

Wi-Fi Advanced Stealth

Black Hat US, Las Vegas – August 2-3, 2006

Laurent BUTTI & Franck VEYSSET France Telecom Division R&D

firstname dot lastname AT orange-ft dot com





Who Are We?

- Network security experts in R&D labs
 - Working for France Telecom Orange (major telco)
- Speakers at security-focused conferences
 - ShmooCon, ToorCon, FIRST, Eurosec...
- Wi-Fi security focused speakers ;-)
 - "Wi-Fi Security: What's Next" ToorCon 2003
 - "Design and Implementation of a Wireless IDS" ToorCon 2004 and ShmooCon 2005
 - "Wi-Fi Trickery, or How To Secure (?), Break (??) and Have Fun With Wi-Fi" ShmooCon 2006





Beginning of 2006...

- We released 3 new tools at ShmooCon 2006
 - Raw Fake AP: an enhanced Fake AP tool using RAW injection for increased effectiveness
 - Raw Glue AP: a Virtual AP catching every client in a virtual quarantine area
 - Raw Covert: a tricky 802.11 covert channel using valid ACK frames
- All this stuff is available at
 - http://rfakeap.tuxfamily.org





Now at BlackHat US...

- We will release
 - Tricks to "hide" access points and stations (madwifi patches)
 - From scanners and wireless IDS
 - Raw Covert v2: new implementation and features
- We will also introduce our new ideas of research
 - 802.11 fuzzing



Wi-Fi Stealth Tricks







802.11 Havoc!

- Since a couple of years, some wireless drivers are much more "flexible" than Prism2/2.5/3 based...
 - Full RAW injection capabilities (possible to modify some critical fields like fragmentation, sequence number, BSS Timestamp...)
 - Demonstrated by Raw Fake AP, Raw Glue AP and Raw Covert
 - Tweaking the driver may also become attractive!
- Such drivers are
 - Madwifi-{old|ng} for Atheros chipsets
 - Prism54.org for Prism54 chipsets
 - Realtek...
- New capabilities implies new risks to address…
 - Especially for Wireless IDS vendors





Two Ways To Achieve Stealth...

- Possibilities are somewhat infinite...
 - We decided to show only two ways that can be extended
- Tweaks in 802.11 drivers to implement a new "proprietary" protocol over 802.11 bands
 - Madwifi patches
- Covert channel using 802.11 valid frames
 - Raw Covert (as a proof-of-concept)



Hiding Ourselves

Black Hat Briefings

orange"



A Quick Reminder

- IEEE 802.11 standards define what 802.11 is
 - At PHY and MAC layers
 - Modulation, frequencies...
 - State machine, frame fields...
 - Security mechanisms
- To be Wi-Fi compliant, every implementation must comply with the 802.11 standard and be certified by the Wi-Fi Alliance certification process
 - Usual stuff if you want to interoperate...





Main Idea

- What would happen if you implement your own 802.11 stack?!
 - Stations that probe for AP will (probably) not see you...
 - Wireless sniffers will (probably) not understand you, requiring manual inspection...
 - Wireless IDS will (probably) not see you...
- Quite stealthy, no?
- What about your own (undetectable) personal AP?
 - Sure the CSO won't appreciate ©
 - Sure wardrivers won't appreciate either (until now...)



Implementation

- Successfully tested on Atheros chipsets with a patched madwifi-ng driver
 - Patched stations and access points will be able to see and associate themselves (they speak the same language)
 - But non patched stations will not see patched access points, and thus cannot associate to them
- Test bed
 - Windows XP supplicant and NetStumbler
 - Wireless Tools (iwlist) with
 - hostap, (non patched) madwifi-ng, ipw2100, prism54





Live Demonstration

- First, we set up a "special" Access Point
 - one laptop with a patched madwifi-ng in master mode
- Then we scan for this AP with unpatched madwifi-ng
 - iwlist (active scan facilities under *nix)
 - Kismet (passive scanner under *nix)
 - Netsumbler (active scanner under Windows)
- Then, we use our "special" client (patched drivers)
 - Tada... it works...





Design Details

802	11	MAC	header
002			ncauci

	Frame Control	Duration ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Network Data	FCS
ŀ									

2 Bytes | 2 Bytes | 6 Bytes | 0 to 2312 Bytes | 4 Bytes

Protocol Version	Туре	Subtype	To DS	From DS	More Frag	Retry	Power Mgmt	More Data	WEP	Order
2 bits	2 bits	4 bits	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit



WTF Is This? Trivial Tweaks!

- What about changing FC field? ;-)
- What about a protocol version of 1? ;-)
 - 802.11 is protocol version 0
- What about swapping types?
 - Management (value 0)
 - Control (value 1)
 - Data (value 2)
 - Reserved (value 3)
- What about swapping subtypes?
 - Is this a Probe Request or a Probe Response? ;-)





Not So Trivial Tweaks

- Everything is possible... Make your own MAC protocol
- SoftMAC: A Flexible Wireless Research Platform
 - http://systems.cs.colorado.edu/projects/softmac



Raw Covert



orange™



Raw Covert (1/4)

- Covert channel
 - In information theory, a covert channel is a communications channel that does a writing-between-the-lines form of communication.
 - Source: Wikipedia, the free encyclopedia
- Writing between-the-lines
 - Use valid frames to carry additional information
 - Valid frames could be management, control or data frames
- This tool is 'only' an example! Possibilities are infinite!





