Wireless LAN Security: Hacking Techniques and Protection
WLAN Hacking Techniques

Topics to discuss

- What this session is about
- Attack Taxonomy
- Well Known broken technologies
- Incorrectly used technologies
- DoS Attacks
- XSS
- Webauth highjack
What this session is about?

- Understand security issues from a wireless perspective
- Know how to protect your network
- Know how real attacks look like
- How issues can arise from errors in Implementation
Attack Taxonomy

Some classification helps
**Attack Taxonomy**

- **What is an Attack?**
  - Exploiting a vulnerability

- **What is a Vulnerability?**
  - It is a weaknesses that allows an attacker to reduce a system information assurance (wikipedia)
Attack Taxonomy

- Multiple classifications are possible
  
  Passive Attacks:
  - Eavesdropping
  - Traffic Analysis
  
  Active Attack
  - Denial of Service
  - Impersonation
  - Privilege escalation

- Protocol Based
- Resource starvation
Terminology

Let’s be clear
Terminology

- Wireless Technology
- 802.11 – WPA – WPA2
- Authentication
- PSK - 802.1x(EAP)
- Encryption
- TKIP – WEP – AES-CCMP
Terminology

Supplicant  

 Probe req/resp  
Authentication req/resp  
Association req/resp  

Authenticator  

EAP Authentication

Backend Auth Server  

802.1x over 802.11 data  
Radius [79] over UDP
EAP Protocol Attacks

A zoo with lots of Animals
**EAP protocols Attacks**

- **What is EAP?**
  
  Extensible Authentication Protocol, RFC 3748
  
  It is an authentication framework over data link layer
  
  Used commonly over PPP or 802.1x networks

- **What Protocols are over EAP?** - A lot...
  
  EAP-MD5: defined by IETF, minimal security
  
  LEAP: defined by Cisco, obsolete and vulnerable
  
  EAP-TLS: RFC 5126, PKI based, high security, complex
  
  PEAP: joint effort, very common support, multiple inner methods
  
  EAP-Fast: TLS based, evolution of LEAP/PEAP, intended to be easier to implement

  And many more: EAP-IKE, EAP-AKA, EAP-TTLS, EAP-SIM, etc, etc…
EAP protocols Attacks II

Possible Attacks against EAP

- Dictionary based attacks: capture packet exchange, then do offline analysis. May use cryptographic “shortcuts”

- Impersonation attack: take the role of client (supplicant) or of AP ( authenticator)

- DoS: break valid connections

- MitM (Man in the Middle attack): intercept complete conversation and highjack the session
## EAP protocols Attacks III

### Protocol Comparison Example

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Dictionary Attack</th>
<th>MitM</th>
<th>Impersonation</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAP-MD5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LEAP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EAP-TLS</td>
<td></td>
<td>✓</td>
<td>✓ (configuration)</td>
</tr>
<tr>
<td>EAP-FAST</td>
<td></td>
<td>✓</td>
<td>✓ (configuration)</td>
</tr>
<tr>
<td>PEAP</td>
<td></td>
<td>✓ (implementation)</td>
<td>✓ (configuration)</td>
</tr>
</tbody>
</table>
EAP-TLS, secure but can be incorrectly used

How to use a nice technology incorrectly
EAP-TLS, secure but be used incorrectly…

- EAP-TLS is secure authentication algorithm
  - Allows mutual authentication between supplicant and authenticator
  - It is offered by almost all wireless infrastructure and client vendors
  - It can be hard to implement: PKI handling is difficult to do properly, certificate renewal may be an issue on some cases
  - Extensive support on dual factor systems (token cards, etc)
  - Fairly secure: so how can we break it?
    - if trust checks were not set properly.
EAP-TLS, secure but you can break it…

- EAP-TLS needs two certificates
  1. Server side: it identifies the authenticator, and it allows client to know who it is talking to…
  2. Client side: it provides the identity of the client to be validated by the authenticator (AP/WLC, etc)

- Typical problem, taking shortcuts: instead of buying certificate to third party, or implementing proper certificate trust in the domain, client is configured to “do not validate”
EAP-TLS, secure but you can break it…

ProSet

ADU
EAP-TLS, Attack Recipe

- What you need:
  - Authenticator that can ignore client identity
    - Example: WLC configured to ignore identity, or FreeRadius
  - A victim
  - A mechanism to be a “interesting” peer for the victim
EAP-TLS, Attack Recipe

- **WLC** example, just tell it to ignore everything...

