Pen-testing Wi-Fi
Defcon 2007
Aaron Peterson
Midnight Research Labs
"What you talking about, Willis?"

We're talking about …

- Pen-testing Wi-Fi with a new wireless auditing tool:
  - **Wicrawl**

- Who am I
- Current state of Wi-Fi scanning
- Wi-Fi Penetration testing
- How Wicrawl can help
- How it works
- Use cases and examples
- Screenshots
- Demo!!
- LiveCD software handout
- Wi-finding robot?
Who am I?

Aaron Peterson
(Aaron@MidnightResearch.com, Aaron@AlphaDefense.com)

- Project manager and Developer for wicrawl
- Founder, Midnight Research Laboratories (MRL)
- Co-Founder, Consultant with Alpha Defense
- Network Security Incident Response Team at Harvard University UIS NOC
- Network Security by day, Pen-tester by night
Who is that?

A Network Security Consulting firm based in Boston, MA that specializes in Network and Web Application Penetration testing

http://www.AlphaDefense.com

Midnight Research Labs is a small security research group (San Francisco, Boston, other). With a focus on security and novel computing, MRL has monthly meetings to discuss and stimulate new development on sponsored projects. Come on out!

http://www.MidnightResearch.com
Standard disclosure

None of the views, statements or opinions expressed in this presentation reflect in any way the opinions of my employer.
Current state of wi-fi scanning (old and busted)

Wi-Fi is nearly ubiquitous, but...

- More and more layers of security means varying levels of access (and varying levels of usefulness)

We don't really care about just finding a large number of useless Access Points anymore because:
- Just knowing an access point exists doesn’t tell us much
- Manual configuration and checks are tedious and take too much time (and get too few useful results)
  - Especially for large numbers
  - That and I'm pretty lazy

The inspiration for Wicrawl
AP Information gathering

- Having WEP no longer means we can’t get on an access point
  (WEP is dead)
- An “open” AP no longer means we can …
- Much more information to gather after association

WEP: “You can put lipstick on a pig… but it’s still a pig…”
Moving forward (new hotness)

What we[*] really care about:

- Penetration-testing --> (* Security Professionals *)
- Finding Rogue access points --> (* Every-day IT *)
- Getting (and staying) on the internet --> (* Business Travellers *)
- Finding "useful or interesting" access point --> (* Hackers, Slackers and Code-crackers *)

What's behind that AP? The magical land of Narnia? or the soft chewy underbelly of my corporate network being exposed?

Need to filter, crawl and examine …
Penetration Testing Wi-Fi

- "Traditional" Penetration testing
  - General Confidentiality/Integrity/Availability
  - Similar methodology to other pen-testing activities
    - Reconnaissance
    - Discovery, scanning and enumeration (foot-printing)
    - Vulnerability/Security/Posture assessment
  - Lots of individual tools

- Rogue Access Point Checks
  - A $20 device can often subvert all security
  - Classic eggshell problem

“How many rogue AP’s does it take to get to the center of your network?”
Wi-Fi Pen-testing difficulties

● AP quantity and density
  – More Wi-Fi gear (antennas, amplifiers, etc) makes this even “worse” when looking for rogue APs
  – Takes lots of time to scan (and crawl, or crack, e.g. WPA PSK)
    ● Hackers have more time than auditors
    ● A multitude of tools, but takes time to setup/configure/run

● Geographic issues
  – Multi-level shared buildings, reflections, latency

● Rogue Access Points
  – Hard to tell if it’s an AP you’re authorized to scan
  – Baselines don’t exist
  – Clients/Traffic (and detection) can be bursty
  – Ultimately can’t prove a negative
Common Tools

- **Discovery**
  - Kismet / wellenreiter / netstumbler / kismac / iStumbler

- **WEP**
  - Aircrack-ng suite
    - (e.g. wepcracking, arp injection, client de-authing, WPA crack (PTW/FMS, etc), WPA brute-forcing, chopchop, fragmentation, dumping, tunneling, etc)
  - Wesside **
  - Easside **
  - Airbase / picocrack
  - Weplab
More common tools