Raw Covert (2/4)

- With 802.11, this may be performed by many means
 - Using a proprietary protocol within valid or invalid frames
 - It gives infinite possibilities thanks to RAW injection
- (Some) 802.11 frames are not considered as 'malicious'
 - Control frames like ACK are lightweight and non suspicious!
 - Frame control (16 bits)
 - Duration Field (16 bits)
 - Receiver Address (48 bits)
 - (Usually) not analyzed by wireless IDS
 - No source nor BSSID addresses ;-)
- (Some) 802.11 drivers do not give back ACK frames in monitor mode (operated in the firmware: e.g. HostAP)
 - Increasing stealthyness





Raw Covert (3/4)

- How it works?
 - A client encodes the information and sends ACKs over the air
 - A server listens for ACKs and tries to decode the information
- Basically, it uses a magic number in receiver address
 - 2 bytes
- Basically, it encodes the covert channel in receiver address
 - E.g. 4 bytes
 - Several ACK frames are needed to send information

orange



Raw Covert (4/4)

- Issues
 - ACK frames can be missed, wireless is not a reliable medium! ;-)
 - Detection may be performed (only) with anomaly detection
- Enhancements
 - Basic encryption technique
 - Basic remote shell and file transfer
- Possible enhancements for the covert channel
 - Using invalid frames
 - Using Information Elements in 802.11 frames (but could be easily detected)
 - Using existing communications (clients and access points)



Raw Covert Enhancements (1/2)

- Invalid frames (in the 802.11 sense, i.e. proprietary frames)
 - But would (?) be detected by any wireless IDS performing sanity check on every frame
- FCS invalid frames
 - Should require driver/firmware modifications to inject bad FCS
 - Wireless IDSs do not analyze such bad frames
 - But should be detected with FCSerr statistics (even if harder to diagnose as a covert channel)





Raw Covert Enhancements (2/2)

- Invalid FCS monitoring
 - Usually a bit is set by the firmware when a FCS is invalid
 - Most drivers discard packets with bad FCS thanks to this information
 - HAL_RXERR_CRC for madwifi
 - rfmon_header->flags & 0x01 for prism54
 - HostAP driver has a facility
 - prism2_param interface monitor_allow_fcserr 1





Live Demonstration

Live demo!

Did you detected it? ;-)



orange

802.11 Fuzzing







Fuzzing Concepts (1/2)

- Fuzzing
 - Fuzz testing is a software testing technique. The basic idea is to attach the inputs of a program to a source of random data. If the program fails (for example, by crashing, or by failing built-in code assertions), then there are defects to correct.
 - From Wikipedia, the free encyclopedia





Fuzzing Concepts (2/2)

- Fuzzing is not something really new...
 - Remember ISIC?
 - http://www.packetfactory.net/projects/ISIC/
- But it is still of interest...
 - Recent work on Bluetooth Fuzzing (Pierre Betouin)
 - http://www.secuobs.com/bss-0.6.tar.gz
 - Fuzzing with Scapy... (Phil Biondi)
 - Plenty of cool things to do with scapy...





Fuzzing 802.11

- IEEE 802.11 amendments are more and more numerous
 - 802.11e, 802.11i, 802.11k, 802.11r, 802.11s, 802.11w...
- Axiom
 - Complexity → more code → more bugs → more vulnerabilities
- Guess what? IEEE 802.11 may be susceptible to fuzzing!





Fuzzing 802.11

- Not so trivial... keep in mind the 802.11 state machine
- Each step of the 802.11 protocol may be fuzzed
 - Scanning process: probe requests and responses, beacons
 - Authentication process: authentication requests and responses
 - (Re-)Association process: (re-)association requests and responses
- Station's associated state can be fuzzed only if
 - Station is in state « Authenticated, Not Associated »
 - (Optionally) There was an (re-)association request sent by the station to the access point were he was previously authenticated





Fuzzing 802.11

- Easiest part: fuzzing clients thanks to probe responses and beacons
 - Listen for probe requests and send back appropriate probe response
- Fuzzing probe responses and beacons
 - Inconsistent Information Elements (Type Length Value)
 - E.g. a SSID Information Element with a length above 32 bytes
 - E.g. a short 802.11 frame (incomplete SSID IE)
 - Incomplete frame length...
 - More on this soon...





Tools, patches available at

http://rfakeap.tuxfamily.org







References

- Laurent Oudot's wknock
 - http://www.rstack.org/oudot/wknock/
- Pierre Betouin's Bluetooth Stack Smasher
 - http://www.secuobs.com/bss-0.6.tar.gz
- scapy (Phil Biondi)
 - http://www.secdev.org
- SoftMAC: A Flexible Wireless Research Platform
 - http://systems.cs.colorado.edu/projects/softmac
- MadWiFi patches and rawcovert
 - http://rfakeap.tuxfamily.org