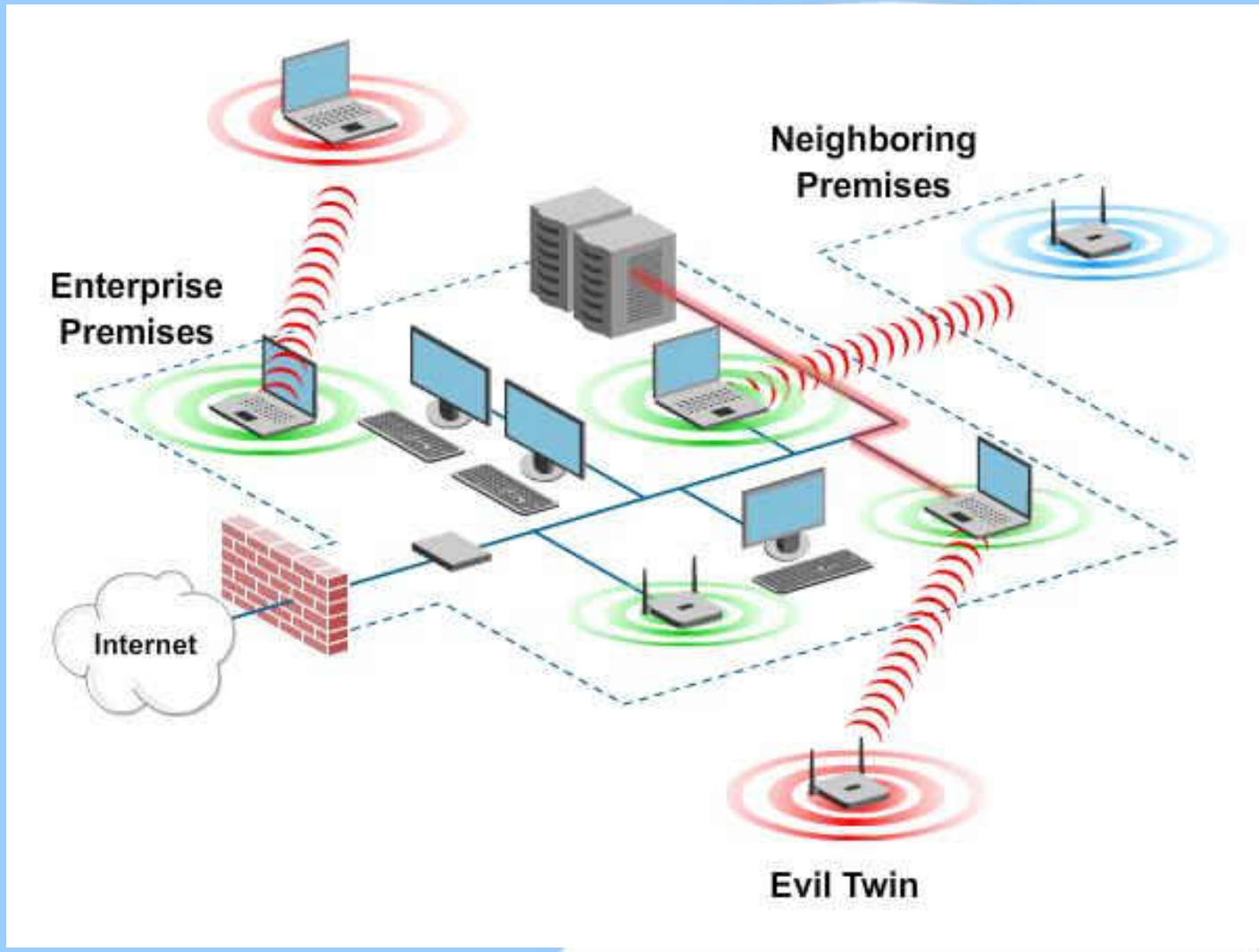





# Real-life Examples: BYOD is rampant!



# Client Extrusions (Mis-associated Clients)



# Misassociations: Deliberate or unwitting connections to external APs

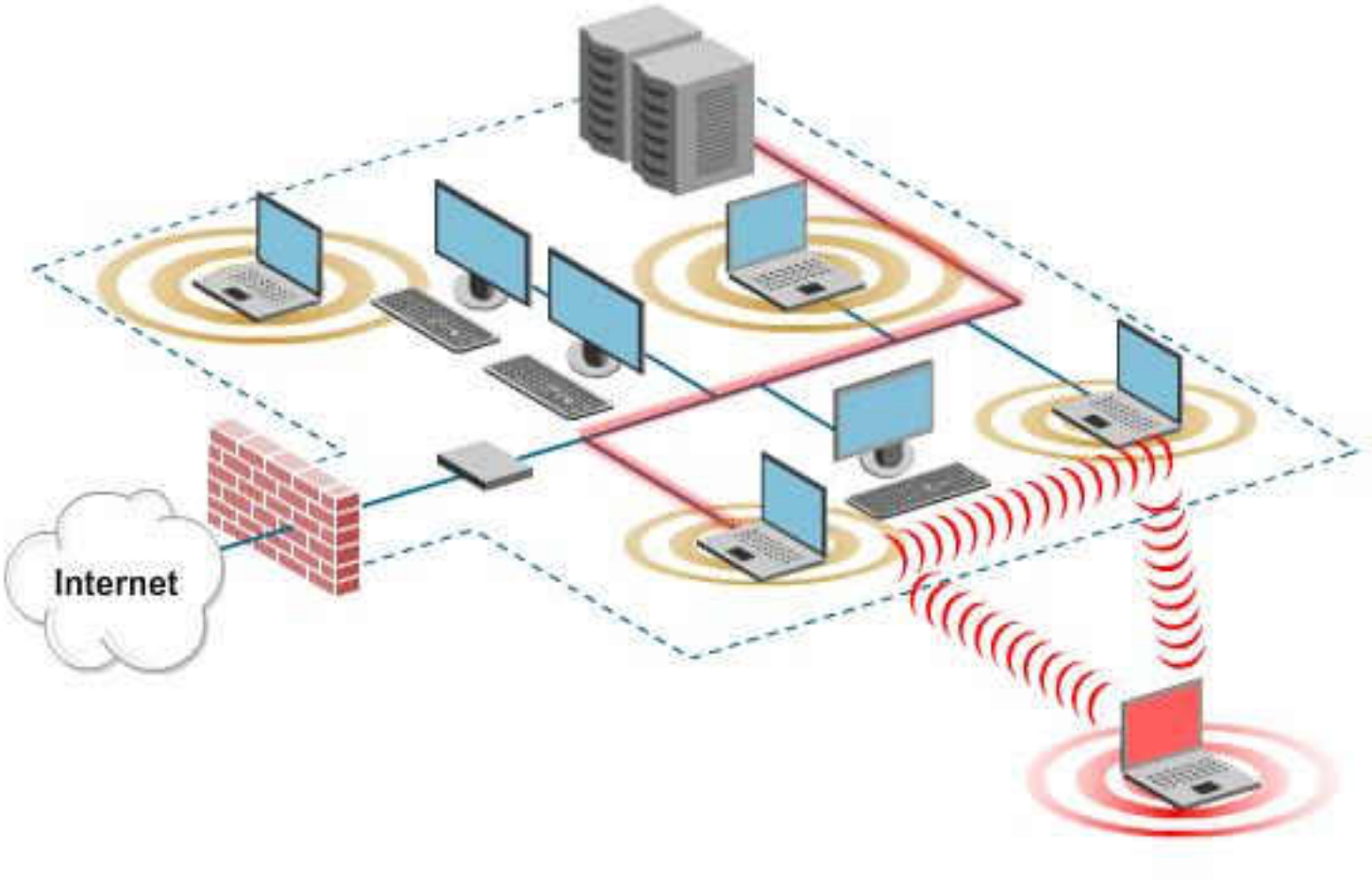
- Deliberate
  - Employees get enticed to connect to Open external APs
    - Unprotected APs in the neighborhood, Hotspots
- Unwitting
  - Windows wireless connection utility caches earlier connected networks
  - Actively seeks to connect to those networks later
    - Most common with default SSIDs (linksys, default) and hotspot SSIDs (tmobile, GoogleWiFi)
- Traffic over such connections bypasses enterprise security controls

# Mis-associations: Evil-Twin Attack

- An attacker sets up an AP that advertises SSID which is being probed by WiFi clients or that advertises SSID of a nearby enterprise or hotspot
- Induces WiFi clients into connecting to it
- Can launch variety of attacks after connection is established
  - Stealing sensitive corporate data
  - Man-in-the-middle/Wi-Phishing
  - Scanning the laptop for vulnerabilities (e.g., Metasploit)
- Honeypot attack tools are freely available over Internet
  - KARMA, Delegated
- Can be easily carried out using just a Smartphone!
  - "Smartpots" (http://www.marketwired.com/press-release/Smartphone-as-Attacker-AirTight-Demos-SmartPots-CSI-2010-Next-Generation-Wi-Fi-Attacks-1341134.htm)


© 2009 Queenconcept
Today, This is all you need!

