ARProtokoll Schwächen

Felix Gröbert 2004 - Version 1.0
http://groebert.org/felix/

Vortrags Struktur

- Protokoll
- Angriffstechniken
- Angriffsszenarios
- Programme
- Resultate
- Demo

Motivation?

- 70% der Angriffe von ‘innen’
- jeder 500ste ‘hasst’ seine Firma
- ARP Schwächen oft unterschätzt

Address Resolution Protocol

- Protokoll zum Zuordnen von Hardware- zu IP Adressen

RFC

- ARP ist definiert im RFC 826 von 1982
- Generieren von Protokoll Adressen aus der Hardware Adresse
- Protokolle wie DOD TCP, Xerox BSP, DECnet

mehr RFC

- RFC 826 wird durch RFC 903, 1293, 1735, 2390 ergänzt (Inverse / Reverse ARP)
- Zeroconf - in Arbeit bei IETF

Einordnung OSI-Model

Transport	UDP, TCP
Network	IP, ICMP, ARP
Data Link	Ethernet, Token Ring, PPP, ATM
Physical	RJ45, 10BASE-T, RS-232

meine Einordnung

Transport	UDP, TCP	
Network	IP, ICMP	Layer 3
ARP		Layer 2,5
Data Link	Ethernet	Layer 2

Wie gehts?

- 192.168.23.3 möchte via IP mit 192.168.23.1 sprechen, Gesucht: MAC
- arp who-has 192.168.23.1 tell 192.168.23.3
- arp reply 192.168.23.1 is-at 00:0f:66:d3:fc:0n
- 192.168.23.3 weiss jetzt die MAC von 192.168.23.1

MAC (1)

- Media Access Control
- Ziel: eindeutige Identifizierung eines Netzwerkdevice
- 48 Bit = 6 Bytes = 6 hexadezimale Zahlen
- Erste 3 Bytes Herstellerkennung
- ac:de:48 ist der Hersteller “Privat”

MAC (2)

- Herstellerpräfixe im Netz können Rückschlüsse geben
- ifconfig eth0 ether 01:01:ac:1d:b4:be
- Broadcast ff:ff:ff:ff:ff:ff
- Multicast 01:00:5e:xx:xx:xx

ARP cache

- MAC-IP Zuordnungen müssen gecacht werden
- cache muss sich aktuell halten (timers)
- requests-src Daten und reply Daten kommen in den Cache

Ethernet Paket

Preamble		8 Bytes
Destination MAC	00:ff:56:a0:00:12	6
Source MAC	00:80:de:ad:b3:3f	6
Ether Type	08:00 | 08:06 | 86:dd	2
Data	IPv4 | ARP | IPv6	46-1500
FCS	CRC	4

ARP Paket Aufbau

Hardware Typ	Ethernet 00:01	2 Bytes
Protokoll Typ	IP 08:00	2
Hardware Länge (x)	MAC = 0x06	1
Protokol Länge (y)	IP Adresse = 0x04	1
Opcode	request 00:01 oder reply 00:02	2
Sender MAC	00:23:42:aa:bb:cc	x
Sender IP	192.168.23.3	y
Target MAC	00:00:00:00:00:00	x
Target IP	192.168.23.1	y

Paketbündelparameter

- Paket Parameter:
  - Opcode (request / reply)
  - Ethernet DST
  - Ethernet SRC
  - ARP SRC MAC
  - ARP SRC IP
  - ARP DST MAC
  - ARP DSP IP

Standard request

			request
Ethernet	source	MAC	00:23:42:aa:bb:cc
ARP	source	MAC	00:23:42:aa:bb:cc
ARP	source	IP	192.168.23.3
Ethernet	destination	MAC	ff:ff:ff:ff:ff:ff
ARP	destination	MAC	00:00:00:00:00:00
ARP	destination	IP	192.168.23.1

Standard reply

			reply
Ethernet	source	MAC	00:0f:66:0c:15:c0
ARP	source	MAC	00:0f:66:0c:15:c0
ARP	source	IP	192.168.23.1
Ethernet	destination	MAC	00:23:42:aa:bb:cc
ARP	destination	MAC	00:23:42:aa:bb:cc
ARP	destination	IP	192.168.23.3

bessere switche

- switch = besseres hub
- VLANs (viel segmentierung)
- switchport protected (Cisco)
- dynamic ARP inspection (Catalyst)

Angriffstechniken

- MAC spoofing
- ARP spoofing
- ARP cache poisoning

MAC spoofing

Switch Port	MAC	User
1	00:03:11:a1:11:12	A
2	00:44:ad:a1:f3:1a	B
3	00:50:cc:c1:23:19	O
4	00:99:11:64:b1:3c	C

- O will Traffic lesen der an B geht
- O sendet von seinem Port ein beliebiges Paket mit
  - ETH DST(O)
  - ETH SRC(B)

Wirkung

- Der Switch hat seine Tabelle geändert, Pakete für MAC(B) gehen nun an Port 3
- Nachteile
  - O muss schnell sein
  - B wird unereichbar
  - Pakete können durchfallen
  - Switch Admin
-

Switch Port	MAC	User
1	00:03:11:a1:11:12	A
2	null	B
3	00:50:cc:c1:23:19 00:44:ad:a1:f3:1a	O
4	00:99:11:64:b1:3c	C

