

CHURCH

⊕ F WIFI
TIME TO PLAY

New Wireless Fun From the Church Of WiFi

Who we are

- Founded by Blackwave
- Reformed in 2005 after Blackwave vanished
- An attempt to facilitate collaboration and skills sharing
- Free for anyone to join, but you should contribute
- Trying to bring ideas to light that might otherwise never be acted on

Members Present

- Speaking
 - RenderMan
 - Thorn
 - H1kari
- In the audience
 - Dutch
 - Joshua Wright
 - Skynetos
 - GB3
 - Assorted others, make yourselves known

Project Updates

- CoWF WPA lookup tables
 - Torrent Online and a few archives of individual files
 - 7 gb
 - Anyone with reliable large hosting, please help!
- Evil Bastard
 - Evil bastard finally here (and so is the project)
 - Demo
- Kiswin
 - Depreciated until newcore

New stuff

- We've been busy since Shmoocon and Layerone:
- Look ma' no head!
- Sneaky bastard
- When hardware attacks!
- Bigger, faster better WPA cracking
- Breaking WPA is fun, but WPA2 is more fun

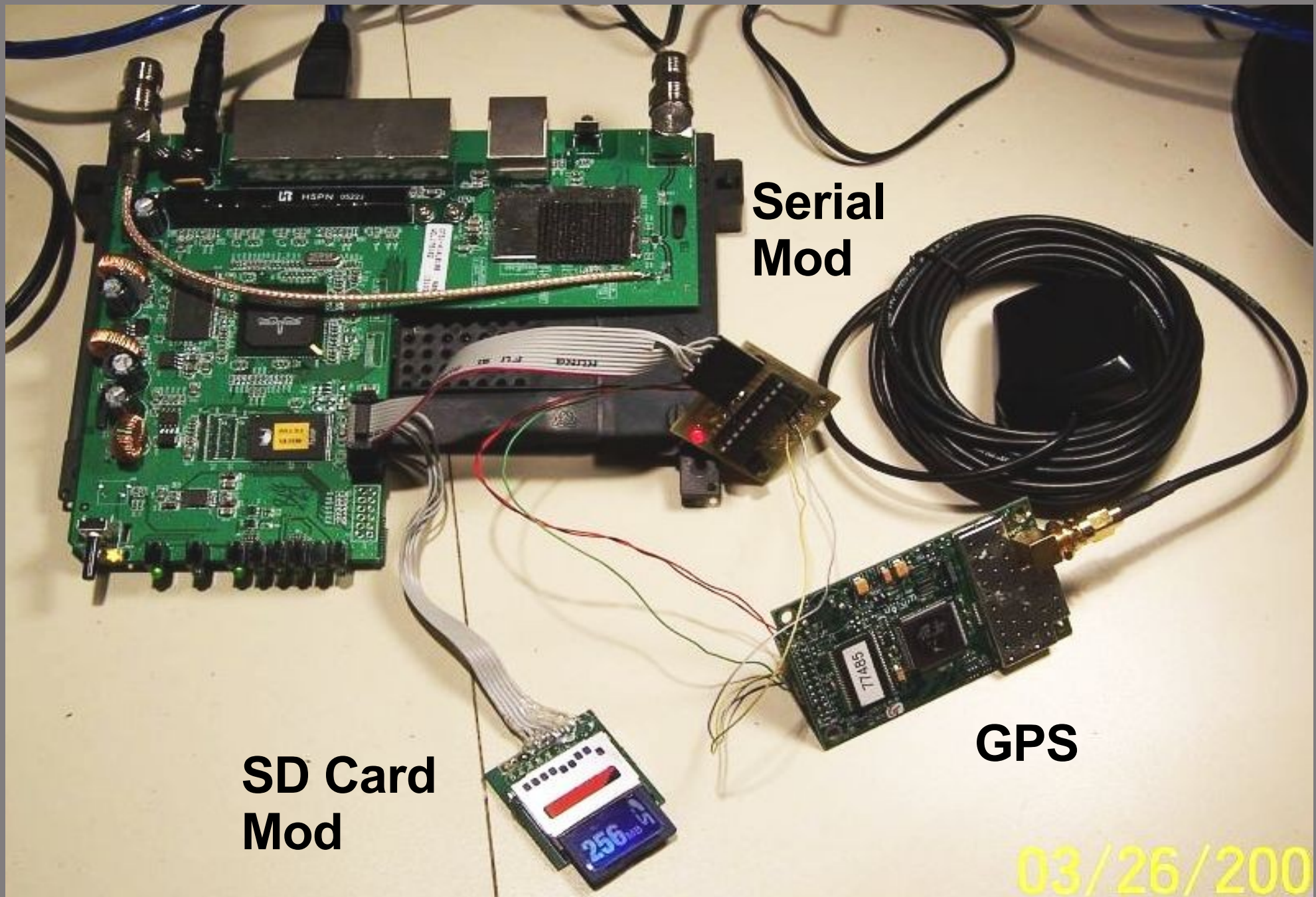
Headless wardriving

- The dream of a small automated wardriving unit
- WRT54G is the perfect platform
- Need GPS, storage, power circuit
- Beakmyn, King_Ice_Flash, Mother, Scrudge and others stepped up
- GPSD ports and most of the hard work is done

Beakmyn's Headless Wardriver

- Integrated GPS internal, SD card
- External active GPS antenna
- Logs saved to removable SD card
- Beakmyn did the hard work (Kismet and GPSPD crosscompiles)
- Rough ipkg available for OpenWRT
- Backup power circuit in the works

