

Pen-testing Wi-Fi

Defcon 2007

Aaron Peterson



**Midnight
Research Labs**

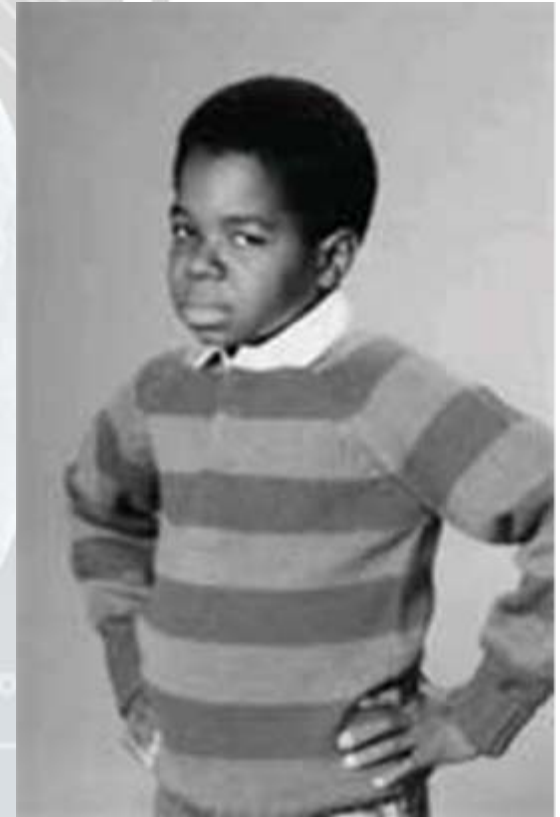
"What you talking about, Willis?"

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**

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

/dev/urandom notes

- 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

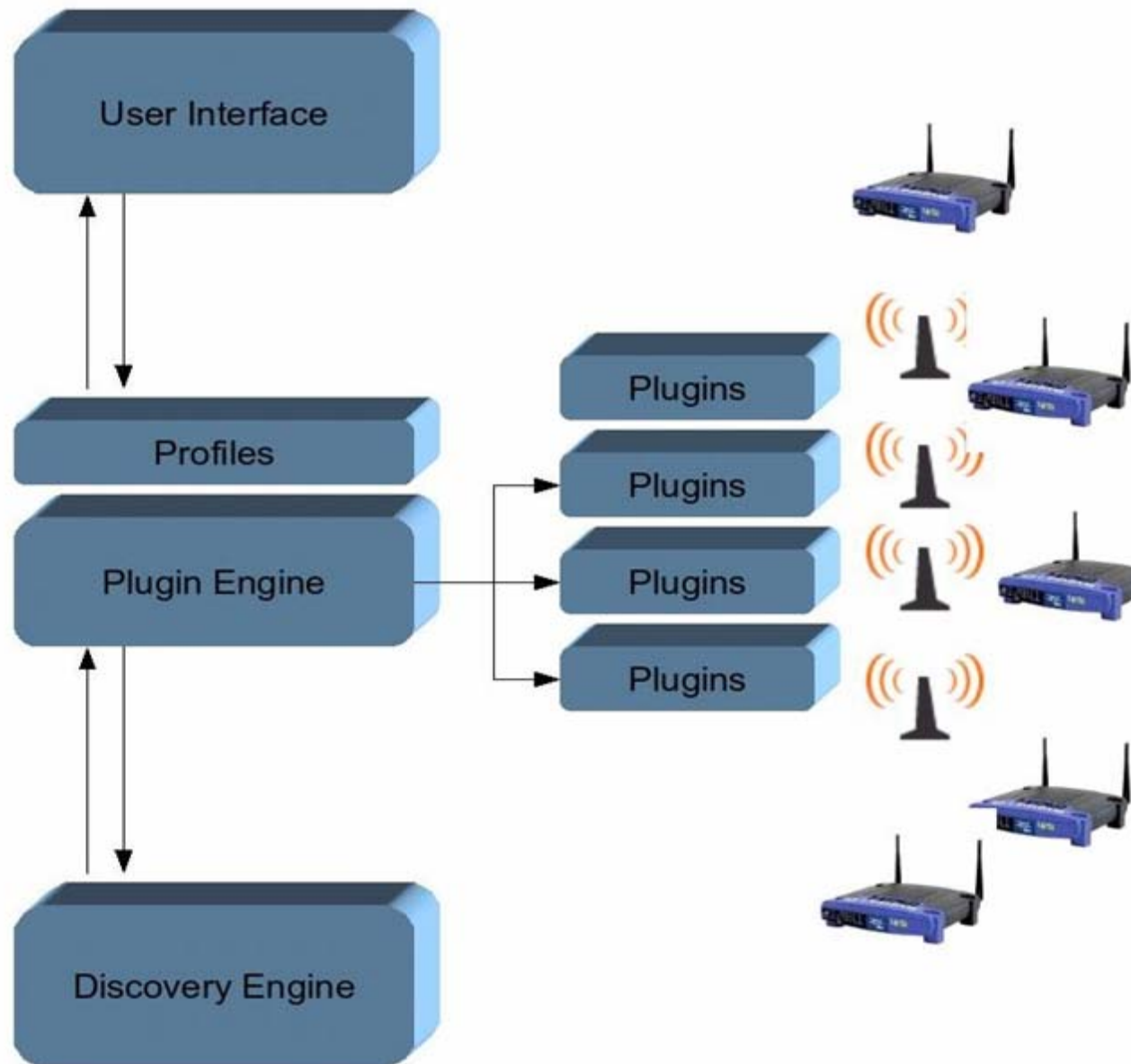
Under the Hood:

Logical Pieces of wicrawl:

- . Discovery Engine*
- . Plugin Engine*
- . Plugins*
- . Profiles*
- . Reporting*
- . UI(s)*



General Architecture



Discovery Engine

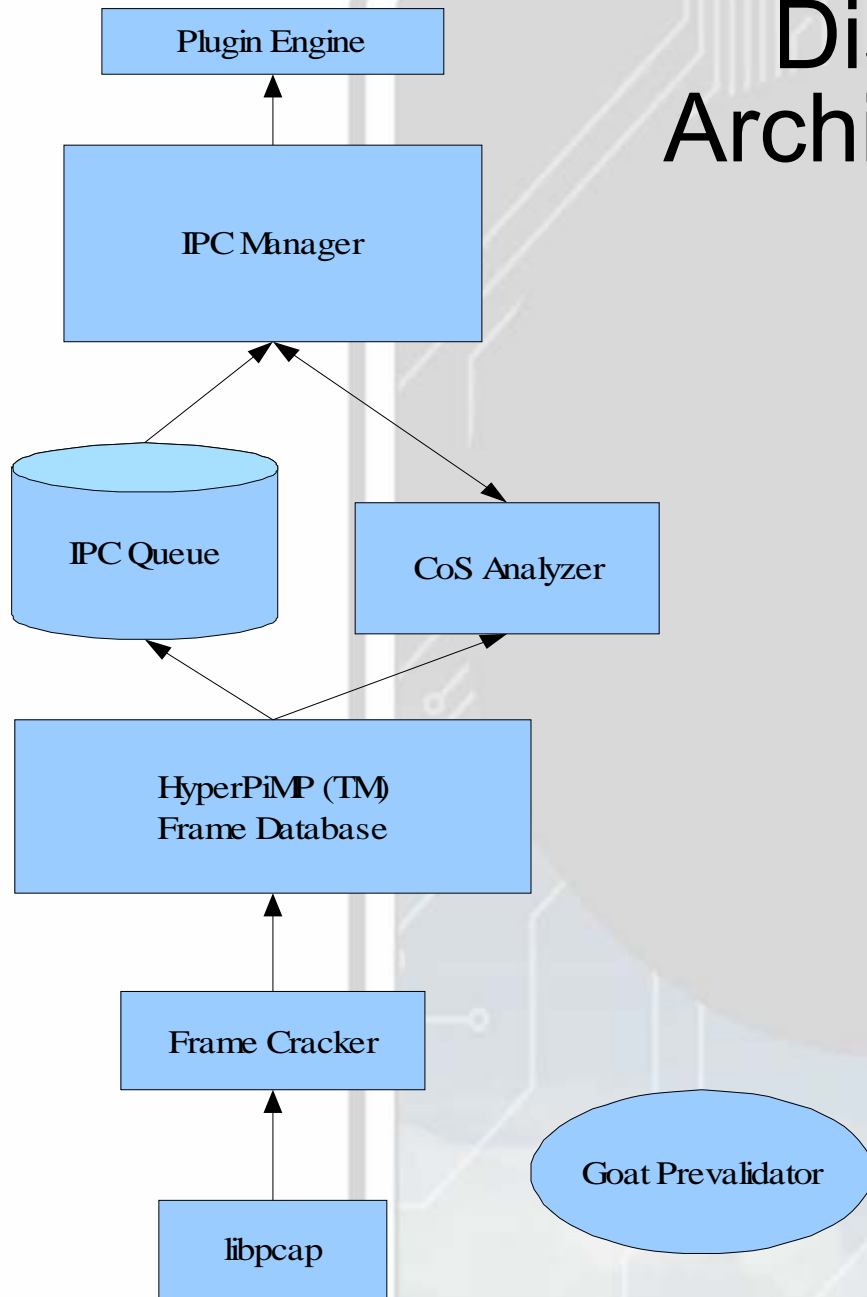


Discovery by itself is similar to what already exists today (e.g. kismet, netstumbler, etc.)

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

Discovery Engine Architecture Overview



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*

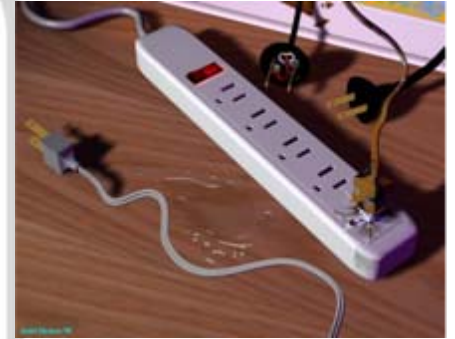
Plugins: Interface

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*



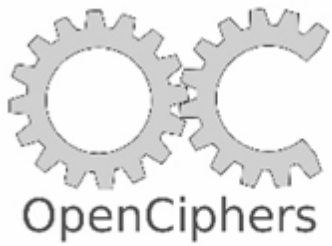
Aircrack-ng plugin



- 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)



(Stolen from h1kari's talk)