- **WPA**
  - coWPAtty / rainbow tables genpmk
  - Aircrack-ng
- **Attacking the client-side**
  - Karma / hotspotter
- **Others**
  - Asleap, THC-LEAPcracker, pickupline, LORCON, wifitap, void11
  - More non-specific tools like nmap, nessus and metasploit, etc
Wordlists are important
  - A large number of passwords are based on company/product data, or a derivative of a default passwords
  - Check out wyd:
    - http://www.remote-exploit.org/codes_wyd.html

Antennas don’t necessarily have to pointed directly at the target to be most effective

People *will* look at you funny (and suspiciously)
Wicrawl can help

- New features for the pen-tester
  - Hardware/FPGA Acceleration (ie. H1kari’s latest work)
  - Better filtering and imported host lists
  - New plugins (metasploit, better captive portal detection and avoidance, etc)
  - Professional reporting (released soon)

- Logical approach

- Automated

- Can cover the whole toolset rather than one at a time

- Parallelized attacks with multiple cards
wicrawl enters the thunderdome...

- Ability to select "goal oriented" wi-fi network checks based on plugins and profiles
- Actually get the info you want -- Don't get the cruft you don't care about!

(Google images rocks)
Wicrawl is:

“... a simple wi-fi scanner and auditor with a flexible and simple plugin architecture with passive discovery and active crawling”

● The Power is in the plugins
  ● Automation of standard tasks, association, DHCP, network-checks, mapping, proxy-check, etc.
  ● Multiple simultaneous Wi-Fi cards for parallel scanning/crawling
  ● Profiles determine when and how scanning is done
  ● Theme-able GTK GUI (with status bar for wardriving)
  ● Extra features: GPSd, TTS, hooks for motorized antenna, reporting (pdf/html/xml/txt)
● http://midnightresearch.com/projects/wicrawl
Wicrawl examples

**Basic example:**
- Does access point discovery
- Associates
- Gets an IP address
- Tries to get to the Internet
- Measures speed/latency

**More Advanced:**
- Runs nmap, nessus
- Triggers metasploit
- Tries to break WEP/WPA-PSK
- Bruteforces WEP dictionary attacks
Under the Hood:

Logical Pieces of wicrawl:

- Discovery Engine
- Plugin Engine
- Plugins
- Profiles
- Reporting
- UI(s)
General Architecture

- User Interface
- Profiles
- Plugin Engine
- Discovery Engine
- Plugins
- Plugins
- Plugins
- Plugins
- Wireless Routers
Discovery Engine

Discovery by itself is similar to what already exists today (e.g. kismet, netstumbler, etc.)

**Wicrawl**
- Passive discovery (Beacons and probes, Oh my!)
- Requires monitor mode (rfmon)
- Handles multiple radio header types
- Pcap traffic dumping
- Sends IPC messages to plugin-engine directly
- Scheduled from plugin-engine
- Written by Jason Spence and Focus
Plugin Engine

- Takes the information that we get from discovery and runs plugins (based on the profile)
- Multiple cards for distributed crawling
- Handles all scheduling decisions:
  - Card per Access Point
  - Turning on and off the discovery engine
  - When to run plugins (hooks/scheduled synchronous/asynchronous as determined by the profile)
- SSID, MAC filtering
- Written by Aaron Peterson
Plugins

- Anything you want them to be
- Super simple interface
- Plugins scheduled by plugin-engine
- AP/state parameters passed into plugin
- Plugin specific config passed in through the environment
- Executable (binary/script/etc)
- Two types:
  - Scheduled
  - Hook
- Bash/Perl/python and even fortran templates exist
- Wraps sometimes difficult to make/build/use tools
- Written by Aaron, Peter Kacherginsky, Focus and you
**Plugins (more)**

**Plugin definitions**

- Event levels
  - New AP
  - Have Association
  - Have IP
  - Have Internet
  - Pre/Post Discovery (hooks only)
  - Pre/Post Access Point (hooks only)
- Can have multiple event levels per plugin
- Run lengths
  - short, medium, long
- Run levels (Plugin ordering, think sysV init)
Workflow

In the UI, select the Cards (and profiles)