  (Cisco Controller) > show local-auth config

  User credentials database search order:
  - Primary ..................................... Local DB

  Timer:
  - Active timeout .............................. 300

  Configured EAP profiles:
  - Name ........................................ cisco
    - Certificate issuer ....................... vendor
    - Peer verification options:
      - Check against CA certificates ........ Disabled
      - Verify certificate CN identity ........ Disabled
      - Check certificate date validity .......... Disabled
  - EAP-FAST configuration:
    - Local certificate required .............. No
    - Client certificate required ............. No
    - Enabled methods .......................... tls
    - Configured on WLANs ..................... 1
EAP-TLS, Attack Recipe

- Match real SSID

(Cisco Controller) > show wlan 1

WLAN Identifier................................. 1
Profile Name........................................ mycompany
Network Name (SSID).............................. mycompany
Status........................................... Enabled

Interface........................................ management

Authentication................................. Global Servers
  Accounting.................................. Disabled
  Dynamic Interface.......................... Disabled
Local EAP Authentication...................... Enabled (Profile 'badguy')

Security

  802.11 Authentication:......................... Open System
  Static WEP Keys.............................. Disabled
  802.1x.......................................... Disabled
  Wi-Fi Protected Access (WPA/WPA2).......... Enabled
    WPA (SSN IE)............................... Disabled
    WPA2 (RSN IE).............................. Enabled
      TKIP Cipher............................... Disabled
      AES Cipher................................ Enabled
  Auth Key Management
    802.1x....................................... Enabled
EAP-TLS, Attack Success!

- Client is now associated to me...

(Cisco Controller) > show client summary

Number of Clients........................................ 4

<table>
<thead>
<tr>
<th>MAC Address</th>
<th>AP Name</th>
<th>Status</th>
<th>WLAN/Guest-Lan</th>
<th>Auth Protocol</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wired</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:1b:77:42:1c:99</td>
<td>LAP1240-4</td>
<td>Associated</td>
<td>1</td>
<td>Yes</td>
<td>802.11g 8 No</td>
</tr>
<tr>
<td>00:1e:be:25:d6:52</td>
<td>LAP1240-4</td>
<td>Probing</td>
<td>N/A</td>
<td>No</td>
<td>802.11b 8 No</td>
</tr>
<tr>
<td>00:1f:3c:ab:b6:c5</td>
<td>LAP1240-4</td>
<td>Probing</td>
<td>N/A</td>
<td>No</td>
<td>802.11b 8 No</td>
</tr>
<tr>
<td>00:40:96:b4:47:6d</td>
<td>LAP1240-4</td>
<td>Probing</td>
<td>N/A</td>
<td>No</td>
<td>802.11b 8 No</td>
</tr>
</tbody>
</table>
EAP-TLS, can I catch this?

- Rogue detection can see this, and flag as malicious:

(Cisco Controller) >show rogue ap summary

Rogue Location Discovery Protocol................ Disabled
Rogue on wire Auto-Contain.......................... Disabled
Rogue using our SSID Auto-Contain............... Disabled
Valid client on rogue AP Auto-Contain.......... Disabled
Rogue AP timeout.................................... 1200

<table>
<thead>
<tr>
<th>MAC Address</th>
<th>Classification</th>
<th># APs</th>
<th># Clients</th>
<th>Last Heard</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:15:2b:d9:59:80</td>
<td>Unclassified</td>
<td>1</td>
<td>1</td>
<td>Thu Feb 18 14:05:42 2010</td>
</tr>
<tr>
<td>00:1e:be:81:95:20</td>
<td>Unclassified</td>
<td>1</td>
<td>0</td>
<td>Thu Feb 18 13:59:42 2010</td>
</tr>
<tr>
<td>00:21:1c:7a:43:e0</td>
<td>Malicious</td>
<td>1</td>
<td>1</td>
<td>Thu Feb 18 14:02:42 2010</td>
</tr>
<tr>
<td>00:a0:c5:f4:7b:b7</td>
<td>Unclassified</td>
<td>1</td>
<td>0</td>
<td>Thu Feb 18 14:02:42 2010</td>
</tr>
</tbody>
</table>
EAP-TLS, can I catch this?