Performance Comparison

PC

Cowpatty

800MHz P3	~25/sec
3.6GHz P4	~60/sec
AMD Opteron	~70/sec
2.16GHz IntelDuo	~70/sec

Aircrack

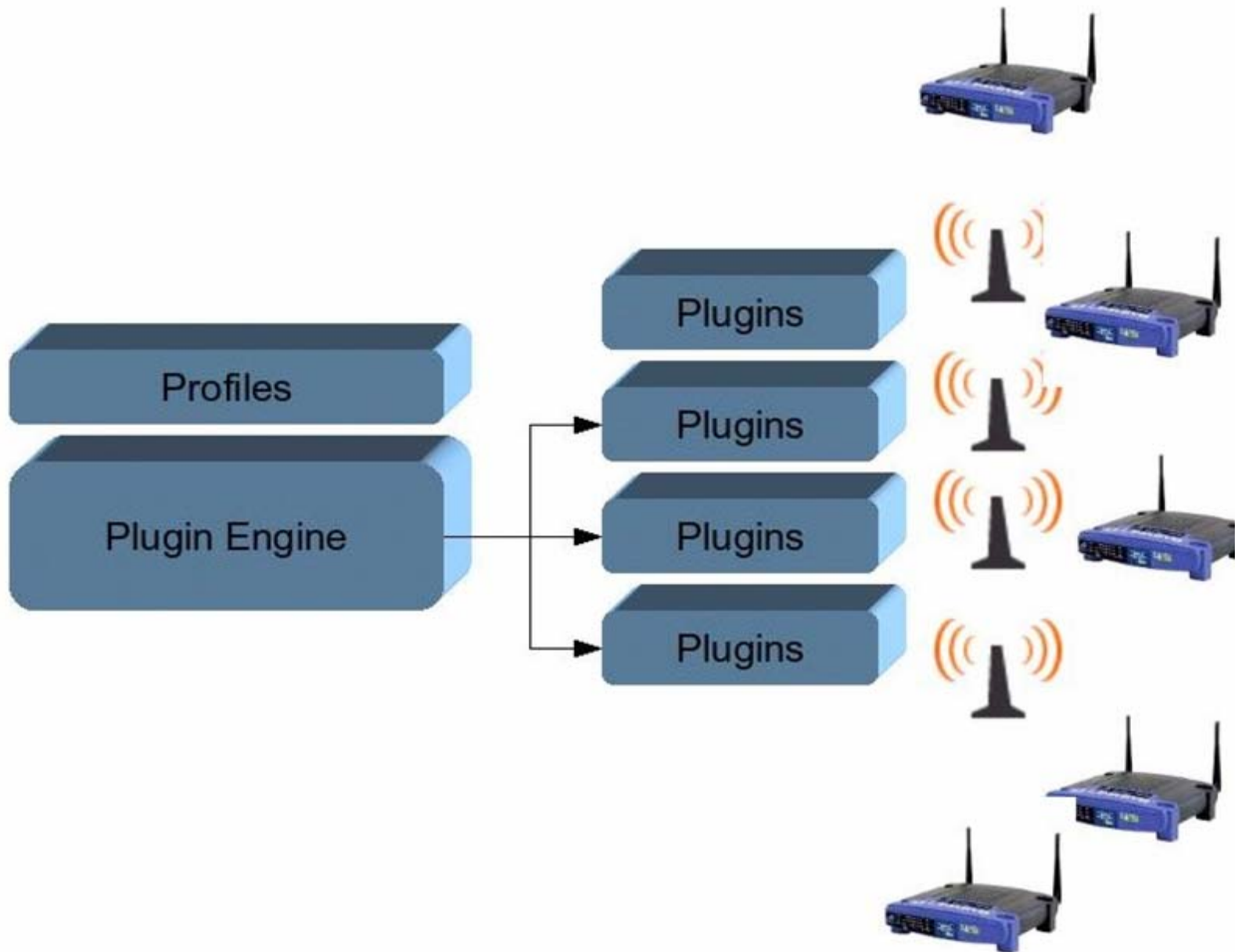
3.6GHz P4	~100/sec
-----------	----------

FPGA

Cowpatty

LX25	~430/sec
15 Cluster	~6,500/sec
LX50	~650/sec

Architecture: Plugins



Plugin Writing:

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)