**Selecting “Start” triggers:**

- Plugin-engine triggers discovery until plugin-scheduling takes over
- Run “short” runlength plugins for 'new-ap' (the first event-level).
- Run plugins (e.g. association, wep-cracking) in this run-level until we are able to associate, then we run plugins in the next event-level (have-association).
- Continue escalating up the event levels until we're stuck (by finishing all plugins in the runlength/event level without escalating)
- Run through all other Access Points
- After all Access Points have been scanned in this runlength, go back for a second pass with the next run-length (medium)
- Replay plugins to get to the current runlevel
- Start the plugins in the medium runlength starting from current event level
- Wash, rinse, repeat in “long” run length if needed until all scheduled plugins have been run
- Start new discovery run
Plugins: Types

Two different types of Plugins:

- **Scheduled**
  - Handles the tools and are scheduled according to the Profile
  - Synchronous
  - Examples: association, mapping, anything associated with an access point

- **Hooks**
  - More timing sensitive
  - Synchronous, or Asynchronous
  - Examples: GPSd, Antenna movement, TTS
Plugins: Interface

Three ways to communicate with plugins:

● Get the report style human readable input from the **STDOUT** of the plugins. This is recorded in the plugins XML file by the plugin-engine.

● Get the programmatic data back from the plugin through the **return code** (This can signal an event level change).

● The plugin can send pre-defined “messages” to the plugin-engine through the **IPC**.
Existing Plugin Examples

- Association
- DHCP
- Internet checks (speed and bandwidth)
- NMAP or other network scanning
- Aircrack-ng (with PTW)
- Nessus
- Bruteforcing (weplab and coWPAtty)
- MAC spoofing
- Metasploit
- GPSD and Text to speech
- And more! …

Future:
- Even better captive proxy handling
- Continue to improve Rogue AP checks
- dsniff, ettercap, etc
Aircrack-ng plugin

- Starts monitor mode
- Starts airodump to gather traffic (IVs)
- Looks for clients participating on the network
- Sends a de-auth to the broadcast
- Sends a de-auth to each client
- If after a while we still don’t see clients, re-de-auth
- Starts aireplay with --fakeauth for the client with the most packets
  - If fake-auth fails it will check again for the best client to spoof
- Run aireplay arp inject attacks to inject traffic (and generate IVs).
  - If after a while we don’t see any arp traffic, re-de-auth
- Runs aircrack-ng once we get enough packets to start
FPGAs and Hacking faster

- H1kari’s coWPAtty patches (part of open ciphers, openciphers.sf.net)
- H1kari has done a lot of great work in FPGA accelerated cracking
- Wicrawl plugin:
  - Takes .pcap file from discovery and checks for a 4-way handshake
  - Runs tcpdump until it finds one
  - Starts the appropriate coWPAtty client based on whether it sees a pico computing FPGA
- 30cps with laptop, 410cps with FPGA
  - (a week to a month job turns into three months to a year of cracking time by using a FPGA)
# Performance Comparison

## PC

<table>
<thead>
<tr>
<th>Device</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cowpatty 800MHz P3</td>
<td>~25/sec</td>
</tr>
<tr>
<td>3.6GHz P4</td>
<td>~60/sec</td>
</tr>
<tr>
<td>AMD Opteron</td>
<td>~70/sec</td>
</tr>
<tr>
<td>2.16GHz IntelDuo</td>
<td>~70/sec</td>
</tr>
<tr>
<td>Aircrack 3.6GHz P4</td>
<td>~100/sec</td>
</tr>
</tbody>
</table>

## FPGA

<table>
<thead>
<tr>
<th>Device</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cowpatty LX25</td>
<td>~430/sec</td>
</tr>
<tr>
<td>15 Cluster</td>
<td>~6,500/sec</td>
</tr>
<tr>
<td>LX50</td>
<td>~650/sec</td>
</tr>
</tbody>
</table>

(Stolen from h1kari’s talk)
Architecture: Plugins
# The name of the plugin
$name="Example PERL Plugin";

# The binary file to run
$bin="my_plugin.pl";

# Version number of the plugin
$version="0.1";

# Card requires to be in monitor mode or not...
#monitor=yes|no
$monitor="no";

# Length the plugin will take to run
# examples dhcpd would be short, aircrack would be long
#runlength=short|medium|long
$runlength="short";

