Wi-Fi Threats and Countermeasures

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

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

1. Discovery
   - Client discovers AP, requests connection.

2. Authentication
   - AP asks Client to prove its identity.

3. Association
   - Client binds its identity to AP.

4. With WPA/WPA2
   - Higher Level Authentication

5. (Encrypted) Data
   - Start communication.
Stages of establishing a WEP-encrypted WiFi connection

- **Step 1**: AP Discovery (SSID, signal strength)
- **Step 2**: Open (No) Authentication
- **Step 3**: Association
- **Step 4**: WEP Encrypted Data Communication
WEP is broken. Let’s move on!
Stages in establishing a WPA-encrypted WiFi connection

Step 1: AP Discovery (SSID, signal strength)

Step 2: Open (No) Authentication

Step 3: Association

Step 4.1: 802.1x (EAP) Authentication

Step 4.2: WEP Shared Key Authentication

Step 5: WEP Like Encrypted Data Communication

Addition of TKIP
Session specific
802.1x or PSK
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

Online WPA cracker with stats - besside-ng companion

Upload your WPA handshake here and your network will be cracked for you automatically. Contribute to WPA security research - the more handshakes you upload, the more stats, and the more we'll understand how feasible WPA cracking is in practice (currently 3% are crackable based on 49877 networks).

Upload WPA handshake capture

To obtain the WPA handshake, use besside-ng (from aircrack-ng's SVN), a tool that will automatically own all the WPA networks it finds. If you have Internet connectivity while running besside-ng, use the -s wpa.darkircop.org option to upload wpa.cap automatically.

WPA cracking in practice (live stats)

Based on 49877 networks and a 46M word dictionary:

- What's the success rate when cracking WPA? 5% (2624/49877).
  WPA cracking works by trying words from a dictionary until the password is found. So the question is equivalent to "how many people use dictionary words - like hello, world - as their WPA password?"
- Is a large dictionary necessary? You'll crack 52% more networks from the crackable ones.
  A large dictionary has more chances of containing the network's password. But, it may be that people either choose very simple passwords (so a small dictionary will suffice) or a very complicated password (practically uncrackable) giving large dictionaries diminishing returns.
- Do rainbow tables help? 2% of the crackable networks will be cracked faster.
  Rainbow tables speed up WPA cracking, but only when cracking networks who's name is present in a predefined list of 1000 SSIDs. And, the passphrase still needs to be in the dictionary.
If using **WPA/WPA2 - PSK**

Use a password with **at least eight** characters long and mix of **alphanumeric and special characters**
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:
  • Tk iptun-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

1. **Step 1**: AP Discovery (SSID, signal strength)
2. **Step 2**: Open (No) Authentication
3. **Step 3**: Association
4. **Step 4.1**: 802.1x (EAP) Authentication
   - Pre-shared Keys (PSK)
5. **Step 5**: CCMP Encrypted Data Communication

- CCMP (Change in h/w encryption engine)
- Session specific
- 802.1x or PSK
Open Authentication

Association

EAP Identity Request

EAP Identity Response

Open Controlled Port allowing only EAP messages to pass through.

Opening Method Handshake

Identity Proof and Master Key Generation

Accept/Provide Master Key

Generate Transient Keys

Generate Master Key

EAP Success

EAPOL 4-Way Handshake

Encrypted Data Exchange

EAPOL Logoff

Open Uncontrolled Port allowing data to pass through.
Open Authentication, Association, EAP Identity Request

Phase 1: Est. TLS tunnel, auth server
- EAP Identity Request
- EAP Identity Response (anonymous@realm)
- TLS Client Hello (Rand1)
- TLS Server Hello (Rand2, server public certificate)
- TLS Client Key Exchange (Encryption key Encrypted with public certificate)

Phase 2: MSCHAPv2 in TLS tunnel, auth Client
- EAP Identity Request
- EAP Identity Response (userid@realm)
- Server Challenge
- Response to Server Challenge / Client Challenge
- Success / Response to Client Challenge /
- Success
- EAP Success
- Accept/Provide Master Key
- EAPOL 4-Way Handshake

Wireless Link
Access Point
Wired LAN
Authentication Server
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