# Scenario #4: Ad Hoc Networks



# “Known” Vulnerable SSIDs Probed For

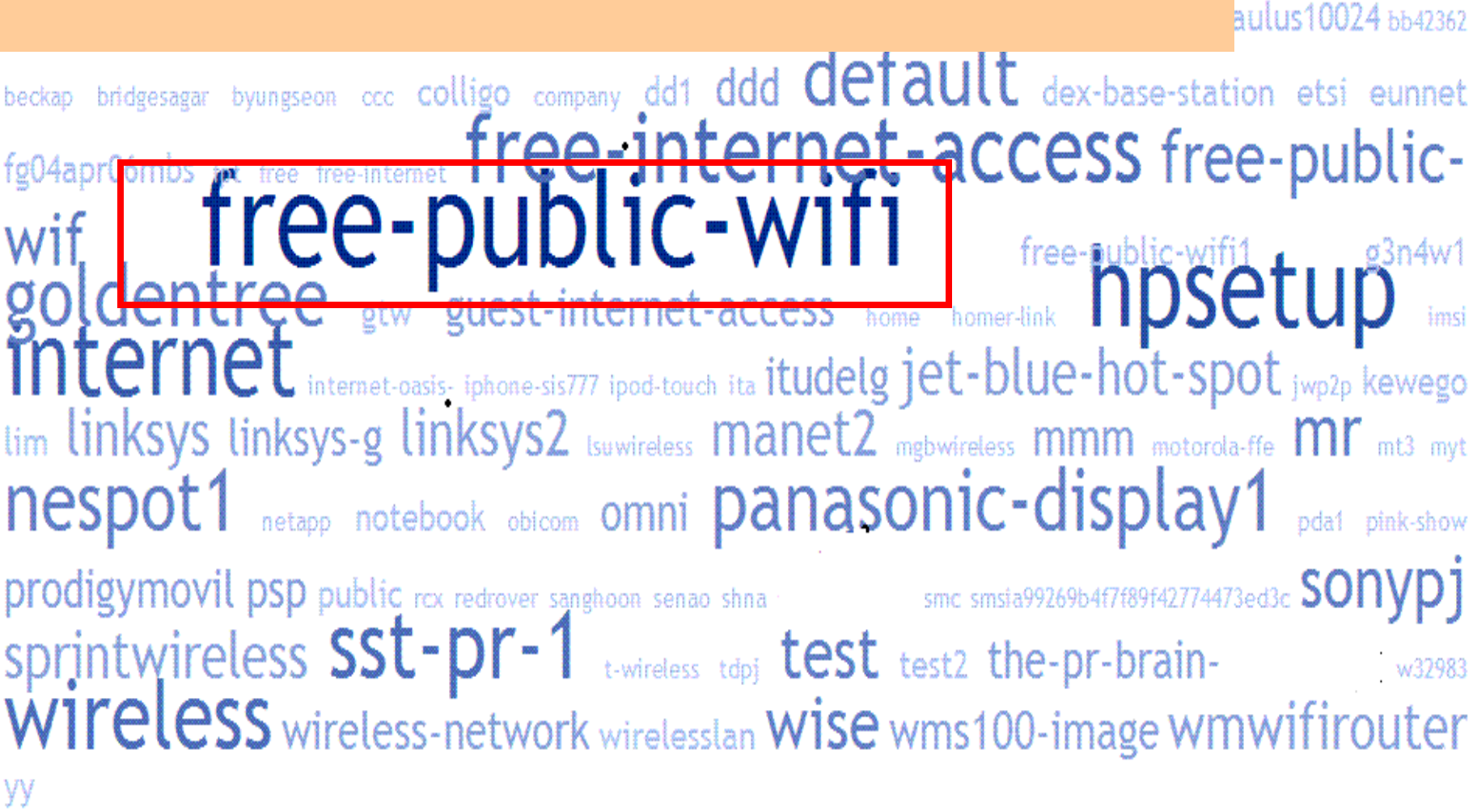
103 distinct SSIDs recorded


Certain (8%) Authorized Clients Probing for 5 or more SSIDs

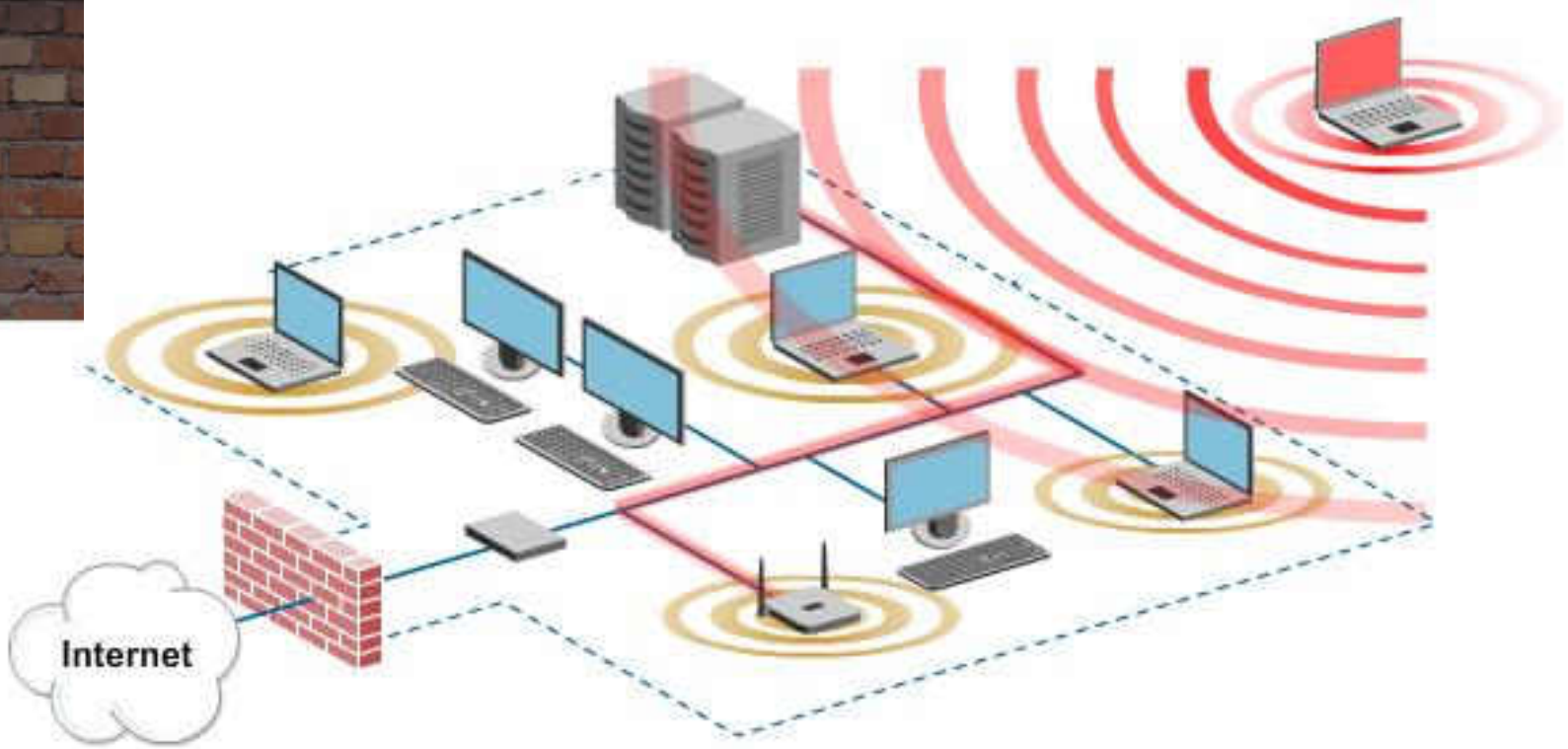
# Adhoc Authorized Clients!

565 distinct Adhoc SSIDs found, About half of them Vulnerable

15% of these are default SSIDs. 26,443 (7%) clients in adhoc mode.



