Saint Louis ISSA

Wireless (In)Security

Overview

- About the Presenter
  - chris@riosec.com
  - http://linkedin.com/in/cbyrd01
  - http://twitter.com/cbyrd01
- What we'll cover
  - Wireless Overview
  - Enterprise Wireless Security
  - Hotspot (Guest) Wireless
  - Demos

Wireless Overview

Wireless Alphabet Soup

- 802.11a – 5GHz OFDM
- 802.11b – 2.4GHz DSSS
- 802.11g – 2.4GHz OFDM
- 802.11n – 2.4/5GHz OFDM

- 802.11i – RSN
- 802.11w – protection of management frames

Terminology

- Wi-Fi™ vs. Wireless LAN (WLAN)
- RF - Radio Frequency
- War[walking|cycling|driving|flying]

- AP - Access Point
- STA - Station (client system)

- BSS - Basic Service Set - AP and STAs
- BSSID - Basic Service Set Identifier

- ESS - Extended Service Set (one or more BSS+LAN)
- ESSID (aka SSID) - Extended Service Set Identifier

- IBSS - Independent Basic Service Set (Ad-Hoc)

Examining Wireless

- DEMO: Wireshark / AirPcap
- DEMO: Wi-Spy

Encryption

- None (Open)
- Static WEP
- Dynamic WEP
- WPA (TKIP)
- WPA2 (CCMP)

Encryption: None (Open)

- We will come back to this...

Encryption:
Wired Equivalent Privacy

- Uses shared key for encryption
- RC4 of IV + WEP key, length creates PRGA
- Data is XOR’d with PRGA and transmitted
- However…
  - Cryptographic issues including:
    - Reuse of IV values
    - Known plaintext
    - Key selection issues
  - WEP can be cracked in minutes

Encryption: Wireless Protected Access

- Intermediate measure by Wi-Fi Alliance
- Based on draft of 802.11i

- Uses Temporal Key Integrity Protocol (TKIP)
- TKIP still uses RC4, but adds
  - Key mixing
  - Counter (prevent replay attacks)
  - Michael Message Integrity Check (prevents packet injection)

- However…
  - New attacks (Beck-Tews, Ohigashi-Morii)
  - Not broken, but showing cracks
  - Only meant as transitional protocol for hardware

Encryption: Wireless Protected Access 2

- IEEE 802.11i Robust Security Networks (RSN)
- Called WPA2 by Wi-Fi Alliance

- Uses Counter Mode with Cipher Block Chaining Message Authentication Code (CCMP), based on AES

- TKIP is still optional

- However…
  - Make sure TKIP is disabled (unless still in transition)
  - Still open to authentication issues (up next…)

Authentication

- Shared Key
  - WEP
  - WPA-PSK
- EAP

Authentication: Shared Key

- Wired Equivalent Privacy (WEP)
- Wireless Protected Access - Pre-Shared Key (WPA-PSK)

- Easy to set up

- However…
  - No (secure) key distribution
  - No perfect-forward secrecy
  - Offline key attacks
  - Online key attacks
  - Device theft

- DEMO: Wireshark PSK decryption

Authentication: EAP

- Extensible Authentication Protocol
- Transmitted using EAP Over LAN (EAPOL)
  - Created for wired in 802.1X standard, extended to wireless
- Three parts:
  - Supplicant – the client
  - Authenticator – Access Point
  - Authentication Server – RADIUS server

Authentication: EAP Types

- LEAP – Cisco Lightweight Extensible Authentication Protocol
- PEAPv0/MSCHAPv2 – Protected EAP / Microsoft Challenge and Response Protocol v2
- PEAPv1/EAP-GTC – Generic Token Card
- FAST - Flexible Authentication via Secure Tunneling
- TLS – Transport Layer Security
- TTLS – Tunneled Transport Layer Security

Auth: EAP-PEAPv0/MSCHAPv2

- Uses Protected EAP (PEAP)
  - Essentially TLS without client certs
  - Windows discloses identity in outer PEAP
- Inside uses standard Microsoft CHAP (MSCHAP) v2
- Network is an open network
  - Attacker free to connect, brute force passwords
  - Offline attacks possible against MSCHAPv2 with Asleep

Authentication: EAP-TLS

- Same TLS used in HTTPS used to transport keying material
- “Requires” client certificates – more on this later
- Client identity disclosed in SubjectName of client certificate
- Addressed in EAP-PEAPv0/EAP-TLS
  - However… limited supplicant support

Wireless Security Myths

- Disable broadcast SSID to cloak your network
- Use MAC address filtering to keep out bad guys
- Disable DHCP so an attacker won't get an IP
- 802.11n can replace wired connections
- DEMO: Kismet

Building a Wireless Lab

- One or more wireless adapters
  - For Windows, need AirPcap TX
  - For Linux, anything supported by Aircrack-ng project should work
- Access Point
  - Linksys APs + OpenWRT = Multiple BSSIDs
  - Linux based access point
- Backtrack 4 distribution is handy

Enterprise Wireless

Challenge 1: Wireless Rogues

- Malicious (placed by an attacker) or non-malicious (placed by an insider)

- Uses cheap, off the shelf hardware or built in software

- Difficult to detect

Types of Rogues

- Hardware - cheap, off the shelf access points
  - Including Bluetooth APs
- Configuration - Ghost in the AP
- Software - Linux, Windows
  - Windows 7 introduced virtual AP - doesn't interfere with normal operation of the client!

```
netsh wlan set hostednetwork mode=allow
ssid=linksys key=sekretbackd00r
```

Detecting Rogues - Over the Air

- Wireless IDS Sensors
- Manual walk-through

- However...
  - Is it on my network?
  - Where is it physically located?
  - What if they use Bluetooth?