# Whether this plugin is offline
#offline=yes|no
$offline="no";

# plugin suggested "runlevel"
# 0-99
$runlevel=11;

# event to register for
$event="associated";

# timeout value
$timeout=30;
Profiles

- Determines "goals"
- Card scheduling types
  - First
  - All
  - Traffic
  - Signal
- What run lengths we want to run
- Persistent plugin path
- Plugin overrides
  - Eventually everything
Profile Examples

- **Pen-testing**
  - 'All' card scheduling
  - Schedule all plugins
  - Short, Medium and long run lengths

- **Wardriving**
  - 'First' card scheduling
  - Schedule only basic, short or even no plugins
  - Short runlengths only

- **Holding Internet Access**
  - 'Signal' card scheduling
  - Only basic plugins, plus hold internet plugin
  - Probably short runlengths only
UI(s)

**wicrawl-gtk**
- Sexy
- Plugin/profile configuration
- Runs plugin-engine
- Themes (think night-time)
- Reads input from XML (APs, and plugin output)
- War-driving roll-up status bar
- Written by Peter Kacherginsky

*Curses based UI in alpha*
- So we can run on WRT54G
  - *(wifly)*
Architecture:
UI

- User Interface
  - fork/exec’d with initial selections (cards/profiles)
  - Plugin output tree held in XML
  - Logfile for status
  - Access Point “Database” held in XML updated by plugin engine

Plugin Engine
<table>
<thead>
<tr>
<th>SSID</th>
<th>BSSID</th>
<th>Time</th>
<th>Packets</th>
<th>Plugin</th>
<th>Event</th>
<th>Timestamp</th>
<th>Encryption</th>
<th>Power</th>
<th>Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>wi-food</td>
<td>00:12:17:28:15:5b</td>
<td>0</td>
<td>0</td>
<td>Internet Speed Check</td>
<td>have-internet</td>
<td>3-7-2006 13:8:7</td>
<td>WEP</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Frog</td>
<td>00:0f:66:95:a0:bd</td>
<td>0</td>
<td>0</td>
<td>iwconfig association</td>
<td>new-ap</td>
<td>3-7-2006 13:9:12</td>
<td>None</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>linksys</td>
<td>00:06:25:54:a6:c1</td>
<td>0</td>
<td>0</td>
<td>DHCP</td>
<td>associated</td>
<td>3-7-2006 13:8:38</td>
<td>None</td>
<td>0</td>
<td>01</td>
</tr>
</tbody>
</table>

Output:

```
[?] Found no new APs in discovery, I'll wait a bit more...
   (last count [3] new count [3])
[?] Found no new APs in discovery, I'll wait a bit more...
   (last count [3] new count [3])
[?] Found no new APs in discovery, I'll wait a bit more...
   (last count [3] new count [3])
Stop was pressed
Killing child [24879]
Child [24879] dead
Discovery and plugin-engine finished
```
<table>
<thead>
<tr>
<th>SSID</th>
<th>BSSID</th>
<th>Plugin</th>
<th>Event</th>
<th>Timestamp</th>
<th>Encryption</th>
<th>Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoldenTree</td>
<td>00:11:93:18:00:20</td>
<td>Internet Speed Check</td>
<td>have-internet</td>
<td>7-3-2007 23:57:21</td>
<td>None</td>
<td>52</td>
</tr>
<tr>
<td>mrl-wep</td>
<td>00:18:f8:4d:37:3b</td>
<td>iwconfig association</td>
<td>new-ap</td>
<td>7-3-2007 23:59:11</td>
<td>WEP</td>
<td>52</td>
</tr>
<tr>
<td>mrl-open</td>
<td>00:16:b6:28:7e:77</td>
<td>NMAP Plugin</td>
<td>have-ip</td>
<td>7-3-2007 23:58:10</td>
<td>None</td>
<td>52</td>
</tr>
</tbody>
</table>
Wicrawl - Midnight Research Labs

Access Points
- GoldenTree (00:11:93:18:00:20)
- mrl-wep (00:18:f8:4d:37:3b)
- mrl-open (00:16:b6:28:7e:77)

Plugins
- text_to_speech
- text_to_speech
- apidentd
- iwconfig_associate
- dhcp
- check_internet
- nmap_plugin

Plugin Output

---
[*] Attempting ICMP check to host [64.71.137.162] through [rausb0]
PING 64.71.137.162 (64.71.137.162) from 192.168.3.110 rausb0: 56(84) bytes of data.