# Scenario #5: War Driving, DoS, Hacking Tools



# DoS By Disassociation Flood

| | | | | | | |
|---|---|---|---|---|---|---|
| 23409 | 38.42605500 | WibhuTec_41:71:f0 | Broadcast | 802.11 | 311 | Beacon frame, SN=67, FN=0, Flags=........, BI=100, SSID=dav_wpa |
| 23412 | 38.43257800 | WibhuTec_41:71:f3 | Broadcast | 802.11 | 419 | Beacon frame, SN=3071, FN=0, Flags=........, BI=100, SSID=dav_open |
| 23413 | 38.43632400 | WibhuTec_d0:38:01 | Broadcast | 802.11 | 302 | Beacon frame, SN=1353, FN=0, Flags=........, BI=100, SSID=LSDK_WPA2_an |
| 23437 | 38.47897100 | WibhuTec_41:71:ff | Broadcast | 802.11 | 174 | Disassociate, SN=3739, FN=0, Flags=........ |
| 23441 | 38.48951900 | Pathscal_d0:05:c0 | Broadcast | 802.11 | 325 | Beacon frame, SN=1005, FN=0, Flags=........, BI=100, SSID=NAT_TS |
| 23442 | 38.49138700 | WibhuTec_d0:33:c0 | Broadcast | 802.11 | 355 | Beacon frame, SN=3721, FN=0, Flags=........, BI=100, SSID=vap1_open |
| 23450 | 38.51418700 | WibhuTec_41:71:ff | Broadcast | 802.11 | 174 | Disassociate, SN=3746, FN=0, Flags=........ |
| 23457 | 38.53431800 | WibhuTec_41:71:f3 | Broadcast | 802.11 | 419 | Beacon frame, SN=3072, FN=0, Flags=........, BI=100, SSID=dav_open |
| 23471 | 38.55411500 | WibhuTec_41:71:ff | Htc_14:8b:9b | 802.11 | 174 | Disassociate, SN=3753, FN=0, Flags=........ |
| 23482 | 38.57615600 | WibhuTec_41:71:ff | Broadcast | 802.11 | 174 | Disassociate, SN=3758, FN=0, Flags=........ |
| 23488 | 38.58260300 | WibhuTec_41:71:ff | Htc_14:8b:9b | 802.11 | 174 | Disassociate, SN=3760, FN=0, Flags=........ |
| 23527 | 38.67215800 | Pathscal_00:11:80 | Broadcast | 802.11 | 337 | Beacon frame, SN=1422, FN=0, Flags=........, BI=100, SSID=MK_11N1 |
| 23531 | 38.67931600 | WibhuTec_41:71:ff | Htc_14:8b:9b | 802.11 | 174 | Disassociate, SN=3781, FN=0, Flags=........ |
| 23541 | 38.71328300 | WibhuTec_90:6e:f0 | Broadcast | 802.11 | 302 | Beacon frame, SN=1775, FN=0, Flags=........, BI=100, SSID=Piy2G-SSID |
| 23548 | 38.72983600 | WibhuTec_41:71:ff | Broadcast | 802.11 | 174 | Disassociate, SN=3790, FN=0, Flags=........ |
| 23556 | 38.74408800 | WibhuTec_d0:38:01 | Broadcast | 802.11 | 302 | Beacon frame, SN=1378, FN=0, Flags=........, BI=100, SSID=LSDK_WPA2_an |
| 23569 | 38.76996800 | WibhuTec_41:71:ff | Broadcast | 802.11 | 174 | Disassociate, SN=3798, FN=0, Flags=........ |
| 23575 | 38.78038500 | Pathscal_d0:09:00 | Broadcast | 802.11 | 393 | Beacon frame, SN=3229, FN=0, Flags=........, BI=100, SSID=PST-c75-1 |
| 23580 | 38.78705400 | WibhuTec_d0:2a:20 | Broadcast | 802.11 | 361 | Beacon frame, SN=3565, FN=0, Flags=........, BI=100, SSID=Spectrum |
| 23583 | 38.78927700 | WibhuTec_41:71:ff | Broadcast | 802.11 | 174 | Disassociate, SN=3802, FN=0, Flags=........ |
| 23584 | 38.78940800 | WibhuTec_41:71:ff | Htc_14:8b:9b | 802.11 | 174 | Disassociate, SN=3803, FN=0, Flags=........ |