Beakmyn's Headless Wardriver



**Serial
Mod**

**SD Card
Mod**

GPS

03/26/200

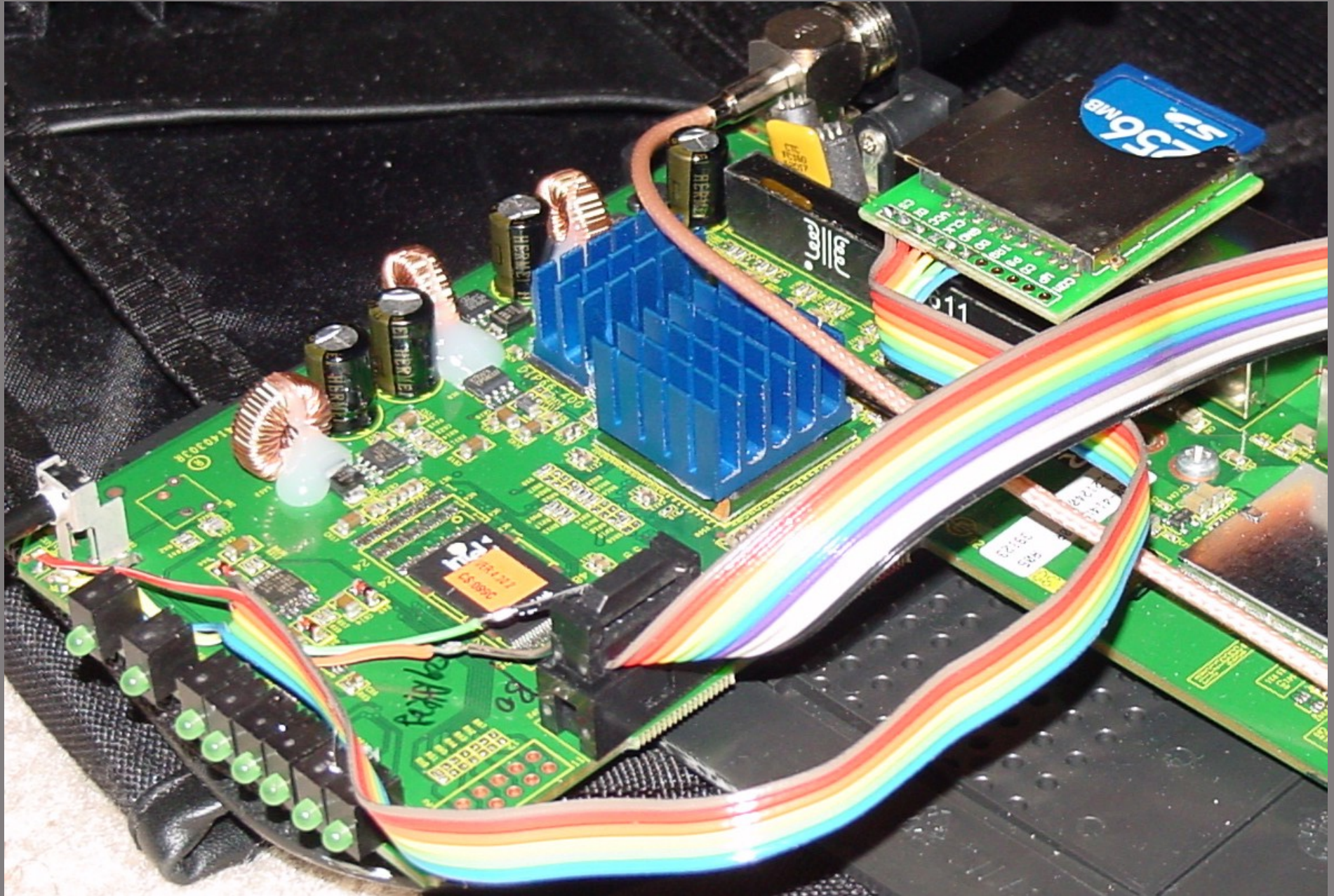
Beakmyn's Headless Wardriver



King_Ice_Flash's Headless Wardriver

- Based off Beakmyn's design with some changes
- Temperature controlled heatsinks and fans
- Sexy laser cut SD card
- Integrated internal GPS