- Rogue rules can easily detect a fake AP:

(wlc) >show rogue ap malicious summary

Number of APs................................................. 2

<table>
<thead>
<tr>
<th>MAC Address</th>
<th>State</th>
<th># APs</th>
<th># Clients</th>
<th>Last Heard</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:1e:4a:8f:75:70</td>
<td>Alert</td>
<td>2</td>
<td>1</td>
<td>Wed May 5 10:17:46 2010</td>
</tr>
<tr>
<td>00:21:1b:64:62:00</td>
<td>Alert</td>
<td>2</td>
<td>0</td>
<td>Wed May 5 10:17:46 2010</td>
</tr>
</tbody>
</table>
EAP-TLS, can I catch this?

- MFP will generate alerts if BSSID is spoofed

(wlc) >show wps mfp statistics

<table>
<thead>
<tr>
<th>BSSID</th>
<th>Radio</th>
<th>Validator AP</th>
<th>Last Source Addr</th>
<th>Found</th>
<th>Error Type</th>
<th>Count</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:1c:b0:ea:63:02 b/g LAP1242-28</td>
<td>00:1C:B0:EA:63:02 Infra</td>
<td>Missing MFP IE</td>
<td>53</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LEAP, lightweight EAP

Not enough for current security requirements
LEAP

- Introduces accommodations for WEP key rotation.
- Used extensively in wireless, not in wired 802.1x.
- Lightweight – hence the name Lightweight EAP.
- Can be programmed into the DSP of the wireless NIC for very fast, hardware based, authentication.
LEAP

Supplicant: Code= Request
Type= Identity

Authenticator: Type-Data= username

Backend Auth Server: Type-Data= client-challenge

Supplicant: Code= Response
Type= Identity

Authenticator: Type-Data= client-challenge

Supplicant: Code= Request
Type= 17 (LEAP)

Authenticator: Type-Data= client-challenge-resp.

Supplicant: Code= Response
Type= 17 (LEAP)
LEAP

Supplicant

Authenticator

Backend Auth Server

Code= Success

Code= Response
Type= 17 (LEAP)

Code= Response
Type= 17 (LEAP)

Type-Data= server-challenge

Type-Data= server-challenge-resp

Radius-Accept
LEAP - Recipe

- Trivial password recovery

- What you need:
  - Sniffer capture of authentication exchange
  - Either be patient or kick user out to obtain exchange

- Main tools available:
  - Asleap (Linux, Windows port)
  - THC-leapcracker
LEAP – Step 1

- Asleap: Uses dictionary hash calculation
- THC-leapcracker: brute force
- Dictionary is faster, but only as good as the initial dictionary input
- Brute Force: slow, but should catch harder variations
LEAP – Step 1 Hash generation

- Let’s focus on Asleap: hash generation

```
root@javier-laptop:/pentest/wireless/asleap# ./genkeys -r /pentest/passwords/wordlists/darkc0de.lst -f hash - index
Genkeys 2.2 - generates lookup file for asleap. <jwright@hasborg.com>
Usage: genkeys [options]
    -r  Input dictionary file, one word per line
    -f  Output pass+hash filename
    -n  Output index filename
    -h  Last 2 hash bytes to filter with (optional)

root@javier-laptop:/pentest/wireless/asleap# ./genkeys -r /pentest/passwords/wordlists/darkc0de.lst -f hash -n index
Genkeys 2.2 - generates lookup file for asleap. <jwright@hasborg.com>
Generating hashes for passwords (this may take some time) ... Done.
1707658 hashes written in 2.09 seconds: 815143.86 hashes/second
Starting sort (be patient) ... Done.
Completed sort in 19429385 compares.
Creating index file (almost finished) ...Done.
root@javier-laptop:/pentest/wireless/asleap#
```
LEAP – Step 2 Obtaining Exchange
LEAP – Step 2 Obtaining Exchange
LEAP – Step 3 Decoding

Ah, that is easy password!
LEAP – Step 3 Decoding

Ok... something better
LEAP – Can I catch this?

- No, if attacker has patience
- Yes, if he/she uses combined Deauth attacks
- Best prevention: do not use LEAP
Security Through Obscurity – Hidden SSID

Proven wrong over and over again
Hidden SSID (non-Broadcast)

- Still a common topic: people hiding the SSID as security measure
- Clients prefer broadcasted SSID
- No RF time savings by removing it
- Only benefit: accidental “double” click by guest/passing users
- Also, complying with auditors
Hidden SSID (non-Broadcast)

- Can we find it?

