ARP Attacks
arp-sk in action

Frédéric Raynal <pappy@miscmag.com>
Cédric Blancher <blancher@cartel-securite.fr>

—

November 15th 2002
Introduction

ARP basics and LAN attacks

arp-sk

Tool presentation

ARP cache poisoning

ARP cache poisoning applications using arp-sk
Introduction: ARP protocol (1/2)

<table>
<thead>
<tr>
<th>0</th>
<th>7</th>
<th>15</th>
<th>23</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>ident. hwa addr.</td>
<td>ident. log. addr.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lng hwa addr.</td>
<td>lng log. addr.</td>
<td>code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>destination ...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... hwa address</td>
<td>destination ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... log. address</td>
<td>source ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... hwa address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>source log address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduction : ARP protocol (2/2)

- ARP (Address Resolution Protocol - RFC 826)
  - binding layer 2 (Ethernet) et layer 3 (IP) addresses
    - client sends a layer 2 broadcast request (who-has)
    - destination answers a layer 2 unicast frame (reply)

- ARP cache usage

- no link between layer 2 and layer 3 information
Attacks: MAC Spoofing

- Ethernet source address spoofing inside Ethernet frame
- Targets switches « CAM tables » (layer 2 addresses)

Pros
- Traffic redirection

Cons
- Target does not receive packets anymore, but continues to emit
- Conflicts inside switch tables
- Not very stealthy
Attacks: ARP Spoofing

- who-has sent using broadcast
- answering instead of legitimate host with spoofed datas

**Pros**
- traffic redirection
- no need to bother with switches

**Cons**
- Incertain result depending on who replies first
Attacks: ARP cache poisoning (1/2)

- create/modify victim ARP cache entries
- victim’s packets are directly sent to attacker

_Create an entry_
- unicast who-has messages (Ok with RFC)
- unicast who-has with spoofed datas

_Modifying an entry_
- ARP spoofing: reply with spoofed datas
Attacks: ARP cache poisoning (2/2)

**Pros**
- traffic redirection
- administrators rarely monitor ARP stuff
- difficult to actively prevent it

**Cons**
- easy to detect
Introduction
ARP basics and LAN attacks

arp-sk
Tool presentation

ARP cache poisoning
ARP cache poisoning applications using arp-sk
arp-sk (1/2)

- Why a new tool?
  - gather functionnalities from different existing tools (arpspoof, macof, arping ...)
  - **complete** ARP headers manipulation (both Ethernet and ARP)
  - add new functionnalities (mapping, promiscuous mode detection, ...)

- arp-sk structure (since version 0.99.0)
  - librairy providing basics functions
  - arp-sk binary
  - modules (dynamic libs) describing « scénarii » (one module == one functionnality)
arp-sk (2/2)

- **arp-sk options**
  - classical ones: emission frequency (in s, \(\mu s\) or random), frame number, interface...
  - basic module: packet type (-w, -r), Ethernet addresses (-s, -d), addresses within ARP message (-S, -D - [IP][:MAC]), randomly generated addresses (Eth and ARP, –rand*)

- **TODO**
  - add new modules
  - porting to other OS (especially OpenBSD and Solaris)
  - create a library to build « network scénarii »
Outline

arp-sk in action

- Introduction
  ARP basics and LAN attacks

- arp-sk
  Tool presentation
  - ARP cache poisoning
    ARP cache poisoning applications using arp-sk
ARP cache updates

- Opportunistic behaviour
- Entries creation
- Entries modification
- Entries deletion

Let’s attack...
Parameters we can play with:

- Ethernet: source MAC address
- Ethernet: destination MAC address
- ARP: layer 2 source
- ARP: layer 2 destination
- ARP: layer 3 source
- ARP: layer 3 destination
Entries creation

- ARP request
- ARP reply (depends on the OS and ARP cache state)
- Gratuitous ARP
- Not very useful
Entries update

- ARP request
- ARP reply
- Gratuitous ARP

- Interesting hosts are usually cached (DNS, GW, etc.)
Entries deletion

- Entries expire
- ARP cache has a limited size (about 500 entries for Linux)
- We flood the cache, but we do not need to delete entries
ARP cache poisoning applications

- **Sniffing**: we can listen to hosts traffic without using promiscuous mode
- **Interception**: we act as a transparent proxy for flows we intercept
- **Modification**: we can inject datas within proxied flows
- **Hijacking**: we can take one part’s place within the flow
- **Decrypting**: classical MiM attack
- **Spoofing**: we can easily spoof another IP within the LAN
- **DoS**: flow destruction
Sniffing with ARP MiM

1. ARP attack
2. ARP attack
3. Routing
4. Robin<->batman traffic
Transparent proxying to intercept and tamper flows

1. ARP attack
2. ARP attack
3. Data interception and modification
4. External data gathering

Robin$\rightarrow$Internet traffic

Batcave−gw

Batman

Joker
Attacked hosts are likely to check their entries...
We can also use ARP MiM ;)

http://www.althes.fr/ressources/avis/smartspoofing.htm
Consequence

→ Once an attacker gets root, he can attack the whole wire flows.
ARP is not secure and easy to fool: security was not in mind. We need stronger mechanisms to enforce security:

- 802.1x
- Secure Link Layer
- Authentication within applications

Must be clear that switches are not security tools!
MISC: french security magazine.

http://www.miscmag.com/
Éric Detoisien for writing winarp-sk and winarp-mim for Win32 plateforms

Laurent Licour and Vincent Royer for their experiments on OS behaviours