King_Ice_Flash's Headless Wardriver



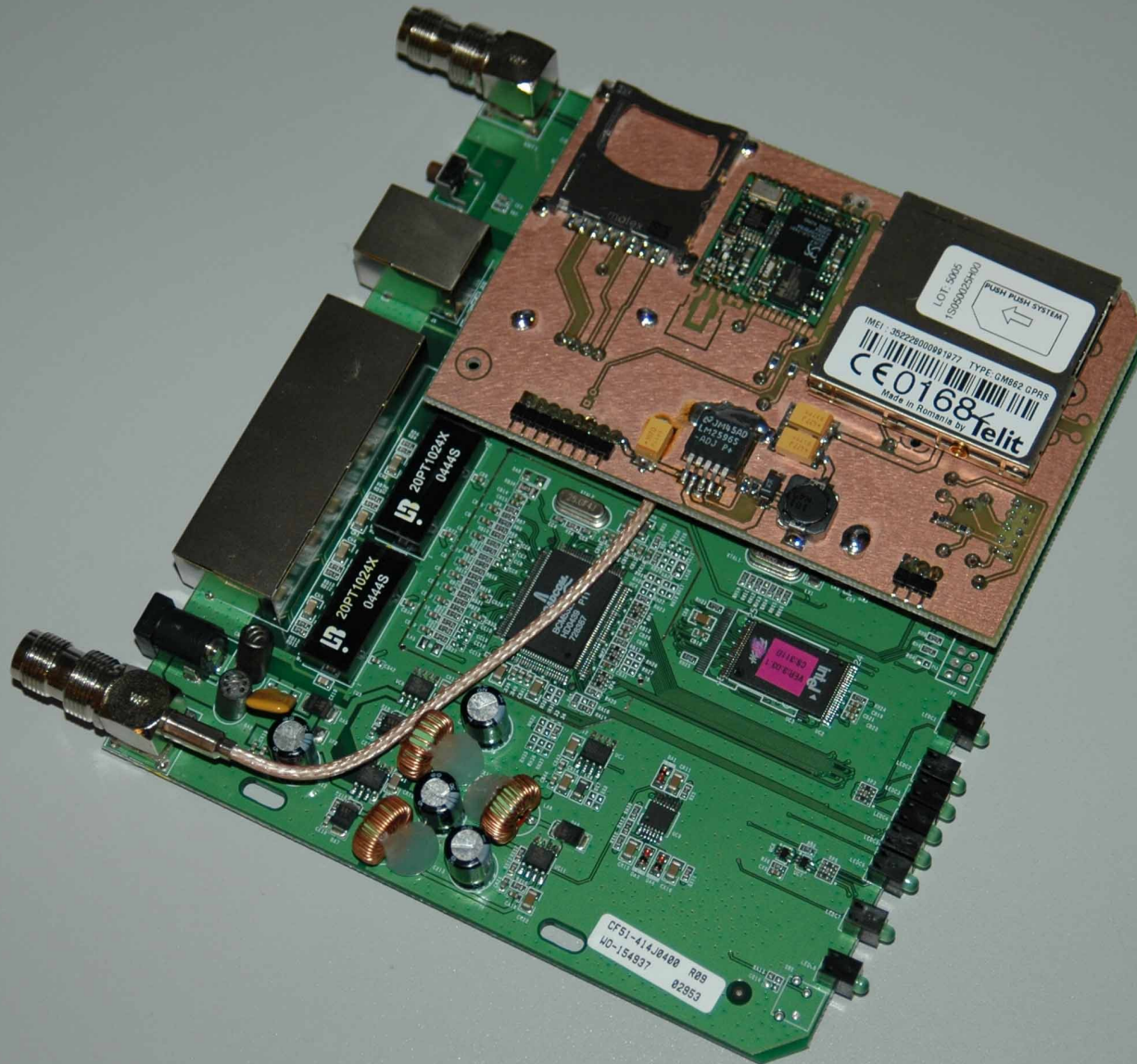
King_Ice_Flash's Headless Wardriver



Mother's Headless Wardriver

- SD card, GPS, GSM/GPRS!
- All integrated on a sexy custom board

Mother's Headless Wardriver



Sneaky Bastard

- Inspired by Inventgeek.com's 'Rogue server' – Hidden storage server in a UPS
- Well, we can do better
- Integrate a rogue access point into a UPS
- Normally plugs into a wall (provides power)
- Normally has network pass through surge protection (network access)
- Needed something less fluffy for pen testing



