

Wireless (In)Security



Who am I?

- Joshua Smith
- Worked 400 days in the IT field
- Spent 10 years in the golf industry
- Currently employed at Jones Stephens Corp
- Also operate Symmetric Wireless

Why you should listen to me:

- Recently attended SANS Orlando 2007
- I learn from the best:
 - Jon Elch (aka. Johnny Cache)
 - David Maynor
 - Joshua Wright
- Many hours of reading, testing, and experimenting with the technology
- Shameless plug for SANS and Joshua Wright



Current Condition Of Wireless

- Open – 33%
- WEP – 59%
- TKIP/CCMP (WPA/WPA2) – 8%
- Wireless continues to gain momentum and popularity, but similar to the wired side, security is still somewhat of an afterthought
- Much like the wired side, attacks are focusing on the client side (don't trust the user)

AirPWN

- Released at Defcon 12 (2004)
- Written by toast
- <http://airpwn.sourceforge.net/Airpwn.html>
- Goal: To implement a hotspot injection attack, exploiting the race condition, thereby spoofing legitimate traffic

AirPWN

- Details:
 - “Airpwn listens to incoming wireless packets, and if the data matches a pattern specified in the config files, custom content is injected "spoofed" from the wireless access point. From the perspective of the wireless client, airpwn becomes the server.”

AirPWN

- Details (cont):
 - How you get a web page
 - AirPWN listens to traffic in monitor mode (similar to promiscuous mode on the wired side)
 - Matches observed traffic against regular expression string
 - Replies with data from config file (attacker dictates what that data is)
 - Exploits the race condition (time it takes to retrieve a web page)
- Real World: Demo

openWRT

- <http://openwrt.org/>
- Goal: OpenWrt is described as a Linux distribution for embedded devices (ie. Wireless access points).

openWRT

- Details:
 - Install a lean, flexible, powerful OS (Linux) on a consumer grade wireless access point
 - After install, you can install any available packages, or build your own for custom applications and environments
 - Advanced features build in or easily available to add on

openWRT

- Details (cont):
 - Some packages available:
 - freeRADIUS
 - openVPN
 - Kismet
 - Snort
 - Chillispot (easily set up Wi-Fi Hotspot)
 - nmap
 - Nessus
 - dsniff
 - ettercap
 - aircrack-ng
 - tcpdump

openWRT

- Real World
 - Very flexible
 - Not ideal for large, corporate roll outs (still no unified management available)
 - Very applicable for small companies that want a lot of options, without the associated price tag
 - Powerful tool for security auditors, researchers, and pen tests

KARMA

- Released by Dino A. Dai Zovi and Shane Macaulay
- Released at PACSEC 2004
- <http://www.theta44.org/karma/>
- Goal: "KARMA is a set of tools for assessing the security of wireless clients at multiple layers."

KARMA

- Details:
 - What is a PNL?
 - Monitor mode listening for probe requests
 - KARMA then becomes network probe is looking for (rogue AP)
 - Client can automatically join created AP
 - Once connected, higher level services collect information or exploit victim (DNS, DHCP, POP3, FTP, SMB, etc)

KARMA

- Real World:
 - Attacker runs KARMA, client systems automatically join rogue AP
 - Once joined, AP dishes out content as decided by them, possibly over several different protocols
 - Authors describe as “BYOX” (Bring your own exploit), and state that “automated agent deployment is also planned”
 - Scariest tool I have seen to date

Along the way (Part 2)...

- You never know what you are going to run across

swversion=8.1-01-2-649

platform=tcd/Series2

TSN=xxxxxxxxxxxxxxxx

tivoconnect=1

swversion=8.1-01-2-649

method=broadcast

identity=xxxxxxxxxxxxxxxx

machine=DVR 6116

platform=tcd/Series2

services=TiVo-ServeTcdVideo-1:2191/tvbus_v3,TiVoMediaServer:80/http

Wireless Fuzzing

- Wikipedia - “Fuzz testing or fuzzing is a software testing technique that provides random data ("fuzz") to the inputs of a program.”
- Goal: To send malformed packets to victim software/drivers in order to find flaws in its execution and handling of the packet, possibly leading to buffer overflows and remote code execution.

Wireless Fuzzing

- Details:
 - Garnered a lot of attention @ Blackhat 2006 in presentation by Jon Ellch and David Maynor
 - Generally exploits flaws at layer 2 of the OSI model (most firewalls start at layer 3)
 - Use scapy to inject malformed packets to attackers desired target
 - Client does not have to be connected to any network
 - Vista not immune to this issue
 - BSSID reveals a lot about the card (fingerprinting)

Wireless Fuzzing

- Details (cont):
 - Mostly a client side attack (wireless drivers), but what about exploiting an AP?
 - How about a TiVo?
 - The IEEE 802.11-1999 spec says that the length of the SSID should be between 0 and 32 octets.

from scapy file...

```
p /= Dot11Elt(
    ID=0,          # SSID IE
    len=400,       # Length of "ssid"
    info="/x01/x0") # SSID string
p /= fuzz(Dot11Elt(
    ID=1))        # Supported rates
```

Wireless Fuzzing

- Real World: Fuzzing a box with scapy

Thats it

- Thanks for listening
- Questions?