1. Public WiFi networks
2. Messing with network
   - Attacking hotspots
   - Attacking Mesh networks
3. Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass
4. Conclusion
5. References
   - Demos
   - Bibliography
Agenda

1. Public WiFi networks
2. Messing with network
   - Attacking hotspots
   - Attacking Mesh networks
3. Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass
4. Conclusion
5. References
   - Demos
   - Bibliography
Public WiFi networks

Open wireless networks
- Anybody can access network
- Zero conf. or so
- Services open to anyone

Anybody can access/play/attack...
Open wireless networks

- Anybody can access network
- Zero conf. or so
- Services open to anyone

Anybody can access/play/attack...
Security specificities

Security?
- No authentication/authorization
- No message authenticity
- No confidentiality
- Etc.

Some clients isolation measures
Open networks

Open infrastructure network: anyone can join
- Generous users who share their access
- Any traffic allowed to Internet
- Sometimes some restrictions (ports, bandwidth)

Legal issues?
Open infrastructure network: anyone can join

- Outbound traffic is filtered out
- HTTP traffic is redirected to auth. portal
- Once registered, client can access Internet
Mesh networks

Adhoc based network
  • Clients to clients links
  • Clients can join/move/leave
  • Dynamic and adaptative routing
AODV, OLSR provide dynamic and adaptative routing
Agenda

1. Public WiFi networks

2. Messing with network
   - Attacking hotspots
   - Attacking Mesh networks

3. Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass

4. Conclusion

5. References
   - Demos
   - Bibliography
1 Public WiFi networks

2 Messing with network
   - Attacking hotspots
   - Attacking Mesh networks

3 Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass

4 Conclusion

5 References
   - Demos
   - Bibliography
Rogue AP
Attack principles

Classical, unexpensive, well known layer 1/2 attack
- Set up AP with same configuration
- Power-up and associate clients
- Divert client traffic and play

Easy, efficient, powerful tools available [KRM]
Rogue AP  
Application to hotspots

Take advantage from traffic redirection

- Credentials interception
- Crypto MiM attack
- Assisted registration

Not very practical if not gifted with impressive 6th sense

\(^a\)Who cares about that f**kin’ popup anyway?
Tracking authenticated clients

Captive portal can only rely on network addresses for clients identification

- MAC address
- IP address

Being able to spoof those addresses allows existing authorization takeover
MAC based authorization tracking

Registered clients are identified by their MAC address
- MAC address is easy to spoof
- No MAC layer conflict on WiFi network
- Just need a different IP
MAC based tracking practical bypass

Change WiFi interface MAC address

MAC spoofing

```
joker# ifconfig wlan0 hw ether $MAC
joker# ifconfig wlan0 $IP $NETMASK $BROADCAST
joker# route add default $FIREWALL
```
IP based authorization tracking

Registered clients are identified by their IP address

- IP address are just a little more tricky to spoof
- ARP cache poisoning helps redirecting traffic
- Traffic redirection allows IP spoofing

So called Smart Spoofing\(^a\)

\(^a\)See my LSM 2002 talk[BLA02], arp-sk website[ARPS] for details
"Smart spoofing"

**IP spoofing**

```bash
joker# echo 1 > /proc/sys/net/ipv4/ip_forward
joker# arp-sk -i ath0 -w -d $FIREWALL -S $BATMAN \
       -D $FIREWALL
joker# iptables -t nat -A OUTPUT -d ! $LAN \
       -j SNAT --to $BATMAN
joker# iptables -t mangle -A FORWARD -d $BATMAN \
       -j TTL --ttl-inc 1
```
MAC+IP addresses based authorization tracking

Teh-Smart tracking technic?

- Previous technic won’t help because of MAC address checking
- Send traffic with spoofed MAC address
- ARP cache poisoning and IP spoofing for answers redirection
Why does MAC+IP does not help either?

Layer2 and Layer3 are close to independant

- No correlation between ARP cache and filtering
- Joker’s MAC spoofed frames are accepted
- Returning frames are sent with Joker’s MAC address
MAC+IP tracking bypass

Joker uses ebtables[EBT] to have output frames spoofed

MAC+IP spoofing

joker# modprobe bridge
joker# brctl addbr br0; brctl addif br0 ath0
[configure bridge interface br0]
joker# ebtables -t nat -A POSTROUTING -o ath0 \ 
   -d $FW_MAC -j snat \ 
   --to-source $BATMAN_MAC

Then IP spoofing can be done, performing ”Smarter spoofing” :)