ARP spoofing

- Angreifer antwortet schneller auf request als das request-Ziel
- ARP cache übernimmt (meist) ersten Reply

arp who-has Bob tell Alice
arp reply Bob is-at 00:13:37:13:37:00
arp reply Bob is-at 00:80:e3:53:a8:ce

Nachteile

- Nur bereits im ARP Cache existierende IP Adressen werden manipuliert
- Wettrennen gegen echten reply

ARP cache poison

- optimale Angriffstechnik
  - eintragen oder ändern von MAC/IP Zuordnungen im cache des Opfer
  - legales Paket auf Layer 2
  - von den meisten Switchen nicht erkannt

Eintragen

Eintrag erzeugen			opcode = request
Ethernet	source	MAC	mac(Angreifer)
ARP	source	MAC	mac(Angreifer)
ARP	source	IP	ip(Gateway)
Ethernet	destination	MAC	mac(Opfer)
ARP	destination	MAC	00:00:00:00:00:00
ARP	destination	IP	ip(Opfer)

Wirkung

- Angreifer täuscht mit der IP Adresse des Gateway einen request vor
- Opfer legt aus Effizienzgründen die source Daten des Angreifer im Cache ab
- Wenn jetzt eine Verbindung zum oder über den Gateway geht, läuft diese erst beim Angreifer auf (Layer 2)

Ändern

bestehenden Eintrag ändern			opcode = reply
Ethernet	source	MAC	mac(Angreifer)
ARP	source	MAC	mac(Angreifer)
ARP	source	IP	ip(Gateway)
Ethernet	destination	MAC	mac(Opfer)
ARP	destination	MAC	mac(Opfer)
ARP	destination	IP	ip(Opfer)

Wirkung

- replys werden ohnen vorhergegangen request akzeptiert
- der reply ist legal
- aus Effizienz wird Angreifer Gateway

Non-Operating Systems

- Windows XP, 2000, 98
- Mac OS X
- Linux 2.6

Operating Systems

- FreeBSD
- Nov 12 06:18:41 fb /kernel: arp: 24.237.82.161 moved from 00:40:c7:81:22:04 to 00:04:ac:1a:4e:e7 on dc0
- OpenBSD
- Feb 7 12:03:42 ob /bsd: arp info overwritten for 66.68.195.49 by 00:30:7b:ff:50:70 on xl0

Angriffsszenarios

- Denial of Service
- Firewall escaping
- Sniffing
- Proxying, Hijacking, Man in the Middle

denial of service

- viel zu viele Möglichkeiten
- broadcast, timeouts, ddos

firewall escaping

- Gateway vergiften, Angreifer bekommt Opfer IP
- Opfer vergiften, Angreifer wird Gateway
- Angreifer sortiert Opfer traffic von seinem traffic

Sniffing

- vergiften des cache der Opfer
- ip_forward 1
- alles läuft durch Angreifer

MITM

- Angreifer host in der Mitte
- Beide hosts vergiften (sorgsam)
- Einen host vergiften (ausreichend)

Programme

arp-sk

- SwissarmyKnife für ARP
- Kommandozeileninterface
- libnet
- sehr flexibel, wie netcat

arp-sk

```
[root@joker]# arp-sk -r -d batman -S robin -D batman
+ Running mode "reply"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)

+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 52:54:05:F4:62:30
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30)
        192.168.1.2 is at 00:10:a4:9b:6d:81

--- batman (52:54:05:F4:62:30) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30):
        192.168.1.2 is at 00:10:a4:9b:6d:81
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

dsniff

- Kommandozeileninterface
- verschiedene tools
- passiv: sniffen
- aktiv: mitm
- libpcap, libnet, libnids

dsniff

- dsniff (password sniffer)
- filesnarf, mailsnarf, urlsnarf
- arpspoof, macof (besser: arp-sk)
- dnsspoof
- sshmitm (SSHv1), webmitm (HTTPS)

ettercap-ng

Ettercap is a suite for man in the middle attacks on LAN. It features sniffing of live connections, content filtering on the fly and many other interesting tricks.
It supports active and passive dissection of many protocols (even ciphered ones) and includes many feature for network and host analysis.

ng

- GTK, ncurses, relativ bunt
- sniffing, mitm, deciphering, pen testing, bla
- einfach erweiterbar mit plugins

Resultate

- Datenverkehr kann in normalen LANs beliebig abgehört und manipuliert werden
- unter Umständen auch SSH oder HTTPS
- der normale User vertraut seiner Netzwerkanbindung
- durch gepatchte binarys oder client-side vulnerabilities können hosts kompromittiert werden
- wegen ARProtokoll Schwachstellen

MITM Code Projekt

- Opfer geht zu einer HTTP-download Seite
- Angreifer übernimmt auf Ethernet Ebene Gateway
- angefordertes binary wird mit payload gepatcht
- CIH ‘98 keine Änderung der exe Größe
- ELF angeblich noch einfacher

demo: LAN

- 10.0.0.3 Angreifer - Gentoo Linux
- 10.0.0.2 Opfer - Windows XP
- 10.0.0.1 Server - Mac OS X

Ressourcen

- Folien, Links, Referenzen, Software
- http://arp.infoflood.de

DON'T PANIC

- static router MAC
- VLANs
- honeynets, NIDS, IDS
- TTL analyse und sha-1 hashes
- HTTPS root-CAs