<table>
<thead>
<tr>
<th>Time</th>
<th>MAC Address</th>
<th>BSSID</th>
<th>Interface</th>
<th>Channel</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:02:11</td>
<td>WibhuTec 90:03:50</td>
<td>802.11</td>
<td></td>
<td></td>
<td>Authentication, SN=2319, FN=0, Flags=...R...</td>
</tr>
<tr>
<td>8:02:11</td>
<td>Htc_14:8b:9b</td>
<td>802.11</td>
<td></td>
<td></td>
<td>Authentication, SN=256, FN=0, Flags=...R...</td>
</tr>
<tr>
<td>8:02:11</td>
<td>WibhuTec 90:03:50</td>
<td>802.11</td>
<td></td>
<td></td>
<td>Authentication, SN=2320, FN=0, Flags=...R...</td>
</tr>
<tr>
<td>8:02:11</td>
<td>Htc_14:8b:9b</td>
<td>802.11</td>
<td></td>
<td></td>
<td>Key (Message 1 of 4)</td>
</tr>
<tr>
<td>8:02:11</td>
<td>WibhuTec 90:03:50</td>
<td>802.11</td>
<td></td>
<td></td>
<td>Key (Message 2 of 4)</td>
</tr>
<tr>
<td>8:02:11</td>
<td>Htc_14:8b:9b</td>
<td>802.11</td>
<td></td>
<td></td>
<td>Key (Message 3 of 4)</td>
</tr>
<tr>
<td>8:02:11</td>
<td>WibhuTec 90:03:50</td>
<td>802.11</td>
<td></td>
<td></td>
<td>Key (Message 4 of 4)</td>
</tr>
</tbody>
</table>

**WPA2**

**Open**

---

Sharkfest 2014
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!
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

<table>
<thead>
<tr>
<th>Time</th>
<th>MAC Address</th>
<th>BSSID</th>
<th>Channel</th>
<th>Type</th>
<th>SSID</th>
<th>Flags</th>
<th>BI</th>
<th>SSID Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11</td>
<td>C0:16:3C:23:45:56</td>
<td>Broadcast</td>
<td>802.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>802.11</td>
<td>00:11:22:33:44:55</td>
<td>Broadcast</td>
<td>802.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
- Beacon frames indicate network availability.
- Disassociation frames indicate network disconnection.

Sharkfest 2014
<table>
<thead>
<tr>
<th>Frame ID</th>
<th>Channel</th>
<th>BSSID</th>
<th>SSID</th>
<th>BSSID</th>
<th>SSID</th>
</tr>
</thead>
<tbody>
<tr>
<td>2613</td>
<td>802.11</td>
<td>WibhuTec_d0:2a:20</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>2520</td>
<td>802.11</td>
<td>WibhuTec_d0:2a:21</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>2955</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3005</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3006</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3008</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3009</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3010</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3011</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3021</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3022</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3023</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3024</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3025</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3026</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3027</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3028</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3029</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
<tr>
<td>3030</td>
<td>802.11</td>
<td>WibhuTec_41:71:f6</td>
<td></td>
<td>Htc_14:8b:9b</td>
<td></td>
</tr>
</tbody>
</table>

Sharkfest 2014
### DoS By NAV Duration

<table>
<thead>
<tr>
<th>No.</th>
<th>NAV Duration</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>761</td>
<td></td>
<td>0 1.898026000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>975</td>
<td></td>
<td>0 2.383255000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>1969</td>
<td></td>
<td>0 4.402246000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>2322</td>
<td></td>
<td>0 5.382442000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>2756</td>
<td></td>
<td>0 6.389207000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>2989</td>
<td></td>
<td>0 6.884016000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>4364</td>
<td></td>
<td>0 9.389662000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5111</td>
<td></td>
<td>0 10.882823000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5576</td>
<td></td>
<td>0 11.885073000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5776</td>
<td></td>
<td>1742 12.281426000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5771</td>
<td></td>
<td>1742 12.286111000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5772</td>
<td></td>
<td>1742 12.286788000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5774</td>
<td></td>
<td>1742 12.287899000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5776</td>
<td></td>
<td>1742 12.293251600</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5777</td>
<td></td>
<td>1742 12.293807000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5778</td>
<td></td>
<td>1742 12.294398800</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5779</td>
<td></td>
<td>1742 12.295055000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5787</td>
<td></td>
<td>1782 12.309864000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5788</td>
<td></td>
<td>1782 12.309961000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5798</td>
<td></td>
<td>1782 12.309984800</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5794</td>
<td></td>
<td>1782 12.311954000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
<tr>
<td>5817</td>
<td></td>
<td>1782 12.339934000</td>
<td>WibhuTec_90:03:50</td>
<td>Htc_14:8b:9b (RA)</td>
<td>802.11</td>
<td>38</td>
<td>Request-to-send, Flags=..................C</td>
</tr>
</tbody>
</table>
RF Jamming