Philippe TEUWENCédric BLANCHER

802.11 Security
Demo

- Captive portal bypass
- MAC+IP spoofing
Few other technics

- Misconfigurations
- DNS based communication[OZY] or tunneling[NSTX]
- Administration network on the same VLAN, accessible through WiFi
- ESTABLISHED,RELATED -j ACCEPT prevents connections drop when authorization expires on Linux based systems
- Etc.
1 Public WiFi networks

2 Messing with network
   • Attacking hotspots
   • Attacking Mesh networks

3 Leurring clients
   • Traffic tampering
   • Bidirectional station isolation bypass

4 Conclusion

5 References
   • Demos
   • Bibliography
Dynamic routing

Mesh networks relies on dynamic routing
- Neighbourhood discovery
- Network announces
- Link table
- Routing table

Lots of networks use OLSR
Dynamic routing abuse

No authentication/integrity measure

- Anybody can announce anything

Scenario

- Use a powerful antenna
- Announce Internet connectivity
- Gather traffic from part of network
- Play with connections
Dynamic routing abuse

No authentication/integrity measure
- Anybody can announce anything

Scenario
- Use a powerful antenna
- Announce Internet connectivity
- Gather traffic from part of network
- Play with connections
Multipoint route injection

One can just inject OLSR messages
- without being part of network
- to multiple links

Route injection
- Includes neighbourhood
- Becomes more consistent
- Stays more stealth

Can use arbitrary messages using Wifitap
Agenda

1. Public WiFi networks
2. Messing with network
   - Attacking hotspots
   - Attacking Mesh networks
3. Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass
4. Conclusion
5. References
   - Demos
   - Bibliography
All known "LAN attacks" are available

- LAN attacks (ARP, DHCP, DNS, etc.)
- Traffic interception and tampering
- Direct station attacks

Think of infamous personal firewalls exception for local network or loose firewall settings...
1 Public WiFi networks

2 Messing with network
   - Attacking hotspots
   - Attacking Mesh networks

3 Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass

4 Conclusion

5 References
   - Demos
   - Bibliography
Traffic tampering

WiFi communication can be listened on the air

- Listen to WiFi traffic
- Spot interesting requests
- Inject your own crafted answers
- You’ve done airpwn-like[AIRP] tool

Applications: ARP spoofing, DNS spoofing, malicious data injection, etc.
Demo

- DNS Spoofing
- Ping answering machine
1 Public WiFi networks

2 Messing with network
   - Attacking hotspots
   - Attacking Mesh networks

3 Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass

4 Conclusion

5 References
   - Demos
   - Bibliography
Stations isolation

Security feature that blocks traffic within BSS

- Station sends To-DS frame
- AP sees destination is in BSS
- AP drops the frame

No From-DS frame, so no communication: stations can’t talk to each other...

\[ a \] Does not work between 2 APs linked via wired network
Joker can inject From-DS frames directly
- No need for AP approval
- You can spoof about anyone
- You’re still able to sniff traffic

Traffic injection allows complete isolation bypass
Bidirectional communication with injection
Sending packets the ninja way

Sending traffic directly to stations allows direct station to station communication, even if:

- AP applies restrictions
- AP refuses association
- AP is out of reach

Talking to stations the ninja way, without being associated
Attacking stations
Proof of concept: Wifitap

Needed a PoC for Cisco PSPF bypass and wrote Wifitap
  - Written in Python
  - Relies on Scapy
  - Uses tuntap device and OS IP stack
  - Use WiFi frame injection and sniffing

Wifitap allows communication with station despite of AP restrictions
Wifitap works for mesh networks as well
Wifitap usage

```
# ./wifitap.py -h
Usage: wifitap -b <BSSID> [-o <iface>] [-i <iface> [-p]]
[-s <SMAC>] [-w <WEP key> [-k <key id>]]
[-d [-v]] [-h]

-b <BSSID> specify BSSID for injection
-o <iface> specify interface for injection
-i <iface> specify interface for listening
-s <SMAC> specify source MAC address
-w <key> WEP mode and key
-k <key id> WEP key id
-d activate debug
-v verbose debugging
-h this so helpful output
```
Wifitap in short

How does it work?

**Sending traffic**
- Read ethernet from tuntap
- Add 802.11 headers
- Set BSSID, From-DS and WEP if needed
- Inject frame over WiFi

**Receiving traffic**
- Sniff 802.11 frame
- Remove WEP if needed and 802.11
- Build ethernet frame
- Send frame through tuntap

Attacker does not need to be associated (AP or Adhoc)
Wifitap in short

How does it work?