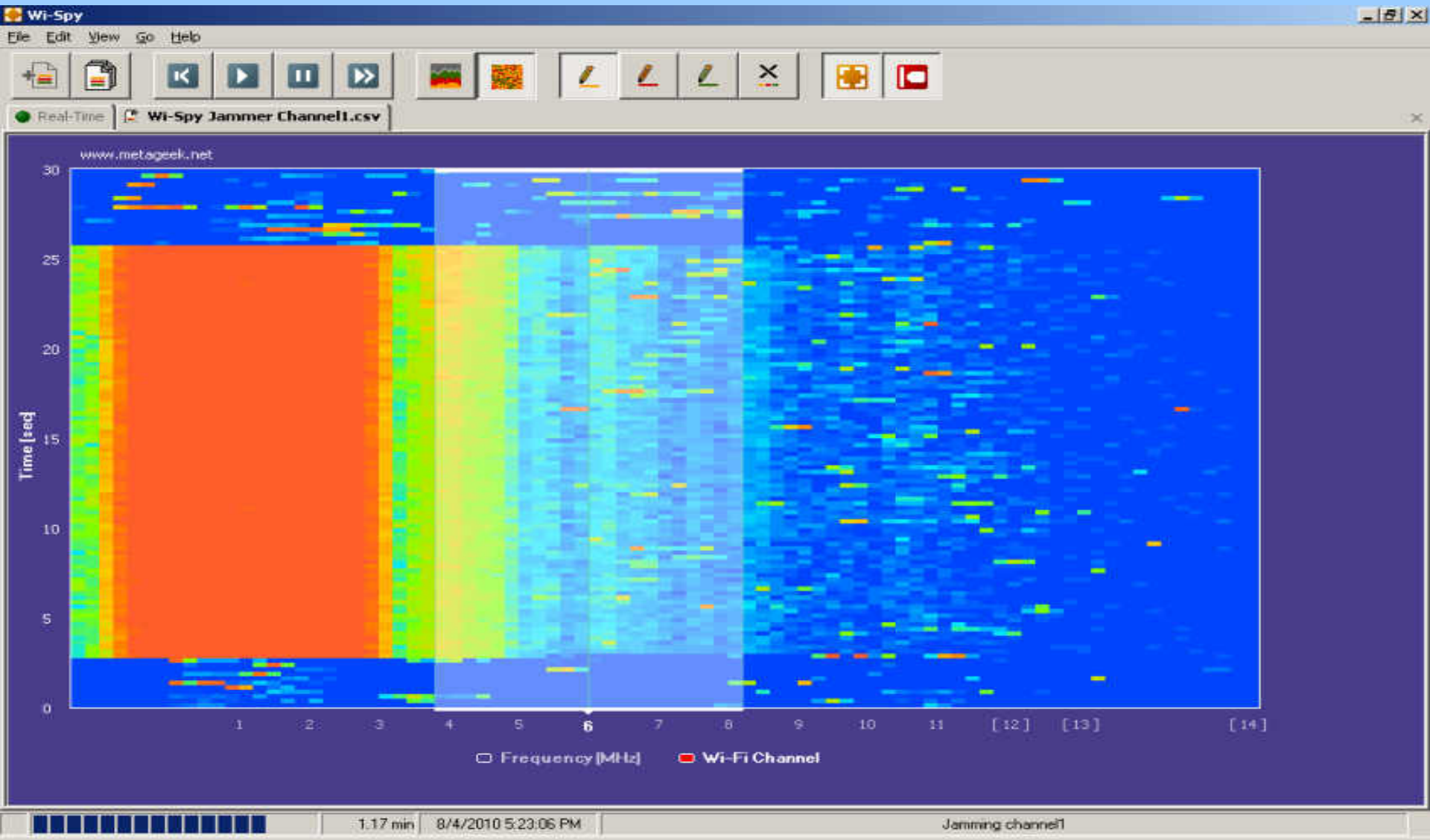
# DoS By RTS Flood

| | | | | | | |
|---|---|---|---|---|---|---|
| 2613 | 5.671256000 | WibhuTec_d0:2a:20 | Htc_14:8b:9b | 802.11 | 345 | Probe Response, SN=2175, FN=0, Flags=........, BI=100, SSID=Spectrum |
| 2620 | 5.685225000 | WibhuTec_d0:2a:21 | Htc_14:8b:9b | 802.11 | 323 | Probe Response, SN=4010, FN=0, Flags=....R..., BI=100, SSID=ATNGuest |
| 2955 | 6.322783000 | WibhuTec_41:71:f0 | Htc_14:8b:9b | 802.11 | 296 | Probe Response, SN=40, FN=0, Flags=....R..., BI=100, SSID=dav_wpa |
| 3005 | 6.417642000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3006 | 6.417676000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3008 | 6.417908000 | Cisco_40:e6:7f | Htc_14:8b:9b | 802.11 | 274 | QoS Data, SN=17, FN=0, Flags=.pm.R.F. |
| 3013 | 6.422589000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3015 | 6.422641000 | Cisco_40:e6:7f | Htc_14:8b:9b | 802.11 | 264 | QoS Data, SN=19, FN=0, Flags=.pm.R.F. |
| 3021 | 6.424925000 | Cisco_40:e6:7f | Htc_14:8b:9b | 802.11 | 264 | QoS Data, SN=21, FN=0, Flags=.p....F. |
| 3084 | 6.530604000 | LiteonTe_01:21:b3 (T | WibhuTec_d0:30:60 (RA | 802.11 | 164 | Request-to-send, Flags=........ |
| 3092 | 6.543894000 | LiteonTe_01:21:b3 (T | WibhuTec_d0:30:60 (RA | 802.11 | 164 | Request-to-send, Flags=........ |
| 3460 | 7.215595000 | SamsungE_4b:c1:f4 (T | WibhuTec_d0:2a:20 (RA | 802.11 | 164 | Request-to-send, Flags=........ |
| 3469 | 7.249669000 | LiteonTe_01:21:b3 (T | WibhuTec_d0:30:60 (RA | 802.11 | 164 | Request-to-send, Flags=........ |
| 3524 | 7.321736000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3526 | 7.326179000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3527 | 7.326661000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3528 | 7.326688000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3529 | 7.327228000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3531 | 7.328814000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3532 | 7.330614000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3533 | 7.330647000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |
| 3534 | 7.330661000 | WibhuTec_41:71:f0 (T | Htc_14:8b:9b (RA) | 802.11 | 164 | Request-to-send, Flags=........ |

# DoS By NAV Duration

| No. | NAV Duration | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|---|
| 761 | 0 | 1.898020000 | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 975 | 0 | 2.383255000 | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 1909 | 0 | 4.402240000 | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 2322 | 0 | 5.382442000 | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 2750 | 0 | 6.389207000 | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 2989 | 0 | 6.884010000 | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 4364 | 0 | 9.389662000 | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5111 | 0 | 10.88282300( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5570 | 0 | 11.88507300( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5770 | 1742 | 12.28142600( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5771 | 1742 | 12.28611100( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5772 | 1742 | 12.28678800( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5774 | 1742 | 12.28789900( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5776 | 1742 | 12.29325100( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5777 | 1742 | 12.29380700( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5778 | 1742 | 12.29439800( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5779 | 1742 | 12.29505500( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5787 | 1782 | 12.30766000( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5788 | 1782 | 12.30906100( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5790 | 1782 | 12.30986400( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5794 | 1782 | 12.31195400( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5797 | 1782 | 12.31315600( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5799 | 1782 | 12.31379400( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |
| 5817 | 1782 | 12.33903400( | WibhuTec_90:03:50 | Htc_14:8b:9b (RA) | 802.11 | 38 | Request-to-send, Flags=........C |

# RF Jamming


Wi-Spy
File Edit View Go Help
Real-Time
Wi-Spy Jammer Channel1.csv
www.metageek.net
Time [sec]
30
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 [12] [13] [14]
Frequency [MHz]
Wi-Fi Channel
1:17 min
8/4/2010 5:23:06 PM
Jamming channel1

# Wi-Fi Threats: A Quick View From the Trenches

# Statistics From Real-Life Deployments
## May-Jun 2014 (Data for 30 days)

| Number of Sites / Threat Instance | Rogue AP | Client Mis-associations | Mobile Hotspots/ Virtual APs | DoS Attacks |
|---|---|---|---|---|
| Customer 1 (258) | 84 | 4963 | 35 | 1 |
| Customer 2 (188) | 4 | 97 | 6 | 33 |
| Customer 3 (507) | 196 | 446 | 48 | 21 |

# Threat Mitigation

# Unfortunately, none of these strategies work!

Let's ban Wi-Fi

We don't have "that" problem because…

# Use Strong Encryption and Authentication For Your Authorized WLAN (WPA2)!

**But, this does not protect against threats due to unmanaged devices!**

# Packet Sniffers & Pen Testing Tools

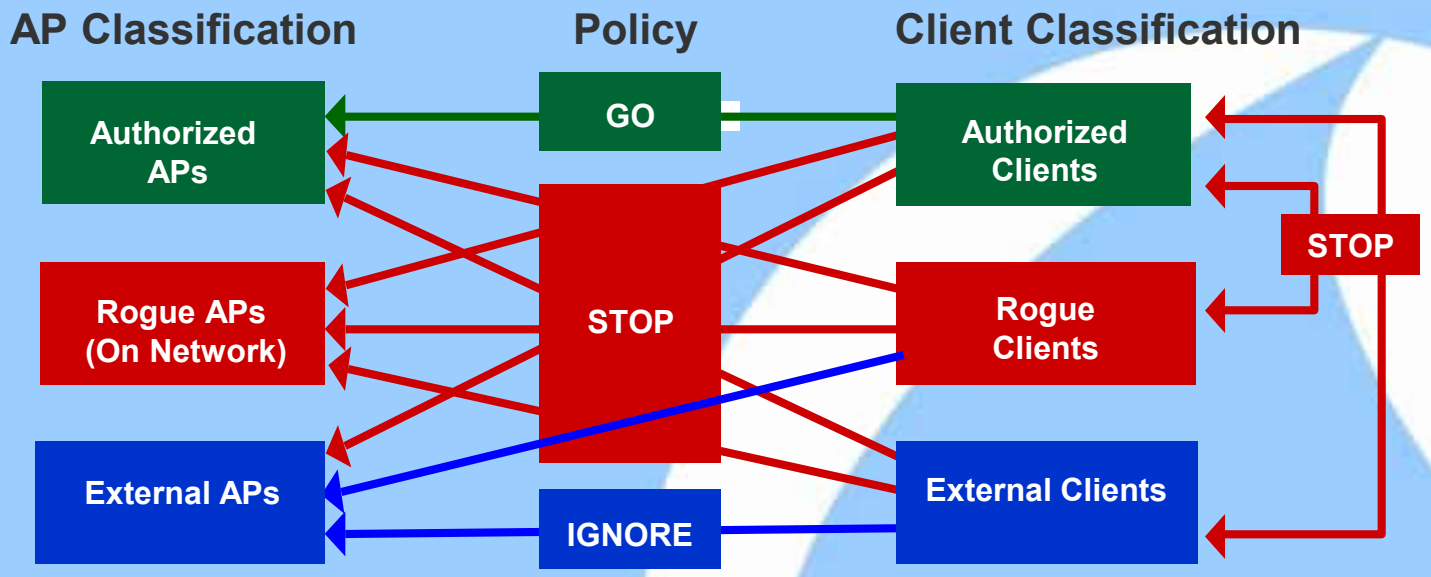
# Several Free and Commercial Sniffers available