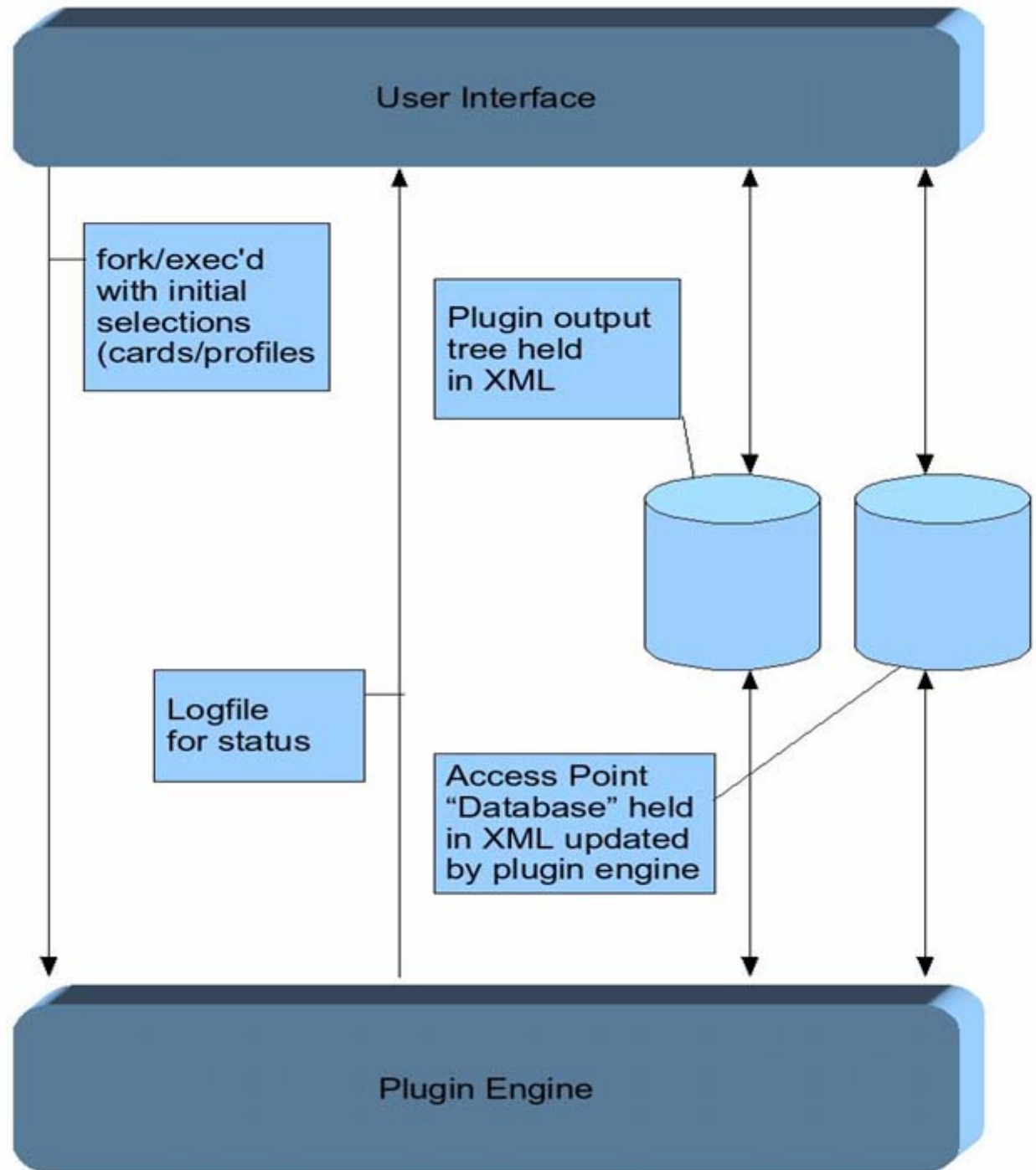
wicrawl-gtk

- *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



SSID Filter



Start



Plugins



Interfaces



Profiles



Minimize



AP Information ×

Plugin Information ×

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

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
```

Wicrawl Interfaces Profiles Plugins Filter View Reports Help

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



	SSID	BSSID	Plugin	Event	Timestamp	Encryption	Channel	
Basic	<input checked="" type="checkbox"/>	GoldenTree	00:11:93:18:00:20	Internet Speed Check	have-internet	7-3-2007 23:57:21	None	52
Advanced	<input checked="" type="checkbox"/>	mrl-wep	00:18:f8:4d:37:3b	iwconfig association	new-ap	7-3-2007 23:59:11	WEP	52
	<input checked="" type="checkbox"/>	mrl-open	00:16:b6:28:7e:77	NMAP Plugin	have-ip	7-3-2007 23:58:10	None	52

Output Plugin Output(rausb0) Plugin Output(wlan0) Plugin Output(eth0)

```

[-] 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

```

Total APs: 3

Encrypted: 1

Ratio: 33%

Packets: 0

1

1

1

Basic

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)

Advanced

Plugins

text_to_speech

text_to_speech

apidentd

iwconfig_associate

dhcp

check_internet

nmap_plugin

Plugin Output

```

===== <internet check plugin> =====
[*] 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
, pipe 2
[!] ICMP Check Failed!
[*] Attempting HTTP external check to [64.71.137.162]
[!] HTTP Check failed.
[*] Attempting DNS TXT check to host [64.71.137.162]
[!] DNS Check Failed!
[!] Internet check for ICMP/DNS/HTTP failed!!
===== <internet check plugin > =====

```

Output

Plugin Output(rausb0)

Plugin Output(wlan0)

Plugin Output(eth0)