--- 64.71.137.162 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3008ms,
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3008ms,

pipe 2
[*] ICMP Check Failed!
[*] HTTP Check Failed.
[*] DNS Check Failed!
[*] Internet check for ICMP/DNS/HTTP failed!!

---

[...] Reaped child [9613], scheduling interface [rausb0]
[...] Forked [9614] to manage [mrl-open] with [rausb0]
[**] Running plugins for Access Point [mrl-open]
[!] There are no plugins configured for this event and run length
[...] Child managing [mrl-open] with [rausb0] exiting now...
[...] Child [9614] finished. (wait returned [9614])
[!] Children finished.
[**] Wicrawl run [0] finished, starting next run
Stopping...
Child [9442] dead
Discovery and plugin-engine finished
Wicrawl - Midnight Research Labs

Total APs: 3  Encrypted: 1  Ratio: 33%  Packets: 0

Access Points
- GoldenTree
  (00:11:93:18:00:20)
- mrl-wep
  (00:18:f8:4d:37:3b)
- mrl-open
  (00:16:b6:28:7e:77)

Plugins
- text_to_speech
- apidentd
- checkinternet
- cowpatty-wpa-psk-bruteforce
- ettercap
- extip
- gpsd
- iwconfigassociate
- metasploit-autopwn
- nmmaplocale
- nmap_plugin
- pickupline
- randommac
- testme
- weplab-bruteforce

Active plugins:
- aircrack-wep-cracking
- apwebcrack
- checkspeed
- checkspeed
- dhcp
- example-bash
- example-perl
- findip
- holdinternet
- iwconfigold
- nessus
- rogueapcheck
- texttospeech

NOTE: All changes will be saved to custom profile

OK
Status:

- Full Release
- Linux only this release
  - BSD/Mac next targets
- Few bugs, and some plugin cleanup
- Card support needs to be (pre)validated
- Plugins -- Need more!!
- Need to test/complete TUI
- Need to finish pdf “professional” reporting
- metasploit & wesside plugins released soon!™
Future -- Infinity and Beyond!!

- Multiple computers
- Multi-Plexing APs (2.0)
- Multiple card discovery (close)
- Plugins, plugins, plugins
- Info registry
- Card capabilities database
  - (Lorcon?)
- Plugin reporting formats
- Ultra-mega-AP-scanning behemoth
  - Wicrack – Wi-fi distributed cracking flash mob
Wi-fi Scanning/crawling liability

- Only you are responsible!
  - Sticky case-law and enforcement (examples)
- If you're not sure, only scan your own APs
- Use AP filters to restrict scanning and crawling
- Use non-invasive profiles when appropriate
- Pen-testers – ALWAYS GET PERMISSION, contracts, insurance, etc.

Thanks to:

- Midnight Research Labs
- Peter Kacherkinsky
- Jason Spence
- Focus
- Vanessa Peterson (my wonderful wife)
- Defcon -- w00t!
- Mati/Muts and the Backtrack project
- aircrack-ng and Christophe Devine
- Jose Ignacio Sanchez (weplab)
- H1kari and Pico Computing
- Josh Wright (coWPAtty)
- Jennifer Grannick

And you!
Demo and LiveCD handouts

- Real Live Demo
- LiveCD based on Backtrack

References

- http://midnightresearch.com
Other related projects:

- Wi-finding robot
  - R/C base
  - Motorized bi-quad antenna
  - Webcam and IR distance sensor
  - Mounted laptop as the brains (running wicrawl, :)
- Make controller
- Wicrawl plugins
  - Tell bot when to search
  - Move antennas, and record location. Replays antenna location for each AP and runs other plugins
- DEMO!
- Wifly?