**Sending traffic**
- Read ethernet from tuntap
- Add 802.11 headers
- Set BSSID, From-DS and WEP if needed
- Inject frame over WiFi

**Receiving traffic**
- Sniff 802.11 frame
- Remove WEP if needed and 802.11
- Build ethernet frame
- Send frame through tuntap

Attacker does not need to be associated (AP or Adhoc)
How does it work?

**Sending traffic**
- Read ethernet from tuntap
- Add 802.11 headers
- Set BSSID, From-DS and WEP if needed
- Inject frame over WiFi

**Receiving traffic**
- Sniff 802.11 frame
- Remove WEP if needed and 802.11
- Build ethernet frame
- Send frame through tuntap

Attacker does not need to be associated (AP or Adhoc)
Demo

- Wifitap in action
Hotspots with isolation

Some hotspots implement isolation to prevent clients from attacking each other

- Does not protect against "session" hijacking
- Attacker must then to take over victim’s session
- Victim does not have access anymore, and still pays for it

And among all, it’s pretty useless...
More hotspot bypassing...

Hijacking people authorization is not very kind

- Use Wifitap to bypass isolation
- Now you can route back his traffic to your victim

Your victim and you are both able to surf transparently

Now, you "can be a true gentlemanly [h|cr]acker" [ISCD] ;)

Philippe TEUWEN
Cédric BLANCHER
802.11 Security
1. Public WiFi networks
2. Messing with network
   - Attacking hotspots
   - Attacking Mesh networks
3. Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass
4. Conclusion
5. References
   - Demos
   - Bibliography
So you thought dropping the wire was that easy?

- No privacy, no integrity
- Public accesses are just so insecure
- Crackers do know about that
Conclusion

What do we do to fix that?

Clients

- Open network services can’t be trusted
- Open network traffic neither
- Think authentication, encryption, VPN

Don’t forget to tunnel DNS as well :)

Infrastructure

- Considering WEP ? Forget it !
- Consider real stuff : WPA/WPA2 w/EAP
- Now supported on most devices/OS
## Conclusion

What do we do to fix that?

### Clients
- Open network services can’t be trusted
- Open network traffic neither
- Think authentication, encryption, VPN

Don’t forget to tunnel DNS as well :)

### Infrastructure
- Considering WEP ? Forget it !
- Consider real stuff : WPA/WPA2 w/EAP
- Now supported on most devices/OS
## Conclusion

What do we do to fix that?

### Clients
- Open network services can’t be trusted
- Open network traffic neither
- Think authentication, encryption, VPN

Don’t forget to tunnel DNS as well :)

### Infrastructure
- Considering WEP ? Forget it !
- Consider real stuff : WPA/WPA2 w/EAP
- Now supported on most devices/OS
Thank you for your attention and...

Greetings to...

- **BCS Asia 2006** people, partners and sponsors
- **EADS CRC/DCR/STI/C team**
- **Rstack.org** team
  http://www.rstack.org/
- **MISC Magazine**
  http://www.miscmag.com/
- **French Honeynet Project**
  http://www.frenchhoneynet.org/

Download theses slides from http://sid.rstack.org/
Agenda

1. Public WiFi networks
2. Messing with network
   - Attacking hotspots
   - Attacking Mesh networks
3. Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass
4. Conclusion
5. References
   - Demos
   - Bibliography
1 Public WiFi networks

2 Messing with network
   ● Attacking hotspots
   ● Attacking Mesh networks

3 Leurring clients
   ● Traffic tampering
   ● Bidirectional station isolation bypass

4 Conclusion

5 References
   ● Demos
   ● Bibliography
Demos

- Captive portal bypass
- Traffic tampering
- Bidirectional isolation bypass
1 Public WiFi networks

2 Messing with network
   - Attacking hotspots
   - Attacking Mesh networks

3 Leurring clients
   - Traffic tampering
   - Bidirectional station isolation bypass

4 Conclusion

5 References
   - Demos
   - Bibliography

[BLA02] C. Blancher, Switched environments security, a fairy tale, 2002,
http://sid.rstack.org/pres/0207_LSM02_ARP.pdf

http://sid.rstack.org/pres/0307_LSM03_L2_Filter.pdf

[BLA06] C. Blancher, WiFi traffic injection based attacks, 2005-2006
http://sid.rstack.org/pres/0602_Securecon_WirelessInject
Bibliography II


[OZY] OzymanDNS,
http://www.doxpara.com/ozymandns_src_0.1.tgz


Bibliography III

[WTAP] Wifitap,  
http://sid.rstack.org/index.php/Wifitap_EN

[ISCD] ISC Handler’s Diary,  