- Not in the beacons…
Hidden SSID (non-Broadcast)

- Just wait for client… or deauth it so it has to reconnect
WPA Preshared Key Attack

Nice and easy, not up to Enterprise security levels
WPA(2)-PSK

- Allows secure authentication of 2 parties, without Radius server
- Easy implementation, designed for home use
- If key is compromised, you need to change all devices in the network
- As any shared key algorithm, it is vulnerable to brute force/dictionary attacks
- Key space is from 8 to 63 bytes
- If you have PSK + EAPoL exchange, you can decrypt traffic!
WPA(2)-4way handshake

Supplicant

Snonce=Random

EAPoL M1 (ANonce, etc)

EAPoL M2 (SNonce, MIC, RSNIE, etc)

EAPoL M3 (Key RSC, Anonce, MIC, RSNIE, GTK[keyID], etc)

EAPoL M4 (MIC, etc)

Auth Server

Snonce=Random
WPA-PSK Recipe

- Main ingredient: EAPoL exchange (M1 to M4) + SSID name
- For 8 characters + numbers: brute force, Cain tool: 750 years
- For 10 characters + numbers: brute force, Cain tool: 673,706 years

Maybe too long!!
WPA-PSK Observations

- A pure brute force classical attack is not really feasible for keys longer than 8 characters
- Dictionary attacks are real, but easy to protect
- Several optimizations in the market make PSK weak
  - GPU based tools
  - Rainbow Tables
WPA-PSK GPU attacks

- Basic laptop 2200 key tests per second on brute force
- High end card 52400 key tests
- This means between 2.5 and 61.5 years vs 750 on same key space when compared to CPU attacks
WPA-PSK Rainbow table attacks

- Rainbow table: look up table, used to recover plaintext password from a password hash
- You can generate your own: pyrit (GPU accelerated)
- You can download: “Church of WiFi” 40 GB for 1000 most common SSID names for 1M possible passwords, generated by FPGA hardware in 3 days
- FPGA for this is open source
- coWPAtty tool can use rainbow table
- If your SSID matches, and the password was in the dictionary: recovering is a matter of seconds.

http://imgs.xkcd.com/comics/security.png
WPA-PSK Summary

- To be secure, you need a real random password, with more than 12 characters (the more the better)
- Avoid pre calculated SSIDs
- Remember, this is WPA and WPA2, irrelevant of encryption used.
- If you got PSK, and EAPoL exchange, you can decrypt all captured traffic for that EAPoL session
- If PSK key is compromised, you need to change all devices: not good for enterprise security
DoS

Killing it…
DoS Attacks

- Resource Starvation
  “You can always send too much of something”
  Workarounds for some of the techniques

- Protocol Based
  “Ping of Death”
  Can be device bugs, or protocol weaknesses

- Wireless has large availability of potential DoS attacks:
  TKIP MIC, Auth flood, Assoc flood, Deauth, NAV, RF Jamming, etc

  Our equipment has detection and prevention countermeasures.
DoS - RF

- Easy to do, easy detection, but effective
DoS TKIP MIC

- Good example of protocol based
- MIC is used as protection for weak checksum algorithm
- If 2 group key errors are detected, AP starts countermeasures killing the BSSID for 60 seconds
- MIC weaknesses are used as part of the TKIP injection attacks (Beck-Tews, Halvorsen, Ohigashi-Morii)
- Effect: you can DoS any TKIP based network
DoS TKIP MIC Recipe

- What do you need?
  - attack tool: mdk3 (covers multiple DoS techniques)
  - You can add to most distributions, or use backtrack
DoS TKIP MIC Recipe
DoS TKIP MIC Can I catch it?

- Yes, MIC errors are reported
  
  ```
  show trap1
  
  Number of Traps Since Last Reset ............ 427
  Number of Traps Since Log Last Displayed .... 2
  
  Log System Time Trap
  --- ------------------------ ---------------------------------------------------
  1 Mon May 3 15:26:11 2010 WPA MIC Error counter measure activated on Radio
  with MAC 00:1c:b0:ea:63:00 and Slot ID 0. Station
  MAC Address is 00:40:96:b5:11:19 and WLAN ID is 6
  ```

- You can disable countermeasures, so you are not affected -
  price: more vulnerable to traffic injection attacks

- Be careful with false positives

- Best action: use WPA2+AES
## DoS TKIP MIC Recipe

show traplog

Number of Traps Since Last Reset ............ 427
Number of Traps Since Log Last Displayed .... 2