**Would you suspect
this pair of intruders?**

AP inside Teddy Bear

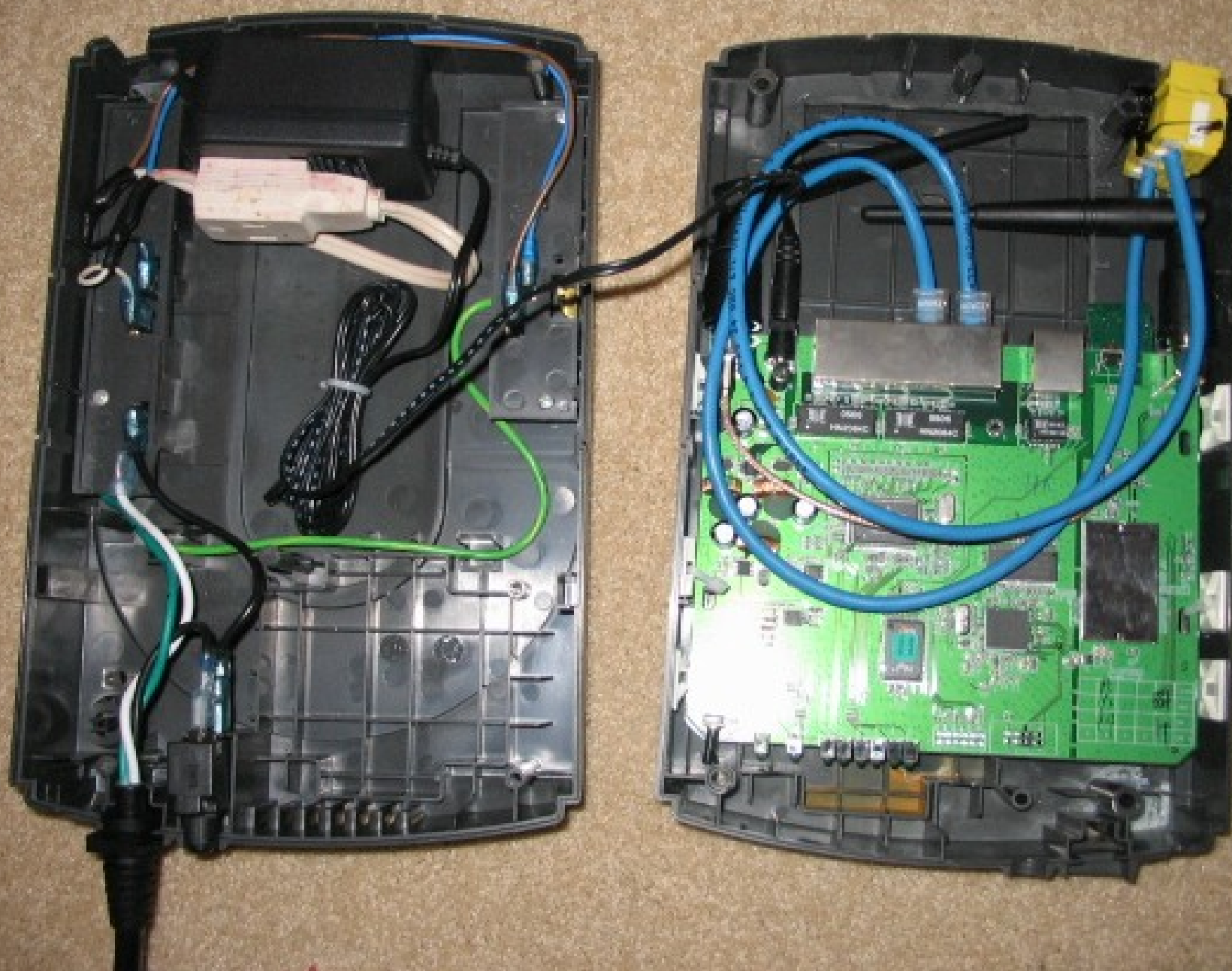


Sneaky bastard

- APC 350 Ups that got wet and fried
- Gutted power circuitry, removed battery case
- Jumpered around the circuitry and powered the sockets (they still work!)
- Spliced in WRT54G wallwart
- Installed patch cables into network surge protector (both hooked to the WRT switch)
- Voila!, Hidden rogue AP
- Load EB firmware for maximum fun