www.metageek.net
Wi-Fi Threats: A Quick View From the Trenches
## Statistics From Real-Life Deployments

May-Jun 2014 (Data for 30 days)

<table>
<thead>
<tr>
<th>Number of Sites</th>
<th>Rogue AP</th>
<th>Client Mis-associations</th>
<th>Mobile Hotspots/Virtual APs</th>
<th>DoS Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>84</td>
<td>4963</td>
<td>35</td>
<td>1</td>
</tr>
<tr>
<td>(258)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer 2</td>
<td>4</td>
<td>97</td>
<td>6</td>
<td>33</td>
</tr>
<tr>
<td>(188)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer 3</td>
<td>196</td>
<td>446</td>
<td>48</td>
<td>21</td>
</tr>
<tr>
<td>(507)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

What is Kismet?

Kismet is an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system. Kismet will work with any wireless card which supports raw monitoring (rfmon) mode, and (with appropriate hardware) can sniff 802.11b, 802.11a, 802.11g, and 802.11n traffic. Kismet also supports plugins which allow sniffing other media such as DECT.

Kismet identifies networks by passively collecting packets and detecting standard named networks, detecting (and giving time, dectloking) hidden networks, and inferring the presence of nonbeaconing networks via data traffic.

News

Wed Sep 25 2013 - Released the first version of Smart Air: Wi-Fi Manager for Android - Automatically learn where you use Wi-Fi and keep the radio disabled when you aren't near a known spot.

Mon Apr 08 2013 - Kismet-2013-05-11b released. Somehow the latest configure script didn't get into the R1 release so it blew up on lttnc11 detection: No code changes, no package changes.

Wed Mar 27 2013 - Kismet-2013-03-11 released! While this does not have major new features (phy module is still in development) it includes a long list of bug fixes, including better names color support, radiotap fixes, better interface control when setting channels, memory leak fixes, better packaging of distro, and various other quirks. At the usual download page

Thu Dec 06 2012 - Busy busy busy. Two new Android utilities now up - rfmon with a USB NIC, capturing to pcap, no root required, in Android PCAP Capture. To go with that, an easy way to upload pcap files from Android to CloudShark, a web-based implementation of WireShark: CloudShark Uploader
Threat Mitigation: The Essence

AP Classification
- Authorized APs
- Rogue APs (On Network)
- External APs

Policy
- GO
- STOP
- IGNORE

Client Classification
- Authorized Clients
- Rogue Clients
- External Clients

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

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.

Definitive “on-wire / off-wire” test
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

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

STOP unapproved devices!

Authorized APs → Authorized Users

GO

STOP

External APs → Mobile Hotspots
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

Legitimate Deauth

Legacy Deauth [MIC]

Secret key shared between AP and Client

MIC (Message Integrity Code) added using shared key

No MIC or bad MIC
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

<table>
<thead>
<tr>
<th>Recommended Best Practice</th>
<th>WiFi deployed</th>
<th>WiFi not deployed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use strong authentication and encryption</strong>: Use the best standards for authentication and encryption (e.g., WPA/WPA2) when deploying WiFi networks</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td><strong>Monitor guest WiFi access</strong>: Authenticate guest users and monitor unauthorized access when providing guest access over WiFi networks</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td><strong>Conduct wireless security audits and scans</strong>: Periodically conduct wireless scans to detect presence of unauthorized WiFi devices and activity in your premises.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Follow endpoint wireless security best practices</strong>: 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.</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Use a Wireless Intrusion Prevention System (WIPS)</strong>: Prevent leakage of sensitive data and protect your network from wireless security threats with 24/7 wireless monitoring</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
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