- Wireshark
- Airpcap
- Backtrack
- KARMA
- Metasploit
- AirCrack-ng

# Wireless IDS (WIDS)

# WIDS: Sniff and Detect Threats

# Threat Mitigation: The Essence

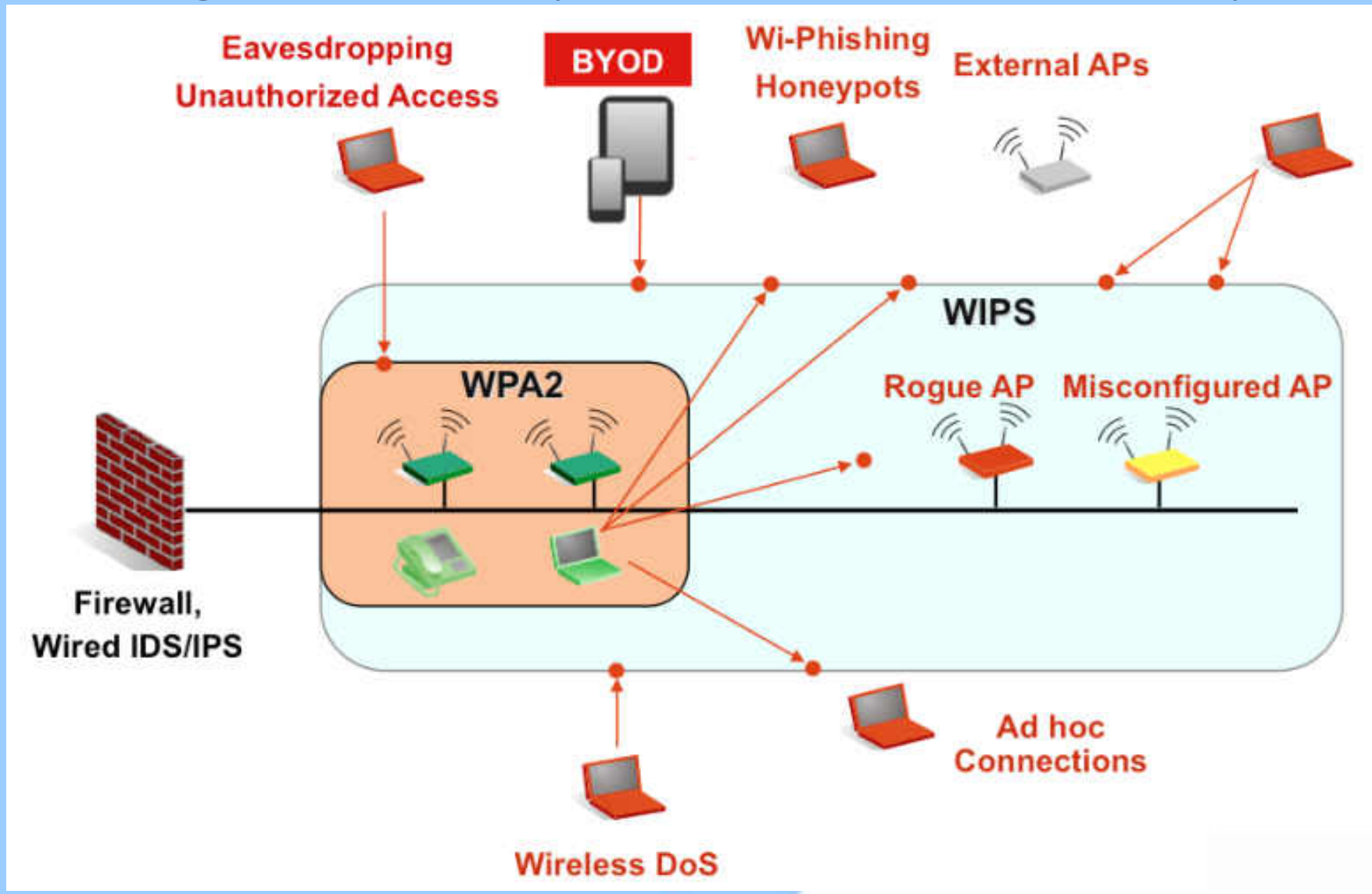



**AUTOMATICALLY DETECT AND BLOCKS RED PATHS!**

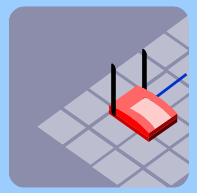
# Wireless IPS (WIPS)

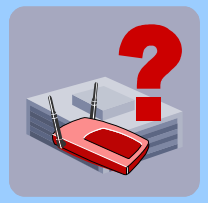
# WIPS – 24x7 Visibility & Protection

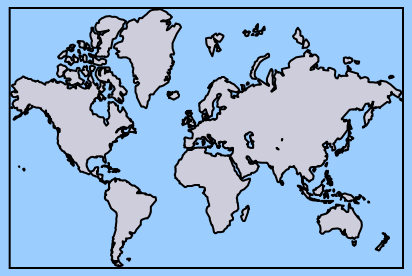
## Adding another layer to Network Security



# Capabilities of a WIPS



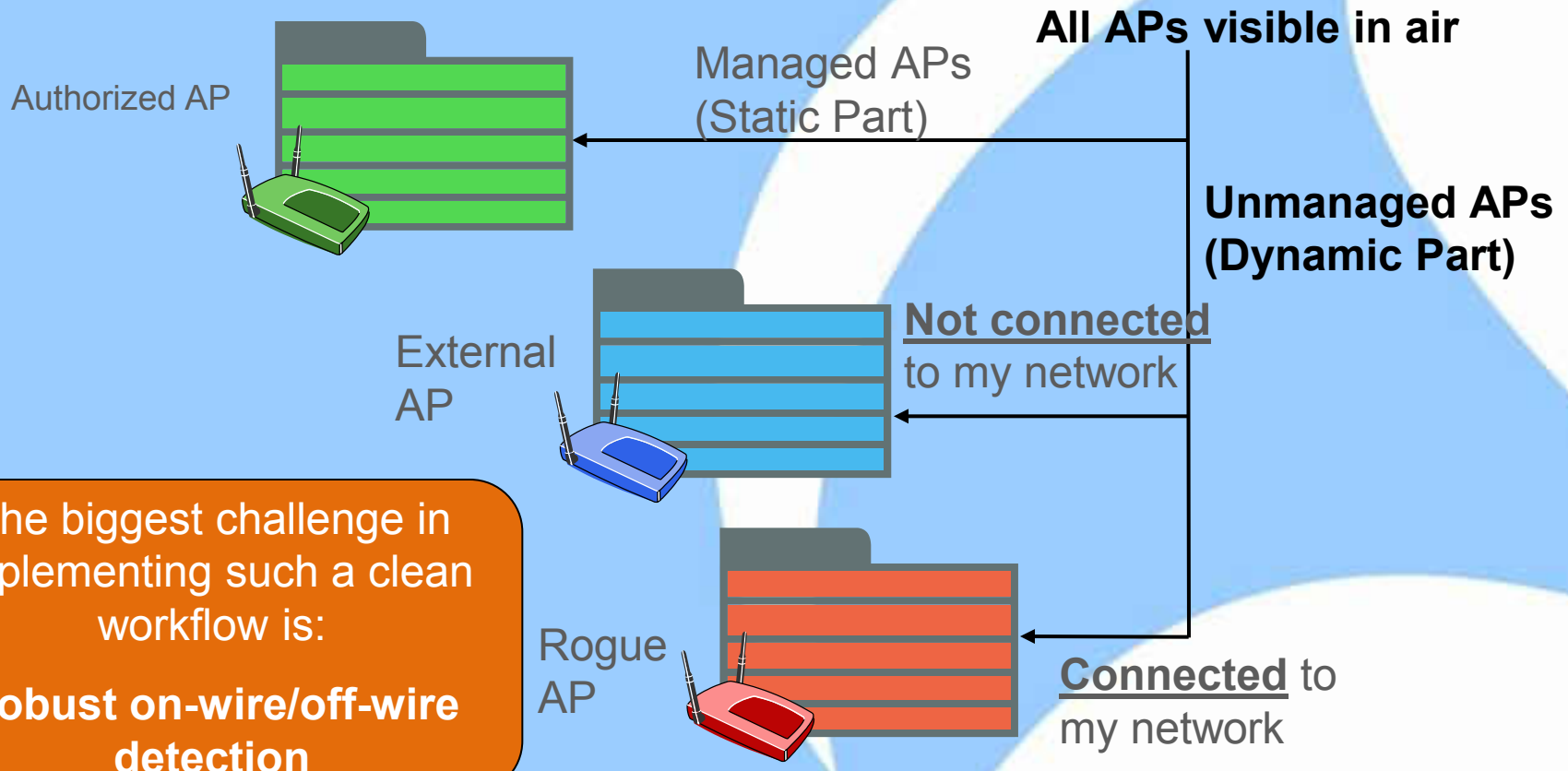




- Report wireless vulnerabilities proactively and detect all types of threats in real-time
- Classify what is a real threat and if it is on your network
- Automatically block unauthorized wireless activity
- Physically locate and remove threats
- Enforce security policies at multiple distributed sites without leaving your desk

# Rogue AP Detection

- Automatically classifying APs visible in airspace into three categories: Authorized, External and Rogue




The biggest challenge in implementing such a clean workflow is:

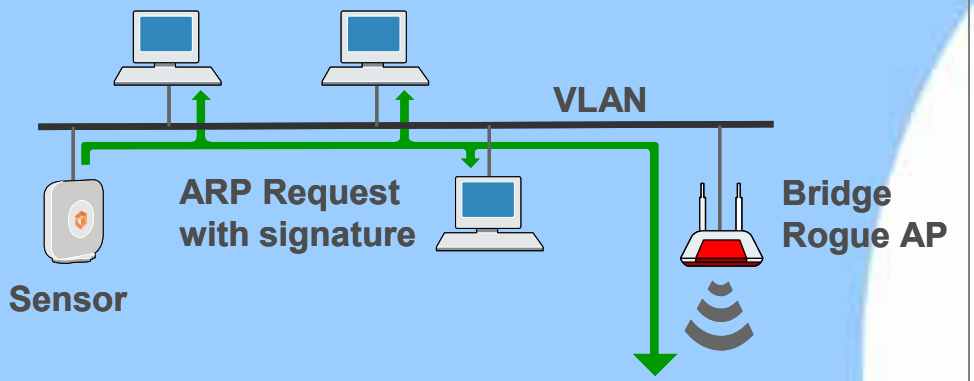
**Robust on-wire/off-wire detection**

# Key Enabler For Connectivity

**Definitive "on-wire / off-wire" test**

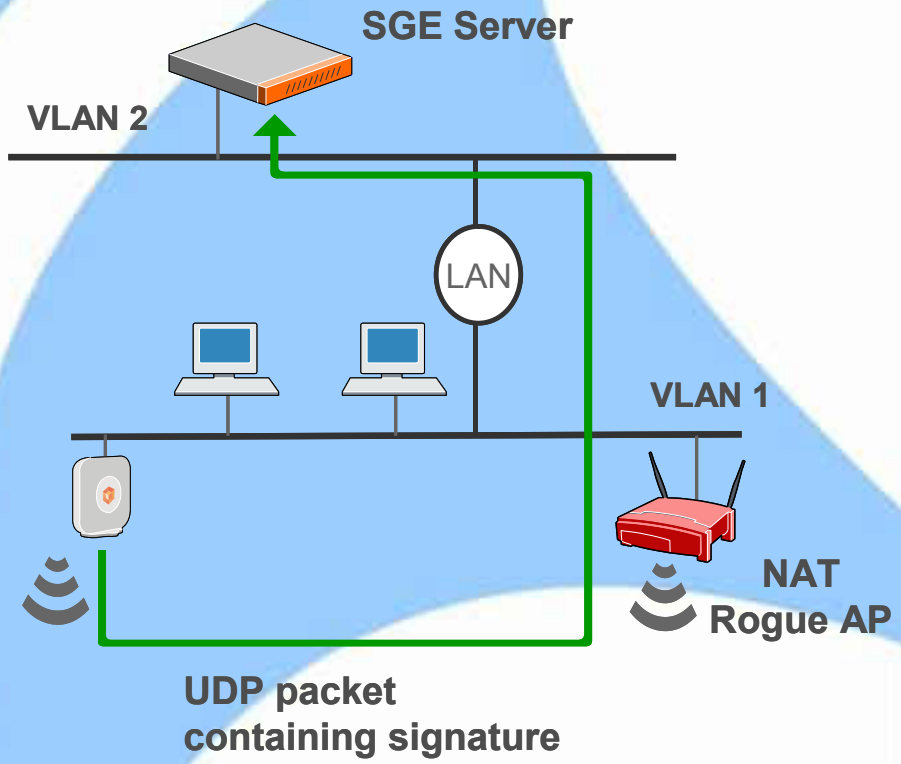
### ARP Request Marker Packet

Sensor sends ARP requests with signatures on the wire and detects if any get forwarded onto the wireless side




### UDP Reverse Marker Packet

Sensor sends UDP packets with signatures in the air and server detects if any get forwarded onto the wire

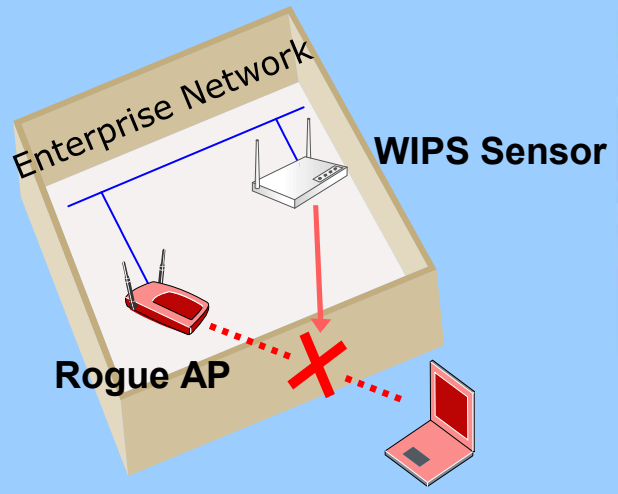


# Can wire side only scanning protect from all Rogue AP

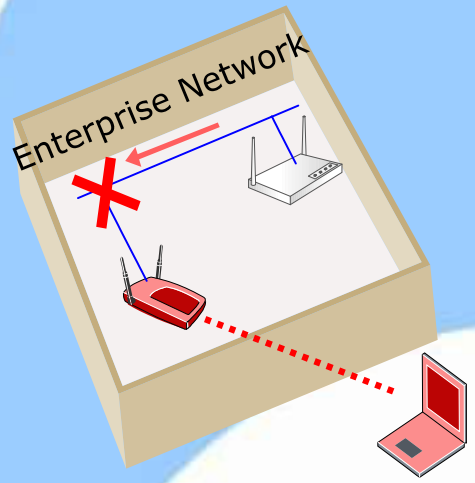
- No!
- Several Rogue AP types are undetectable by wire side only scanning, examples:
  - Bridging APs on a subnet inconsistent with their wired IP address (default configuration)
  - Soft APs
  - Router (NAT) APs with cloned wire side MAC address
- See http://blog.airtightnetworks.com/rogue-ap-detection-pci-compliance/ for more details

# How does WIPS block Rogue AP

- Over the air quarantine
  - WIPS sensor blocks client's connection to Rogue AP by transmitting spoofed disconnection frames
  - Deauthentication is popularly used disconnection frame

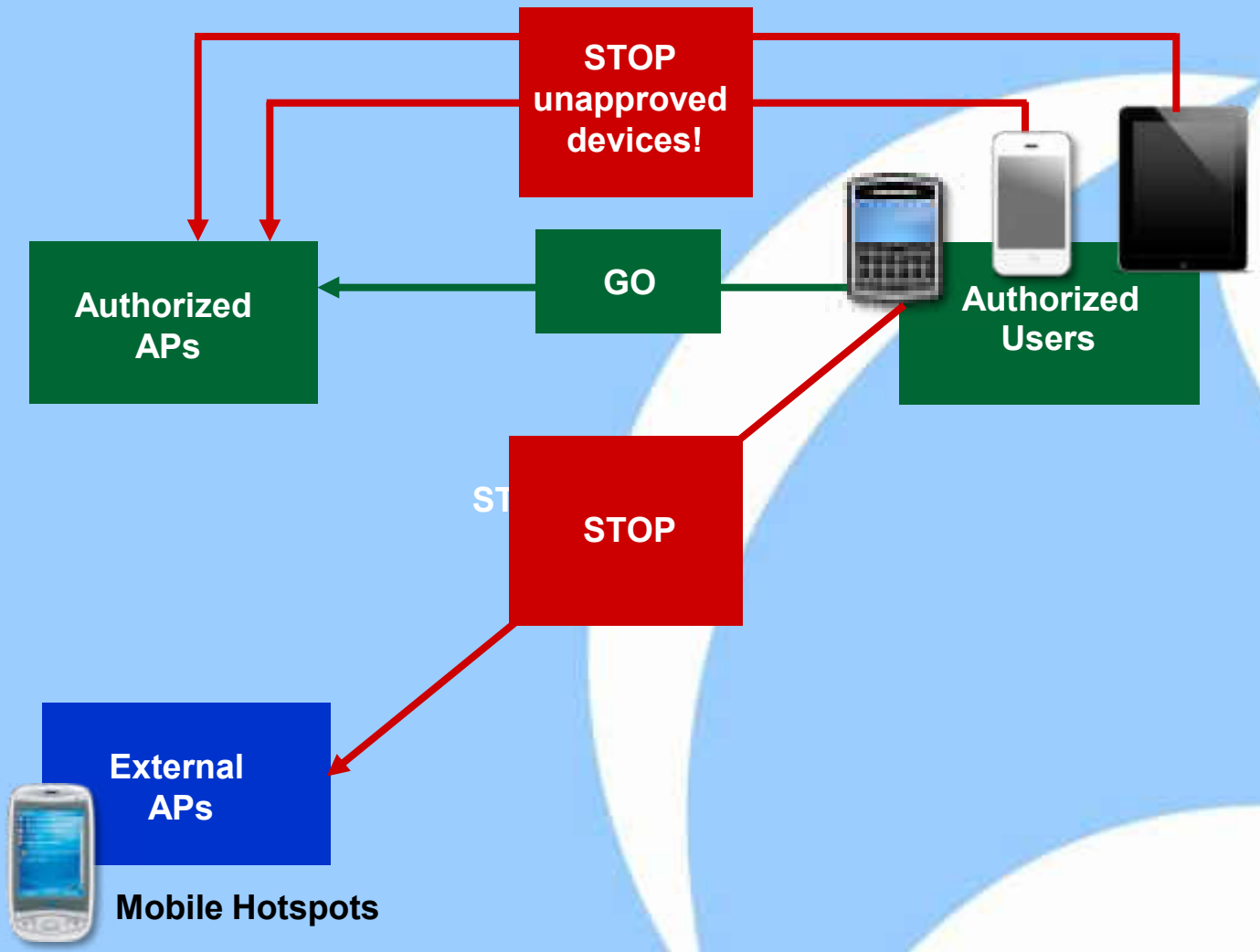



- Switch port disable
  - WIPS attempts to locate switch port into which Rogue AP is connected
  - If found, disables the switch port using SNMP