Sneaky Bastard



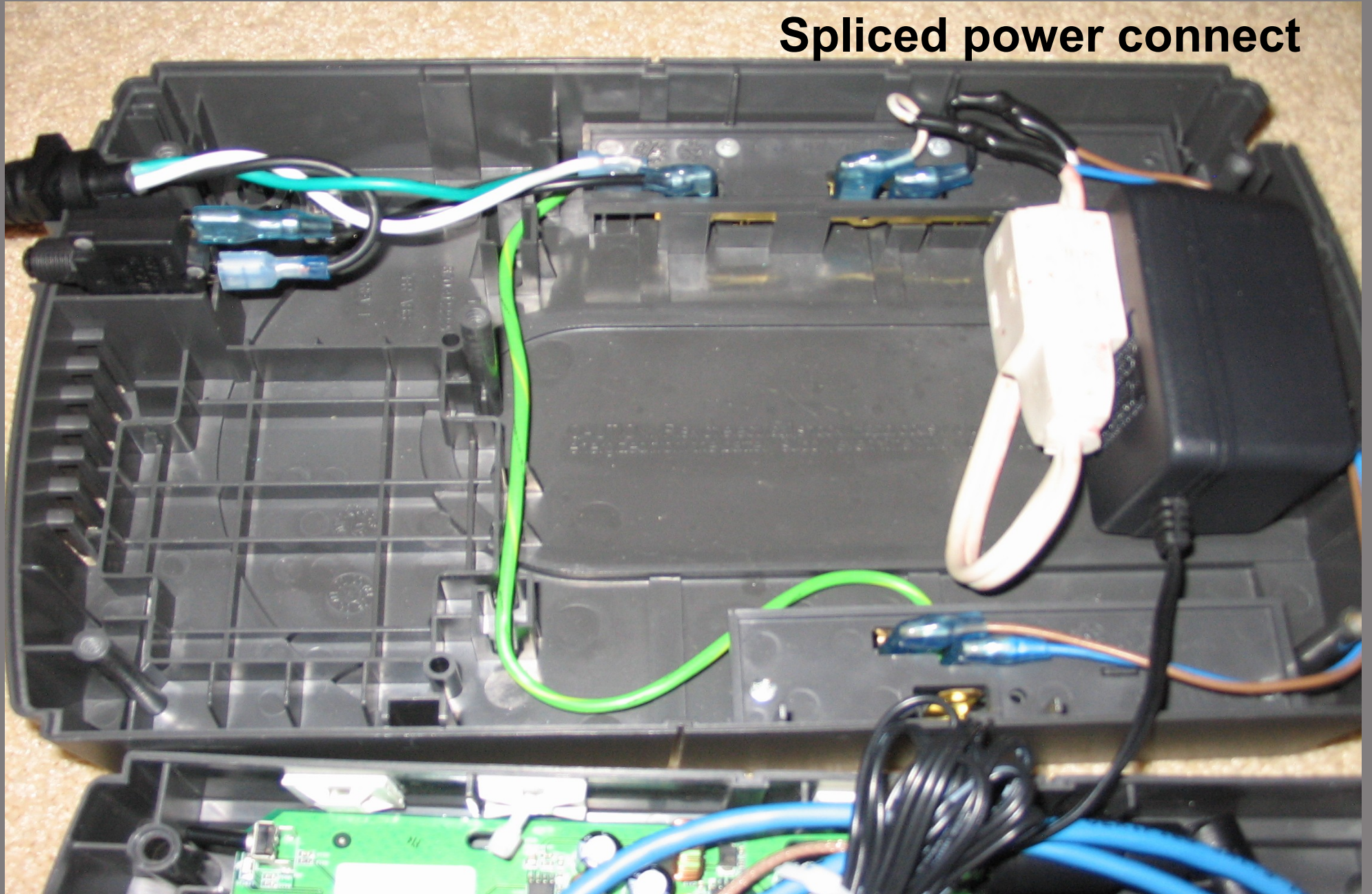




Modded WRT54G

Network Pass Through

Spliced power connect



When hardware turns evil

- Based off ideas from Ciscogate and the Evil Bastard
- If you own the hardware in the middle, you own everything
- How do you know the firmware your running is original
- What about consumers? How do you clean a virus from your router? How do you know it's really a linksys firmware?

Wireless virus concept

- No POC, too dangerous
- Based off of Pre-Set Kill Limit's 'killbot' WRT
- Open source firmware gives intimate knowledge of internals, vulns
- Settings maintained through flash in the Nvram
- Raises many questions about what consumers and admins can do
- No integrity checking for hardware, particularly COTS gear

The Problem

- Default AP's a lot more dangerous
- User has no way to verify the integrity of the running firmware, no anti-virus solution
- Any vulnerability on the WAN side could expose a huge number of AP's to re-flashing/infection/bricking
- Spambots, backdoors, DoS bots, whatever
- Could be used to infect connected hosts
- Evil bastard type actions

Step by step

- Assume a WRT54G(S)(L)
- At 3am, WRT runs cron job and goes into client mode and scans for nearby networks named 'linksys'
- If open, connect and try the admin page
- If no password (or default password), upload copy of the evil firmware (Yes, you can reflash over wireless)
- Unit reboots, maintaining SSID, channel, etc, but running new firmware
- Newly infected unit repeats process, passing the infection along
- AP continues to function normally, user unaware of what firmware is now doing (except for outage at 3am)



Patient 0



Linksys #1



Linksys #2

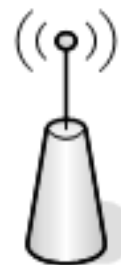


Patient 0
(Client mode)