<table>
<thead>
<tr>
<th>Log System Time</th>
<th>Trap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mon May 3 15:26:11 2010</td>
<td>WPA MIC Error counter measure activated on Radio with MAC 00:1c:b0:ea:63:00 and Slot ID 0. Station MAC Address is 00:40:96:b5:11:19 and WLAN ID is 6</td>
</tr>
</tbody>
</table>
DoS 802.11 Floods

- Management traffic on 802.11 can be spoofed
- Floods can lead to resource exhaustion:
  - Thousands of source addresses trying to associate to same AP
  - Memory is reserved, association IDs used, CPU utilization is high, etc
- No easy solution:
  - Rate limiting on AP
  - Client MFP only limited protection
- Impact is limited: temporary loss of service
DoS in 802.11

- How to protect:
  WLC, 12.4(21a)JY have DoS auth prevention

- Detection is best option
  - Floods show clearly on WLC default signatures
  - Protocol attacks like NAV changes can be easily detected
  - WCS maps can show approximate location
  - MFP can detect spoofed Aps
  - Rogue rules can easily catch fake APs
XXS

Sneaking the link
XSS – Cross-site scripting

- Injecting client side script into webpages
- One of the most common types of attacks
- It has opportunistic component
XSS – Cross-site scripting

- Some clients have protection (IE8 for example)
XSS – Cross-site scripting

- Cisco does extensive tests on all web applications
- User side should not be forgotten
- Intended action will be under the user security context
Webauth Highjack

No L2 security, gets you that.
Webauth Highjack

- Used on hotspot and guest access networks
- Provides Web based authentication over an open L2 network

Welcome to the Cisco wireless network

Cisco is pleased to provide the Wireless LAN infrastructure for your network. Please login and put your unified wireless solution to work.

User Name 
Password 
Submit
Webauth Highjack

- Open networks means no security protection at wireless level
  - No encryption or authentication!
- It is perfectly possible to launch attacks at L2 level, and highjack connections
- MAC spoofing is trivial
Webauth Highjack

- Attack is as simple as either kill valid client, or wait until it is not on the network, then use its address
- 802.11 makes very difficult to prevent MAC spoofing, unless you have L2 auth/encryption (WPA, WPA2)
- Prevention is to use guest on isolated environments (DMZ), and limit the APs servicing it
- Alternative: PSK + Webauth
IP Address attacks

IP Spoofing, highjack and others
IP Address Attacks

- Not specific to Wireless
- Mentioned here due to several mechanisms present on WLC to provide protection
  - DHCP Required
  - IP/MAC Address binding
  - IP Theft
  - ARP spoofing
- IP can be trivially spoofed. Objective could be:
  - Highjack connection (webauth, traffic – default gateway, etc)
  - DoS. (prevent connection to a resource)
DHCP Attacks

- Exhaust of DHCP addresses
  - Easy on wired, not so on wireless due to 802.11 process needed

- DHCP spoofing
  - Answering on behalf of server.
  - It could be accidental sometimes.
  - Option 82 can help
  - WLC makes unicast to server on default configuration

- Controller can enforce protections on wireless side

- DHCP required: nice, but be careful with non-windows clients.
  - Any new associated client must complete DHCP
IP Attacks

- ARP spoofing
  - WLC is ARP proxy all the time
  - No direct client to network ARP traffic by default
  - Used to redirect traffic (typically default gateway)
  - Newer versions will blacklist client if spoofed ARP matches default gateway

- IP Spoofing
  - IP/MAC address binding in 5.2 and higher prevents address change, this could break Mac OS due to DNAv4 (network discovery)
  - On webauth, it triggers new webauth authentication
IP Attacks

- IP Spoofing detection:
  
  `*%APF-4-REGISTER_IPADD_ON_MSCB_FAILED: apf_foreignap.c:1331
  Could not Register IP Add on MSCB. Identity theft alert for IP address. Address:00:40:96:b5:11:19`

- DHCP Required in action

  `%DTL-1-ARP_POISON_DETECTED: dtl_net.c:1394 STA(Target MAC Address) [00:40:96:b5:11:19, 0.0.0.0] ARP (op ARP REQUEST) received with invalid SPA(Source IP Address) 192.168.32.94/TPA(Destination IP Address) 192.168.32.1`

- IP address change prevention:

  ![Image of Wireshark capture](image-url)
Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Cisco Preferred Access points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

Don’t forget to activate your Cisco Live and Networkers Virtual account for access to all session materials, communities, and on-demand and live activities throughout the year. Activate your account at any internet station or visit www.ciscolivevirtual.com.