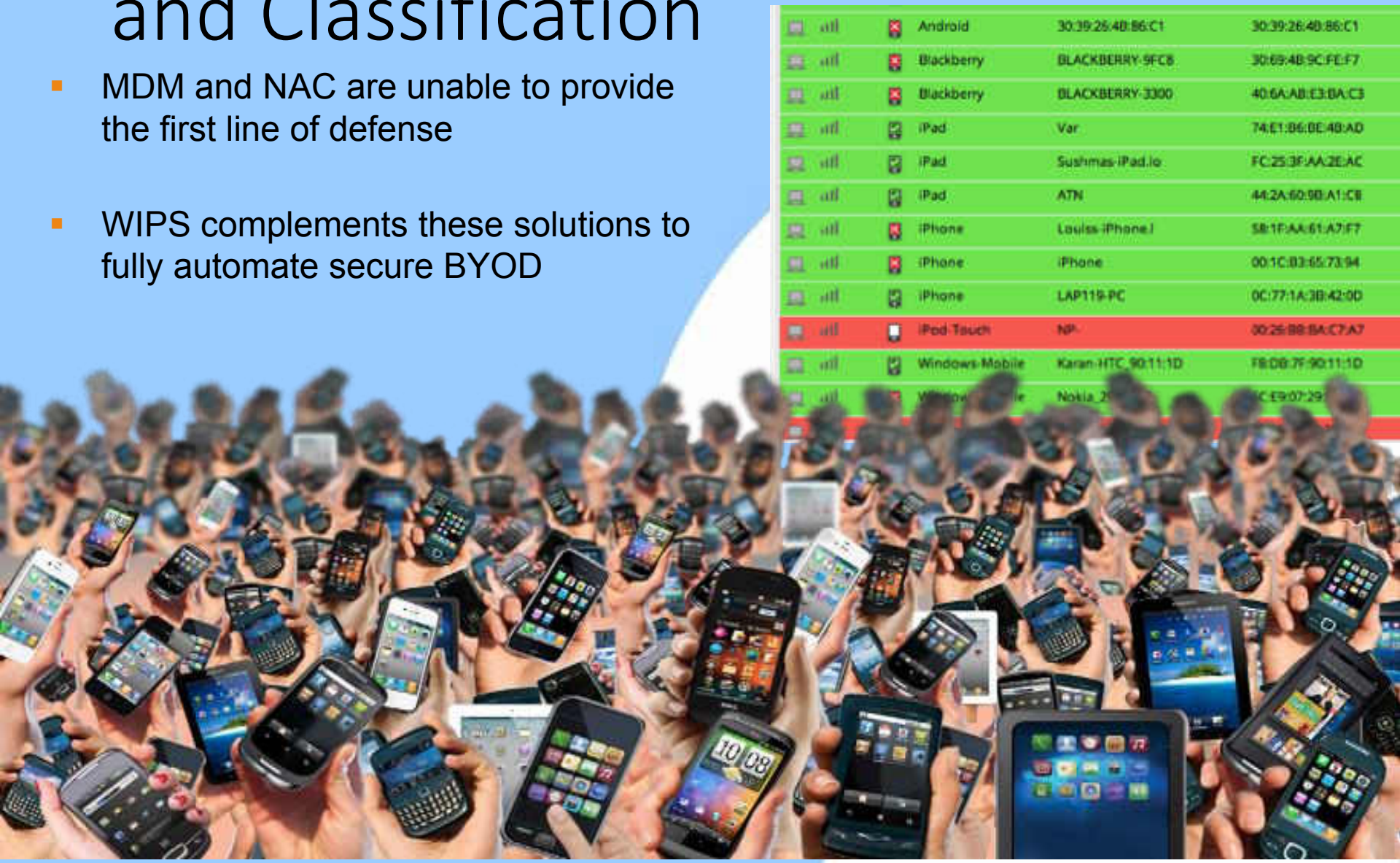
# BYOD Mitigation

# Extending the WIPS for BYOD Policy Enforcement



# Automatic Device Fingerprinting and Classification

- MDM and NAC are unable to provide the first line of defense
- WIPS complements these solutions to fully automate secure BYOD



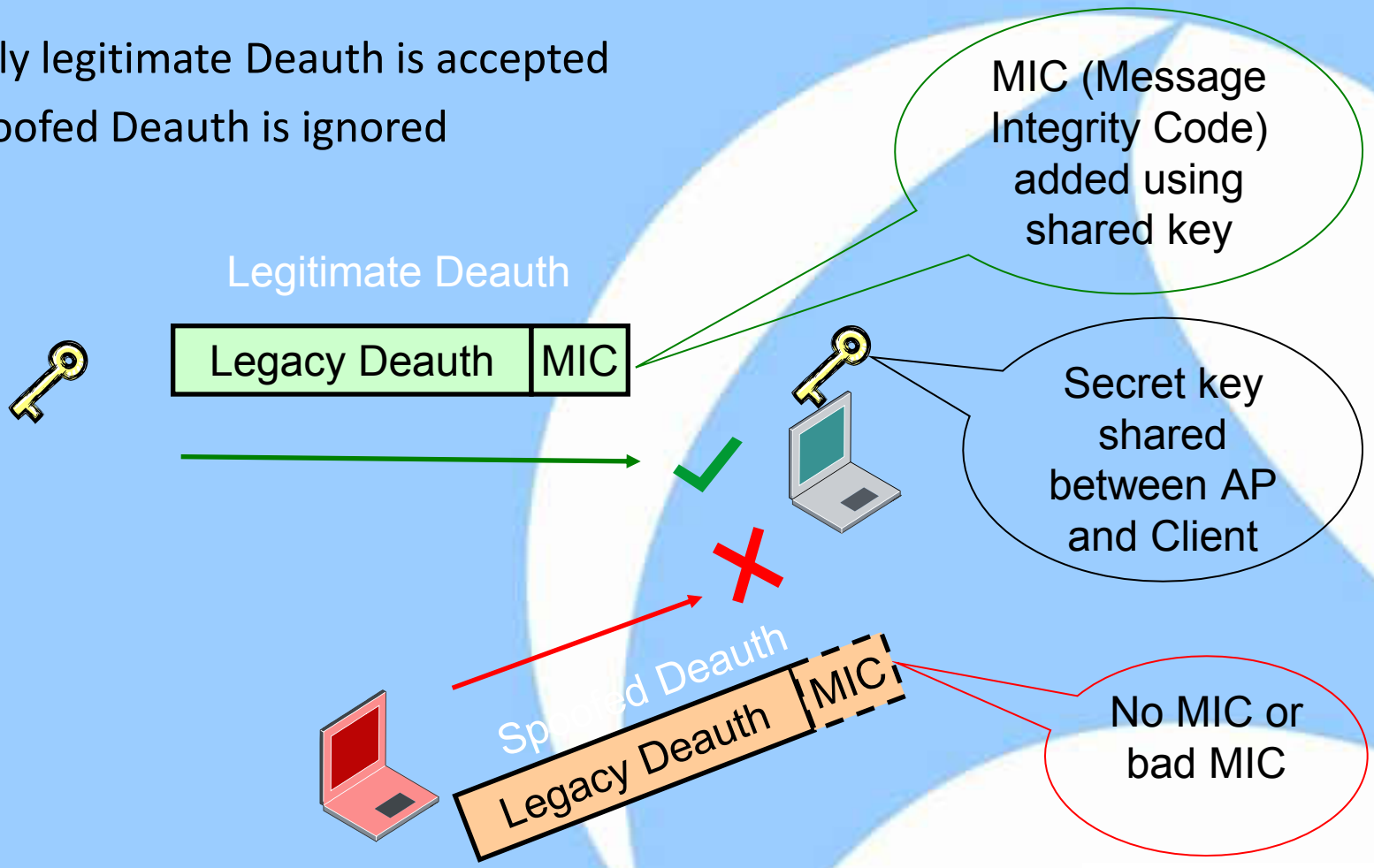
# DoS Attack Mitigation

# 802.11w: Basic Idea

Can we introduce some notion of authentication/integrity in management frames so that a receiver can differentiate legitimate packets from that of an attacker?