Scans and Connects
to open linksys AP



Linksys #1



Linksys #2

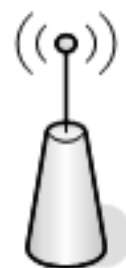


Patient 0
(Client mode)

Checks for admin page
Uploads new firmware



Linksys #1



Linksys #2

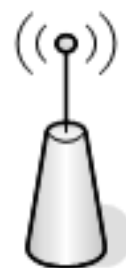


Patient 0
(Client mode)



Linksys #1
(infected)

Process repeats
For other Ap's in range



Linksys #2

Caveats

- Unit must be in default modes
 - Rudimentary brute force for admin page could be made
 - Login banners could trigger unit specific attacks from known exploits
- Not suitable where only one AP in range (campus's on the other hand....)
- Flashing over wireless can brick the router (still not a good thing)
- Only suitable for open firmwares, but there's a lot of AP's using them
- Care could be taken to clone the linksys web admin to appear to the user to be acting normal

Solutions?

- Checksum facility for running firmware?
 - Easily worked around
- Non-flashable hardware
- Proprietary firmware with digital signatures
- TCP profiling for changes
- Suggestions?
- A lot of consumer hardware running open source firmware, problem could grow to switches, modems, etc.

WPA-PSK cracking tables

- Debuted at Shmoocon
- Applying pre-computation attack to WPA-PSK
 - Genpmk util in CoWPAtty 3.0
 - Allows for 3 order of magnitude increase in speed (12 keys/sec to 18,000 keys/sec)
- CoWF lookup tables
 - 1000 top SSID's computed against 172,000 word dictionary
 - 7GB
- Torrent available (finally), thanks Audit, c0n!
- OSX can do it as well (thx beetle)

Bigger, faster, better!

- 7 GB is a good start, but I wanted more
- SkynetOS & GB3 wrangled 14 CPU's for 'testing'
- Mark Burnett provided list of actual used passwords
- Marinate, shake and stir = Million word dictionary
- 3 weeks, 3 blown fuses....