```

[-] 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

```

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

1

1

1

Basic

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

```

===== <nmap plugin> =====
[*] Your IP is: 192.168.3.110
[*] Scanning Local Network on [192.168.3.110/24]
[*] Executing [/usr/bin/nmap -sS -n -A -T4 192.168.3.110/24]
[-]
[-] Starting Nmap 4.20 ( http://insecure.org ) at 2007-08-03 23:58 PDT
[-] Interesting ports on 192.168.3.1:
[-] Not shown: 1696 closed ports
[-] PORT      STATE SERVICE VERSION
[-] 80/tcp open  http   Linksys WRT54GL wireless-G router http config
[-] MAC Address: 00:16:B6:28:7E:75 (Cisco-Linksys)
[-] Device type: general purpose|WAP|storage-misc
[-] Running: Linux 2.4.X, Linksys Linux 2.4.X, Asus Linux 2.4.X, Maxtor
Linux 2.4.X
[-] OS details: Linux 2.4.20 - 2.4.32, Linux-based embedded device
(Linksys WRT54GL WAP, Buffalo AirStation WLA-G54 WAP, Maxtor Shared
Storage Drive, or Asus Wireless Storage Router)

```

Output

Plugin Output(rausb0)

Plugin Output(wlan0)

Plugin Output(eth0)

```

[-] Reaped child [9613], scheduling interrace [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

```

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

Basic

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)

Advanced

Plugins

text_to_s

text_to_s

apidntd

iwconfig_

dhcp

check_int

nmap_plu

custom profile

Active plugins:

- | | |
|---|--|
| <input checked="" type="checkbox"/> apidentd | <input type="checkbox"/> aircrack-wep-cracking |
| <input checked="" type="checkbox"/> checkinternet | <input type="checkbox"/> apwebcrack |
| <input type="checkbox"/> cowpatty-wpa-psk-bruteforce | <input checked="" type="checkbox"/> checkspeed |
| <input type="checkbox"/> ettercap | <input checked="" type="checkbox"/> dhcp |
| <input type="checkbox"/> example-fortran | <input type="checkbox"/> example-bash |
| <input checked="" type="checkbox"/> extip | <input type="checkbox"/> example-perl |
| <input type="checkbox"/> gpsd | <input type="checkbox"/> findip |
| <input checked="" type="checkbox"/> iwconfigassociate | <input type="checkbox"/> holdinternet |
| <input type="checkbox"/> metasploit-autopwn | <input type="checkbox"/> iwconfigold |
| <input type="checkbox"/> nmaplocal | <input type="checkbox"/> nessus |
| <input type="checkbox"/> pickupline | <input checked="" type="checkbox"/> nmapplugin |
| <input type="checkbox"/> randommac | <input type="checkbox"/> proxycheck |
| <input type="checkbox"/> testme | <input type="checkbox"/> rogueapcheck |
| <input type="checkbox"/> weplab-bruteforce | <input checked="" type="checkbox"/> texttospeech |

NOTE: All changes will be saved to custom profile

OK

Output Plugin Output(rausb0)

```
[-] Reaped child [9613], scheduling
[-] Forked [9614] to manage [mrl-c
[**] Running plugins for Access Poin
[!] There are no plugins configured
[-] Child managing [mrl-open] with
[-] Child [9614] finished. (wait returned [9614])
[*] Children finished.
[**] Wicrawl run [0] finished, starting next run
Stopping...
Child [9442] dead
Discovery and plugin-engine finished
```

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.



- <http://www.sans.org/rr/whitepapers/wireless/176.php> --
“*How to Avoid Ethical and Legal Issues In Wireless Network Discovery*”

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!

Questions?



</end>



Demo and LiveCD handouts

- Real Live Demo
- LiveCD based on Backtrack

References

- <http://midnightresearch.com>
- <http://midnightresearch.com/projects/wicrawl>

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?