# 802.11w based Deauthentication Attack Prevention

- Only legitimate Deauth is accepted
- Spoofed Deauth is ignored

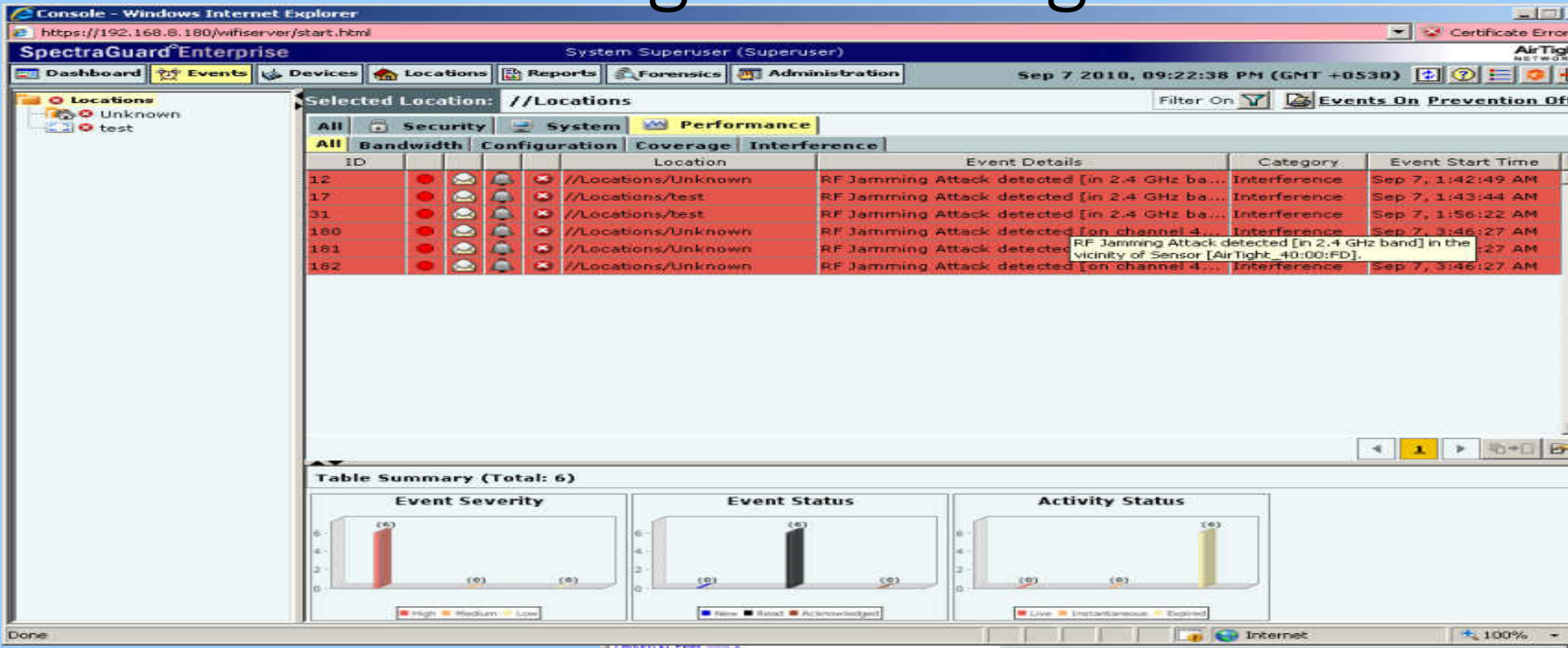


# What does IEEE 802.11w achieve?

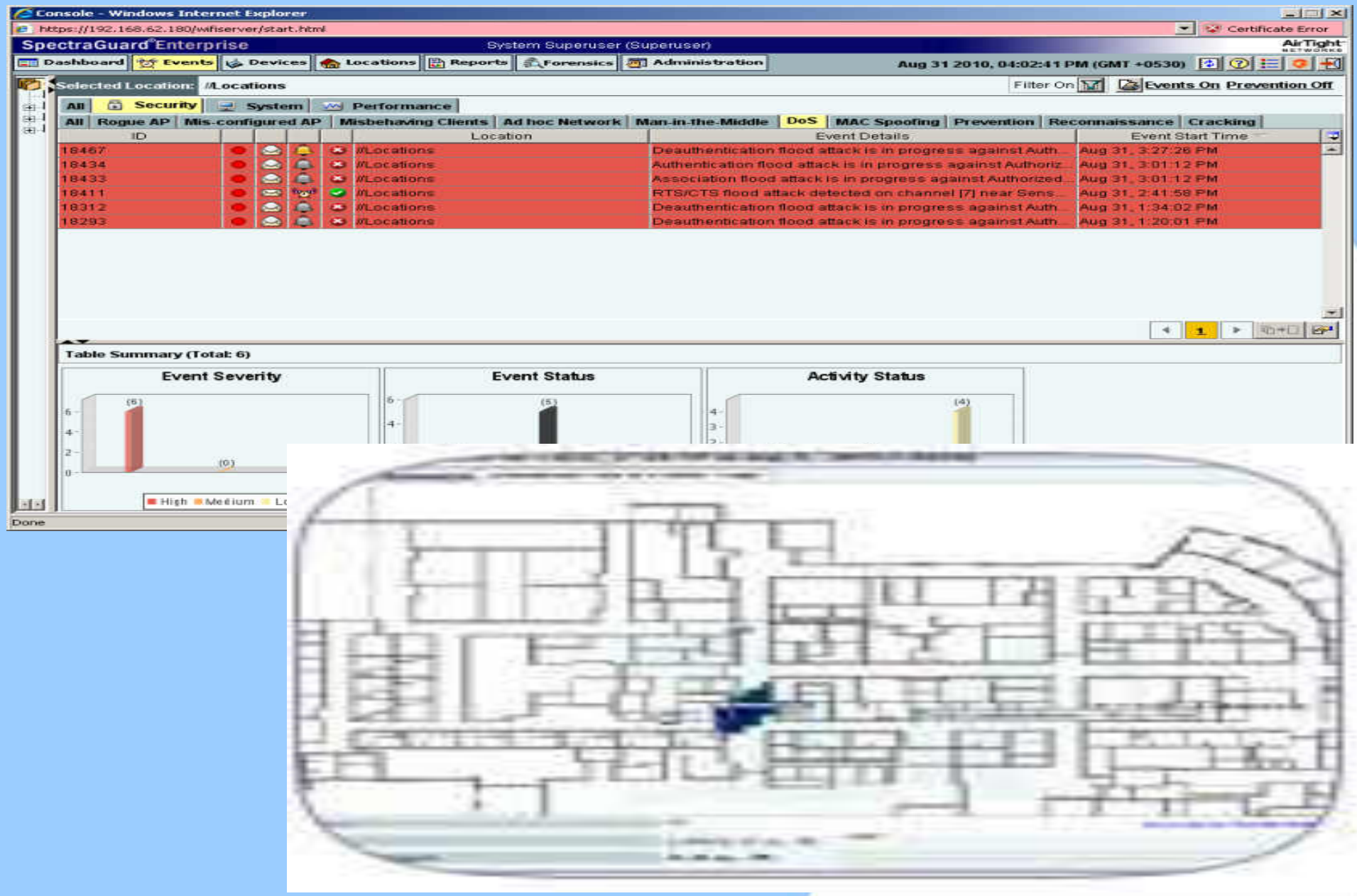
- 802.11w gets rid of certain types of DoS Attacks only
  - "Spoofed Disconnect" DoS attacks resulting from spoofing of
    - (i) Deauthentication (Deauth), (ii) Disassociation (Disassoc), (iii) Association (Assoc) Request in existing connection, or (iv) Authentication (Auth) Request in existing connection
- Certain "Action Management Frames" are also made anti-spoofing
  - Spectrum Management, QoS, BlockAck, Radio Measurement, Fast BSS Transition
- But, other DoS attacks are still possible!

WIPS Complements 802.11w by providing a detection & location based DoS mitigation workflow!

# RF Jamming DOS Mitigation




# MAC Level DoS Attacks



# Summary: Five steps to protect against WiFi security breaches

| Recommended Best Practice | WiFi deployed | WiFi not deployed |
|---|---|---|
| **Use strong authentication and encryption**: Use the best standards for authentication and encryption (e.g., WPA/WPA2) when deploying WiFi networks | ✓ | |
| **Monitor guest WiFi access**: Authenticate guest users and monitor unauthorized access when providing guest access over WiFi networks | ✓ | |
| **Conduct wireless security audits and scans**: Periodically conduct wireless scans to detect presence of unauthorized WiFi devices and activity in your premises. | ✓ | ✓ |
| **Follow endpoint wireless security best practices**: Promote WiFi security best practices among laptop users. Using wireless security endpoint security agent, enforce your enterprise policies seamlessly across all laptops and secure them even when they are away. | ✓ | ✓ |
| **Use a Wireless Intrusion Prevention System (WIPS):** Prevent leakage of sensitive data and protect your network from wireless security threats with 24/7 wireless monitoring | ✓ | ✓ |

# Limitations of Solutions Discussed So Far …

- No one can protect a mis-configured network – e.g., WEP or Open Wi-Fi Network ☺
- Educate your users – otherwise, technology solutions can just go only so much!

# ACKNOWLEDGEMENTS


- Many Thanks To
  - Sharkfest organizing committee
  - Rohan Shah, AirTight Networks
  - Davneet Singh, AirTight Networks
  - Ranganath Jilla, AirTight Networks

Thank You

Questions?

gopi@airtightnetworks.com