Bigger, Faster, Better

- And the damn tables didn't work!!!
- Always doublecheck your UNIX/DOS text files
- Luckily, H1kari steps up with coWPAtty -f

CoWPAtty -f

- Hardware FPGA implementation of PBKDF2 algorithm
- P4 3.8 Ghz – 69 keys/sec
- FPGA card – 200 keys/sec per card
- We have 15 cards!
- ~3000 keys/sec
- What took 3 weeks took 3 days!

CoWF WPA Uber-lookup tables

- 40 gig
- 1000 top SSID's against a million word dictionary
- Torrent online at www.churchofwifi.org after con
- Please seed!

WPA2 Through Brute Force

- Shmoocon release of coWPAtty 3.0 went well, a lot of conversation afterwards
- Josh Wright mentioned possible WPA2 vulns
- Kept a fire under him to follow the path
- So simple he had to explain it twice

CoWPAtty 4.0

- WPA-PSK and WPA2-PSK share the same PBKDF2 function
- The same problem with WPA1 is present in WPA2
- WPA2 might use stronger crypto, but only as strong as the weak link
- CoWPAtty 4.0 supports WPA2-PSK
- Demo



Filter: eapol.keydes.key_info.keydes_ver == 2 Expression... Clear Apply

No. *	Time	Source	Destination	Protocol	Info
50	1.105551	ArubaNet_c2:a4:85	IntelCor_55:98:ef	EAPOL	Key
51	1.113587	IntelCor_55:98:ef	ArubaNet_c2:a4:85	EAPOL	Key
53	1.116723	ArubaNet_c2:a4:85	IntelCor_55:98:ef	EAPOL	Key
54	1.121658	IntelCor_55:98:ef	ArubaNet_c2:a4:85	EAPOL	Key
89	1.899442	ArubaNet_c2:a4:85	IntelCor_55:98:ef	EAPOL	Key
90	1.902858	IntelCor_55:98:ef	ArubaNet_c2:a4:85	EAPOL	Key
92	1.906676	ArubaNet_c2:a4:85	IntelCor_55:98:ef	EAPOL	Key
93	1.909786	IntelCor_55:98:ef	ArubaNet_c2:a4:85	EAPOL	Key
339	7.134701	ArubaNet_c2:a4:85	IntelCor_55:98:ef	EAPOL	Key

Frame 50 (153 bytes on wire, 153 bytes captured)

IEEE 802.11

Logical-Link Control

802.1X Authentication

Version: 1

Type: Key (3)

Length: 117

Descriptor Type: EAPOL RSN key (2)

Key Information: 0x008a

.... .010 = Key Descriptor Version: AES-CBC-MAC for MIC and HMAC-SHA1 for encryption (2)

.... 1... = Key Type: Pairwise key

.... ..00 = Key Index: 0

0 = Install flag: Not set

192.168.0.52 - PuTTY

```
render@Darla:~/cowpatty-3.1beta1$ ./cowpatty -d linksys.hash -r wpa2psk-
```

```
linksys.dump -s linksys
```

```
cowpatty 3.1beta1 - WPA-PSK dictionary attack. <jwright@hasborg.com>
```

```
Collected all necessary data to mount crack against WPA2/PSK passphrase.
```

```
Starting dictionary attack. Please be patient.
```

```
key no. 10000: arrojadite
```

```
key no. 20000: calligraphical
```

```
key no. 30000: contestation
```

```
The PSK is "dictionary".
```

```
38333 passphrases tested in 1.34 seconds: 28532.32 passphrases/second
```

```
render@Darla:~/cowpatty-3.1beta1$
```


BTW...

- Since WPA1-PSK and WPA2-PSK share the same key hash, the lookup tables for WPA1 are compatible with WPA2
- 47 gig of tables all ready for WPA2
- Torrents ready to go!

Future projects

- Integrated WoS (Wall of sheep) appliance
- WiFi grenade
- Bigger tables (please help with hosting)
- Special targeted tables (4.7GB linksys table anyone?)

Links

- <http://www.churchofwifi.org>
- <http://www.inventgeek.com/Projects/projectsilver/Overview.aspx>
- <http://www.frontiernet.net/~beakmyn/OpenWRT%20Kimset%20Server.htm>
- <http://www.renderlab.net/projects/sneaky/>