Detecting Rogues - On the Network

- Network scanning
  - Nmap (rogueap.nse)
  - Nessus (find_ap.nasl)

- Passive detection
  - DHCP server logs
  - Netflow (TTL analysis)

- However...
  - Network scanning false negatives (cloaked/firewalled)
  - Passive detection false positives (complex environment)

Challenge 2: Client Attacks

- Evil Twin attacks
- Open wireless
- Shared key
- VPN bypass
- Data disclosure

- Resulting in...
  - System compromise, and pivoting

Challenge 3: Weak Encryption

- WEP can be broken in minutes
- WPA is showing it's age
- Misconfiguration can enable TKIP on WPA2

Challenge 4: Weak Authentication

- Shared key issues
- Password brute forcing
- Unintended EAP types
- PEAP and TLS certificate validation issues

Challenge 5: Denial of Service

- Physical layer
- Resource reservation
- Management frames

Enterprise Summary

- SSID broadcast enabled
- Don’t disclose info in SSID name
- WPA2 (CCMP only)
- EAP-TLS
  - Client settings defined by GPO
    - Disable Ad-hoc wireless
    - Define preferred network
      - Validate server certificate
      - Specify server names in Connect to these servers…
      - Select specific certificate authority
      - Select “Do not prompt user to authorize new servers or trusted certificates” <- Important
- Nothing can be done about Denial of Service

Hotspot (Guest) Networks

Challenge 1: Portal Bypass

- TCP over DNS
- TCP over ICMP
- Cloning existing sessions
  - MAC and / or IP
- Attack the authentication system

Challenge 2: Information Disclosure

- Like having your system connected to a projector and copy machine
- Pay as you go hotspot CC#s
  - How do these ever pass PCI?
- What about using SSL / VPN?

SSL Issues

- Remember, the attacker owns the medium
- Man in the middle the SSL
  - Ssldump, sslstrip
- Sidejacking
  - Ferret, hamster
- SSL renegotiate flaw
- New research on AJAX SSL sniffing

VPN issues

- Information disclosed as soon as interface comes up
  - System name
  - Internal names / IP addresses
  - User name
- Own the system before VPN starts
  - Cached IFRAMED web pages
  - DNS poisoning
  - Evilgrade, NTLM reflection, more
- What happens if the attacker just blocks VPN?

Challenge 3: Guest Security

- Attack the clients
  - AirPWN
  - KARMA / Karmetasploit
- Evil twin attacks
  - With no keying material, how do you tell the difference?

Solution: Open Secure Wireless

- EAP-TLS does NOT really require a client certificate.

> “The certificate_request message is included when the server desires the peer to authenticate itself via public key. While the EAP server SHOULD require peer authentication, this is not mandatory…” – RFC 5216

- HTTPS would never had become popular if you had to have a client cert to connect
  - Chicken-and-egg problem

Open Secure Wireless

- Although the RFC clearly specifies that CertificateRequest is optional, all servers and clients currently treat it as mandatory.
- This is where the perception of the requirement comes from
  - It’s an implementation problem

Authentication Server Support

- I was able to modify the source for hostapd so that it doesn’t ask for a client certificate.
- It’s a one ***bit*** change. ☺

src/eap_server/eap_tls.c

```
68c68
<  if (eap_server_tls_ssl_init(sm, &data->ssl, 0)) {
---
>  if (eap_server_tls_ssl_init(sm, &data->ssl, 1)) {
```

- Changes it to behave just like PEAP – never asking for a client cert, but moves state machine to SUCCESS

Authenticator (AP) Support

- Access Points just pass EAPOL to the Authentication Server, EAP types are transparent
- Therefore existing APs work without modification

Supplicant Support

- The bad news: Windows, Linux supplicants require a client certificate
- Basic IF statement:

```
wpa_supplicant-0.6.9/src/eap_peer/eap_tls.c
if (config == NULL ||
        ((sm->init_phase2 ? config->private_key2 : config->private_key)
         == NULL &&
         (sm->init_phase2 ? config->engine2 : config->engine) == 0)) {
            wpa_printf(MSG_INFO, "EAP-TLS: Private key not configured");
            return NULL;
    }
```

Supplicant Support

- Configuring supplicants with ANY certificate satisfies this requirement
  - It doesn’t have to be a valid cert
  - They’ll never be asked for it anyway
- Changing wpa_supplicant to remove the if statement removes the certificate requirement
- From observed behavior, the proprietary Windows client works the same way

Open Secure Wireless Results

- You can connect to a secure wireless network that mitigates ALL of the above hotspot issues, without client authentication
- Hotspot operators can still run a captive portal to authenticate visitors
  - And the captive portal is protected at the transport layer
- To be truly useful, would also need UI changes on supplicants

Open Secure Wireless

- DEMO: Windows 7 client on Open Secure Wireless

Future work – Cert validation

- Web browsers compare host name from URI to the CN or SubAltName
- There’s no DNS during EAPOL
- But there is a 32-byte SSID
- Change supplicants to validate SSID against CN or SubAltName
  - wifi.coffeeshop.com, guestwifi.company.com
- Some limitations, but standard CA verification procedures would work

Future work – intended use IE

- Currently WLAN networks don’t advertise their intention
  - Is it open because it’s meant for anyone, or because the owner didn’t secure it?
- Use a custom Information Element to advertise intention – public, guest, private, etc.