**Giao thức và những máy chủ 2**

Ở bài trước, chúng ta đã đi qua rất nhiều giao thức:

* telnet
* http
* ftp
* smtp
* pop3
* imap

Những máy chủ thực thi những giao thức này là mục tiêu của nhiều loại tấn công như:

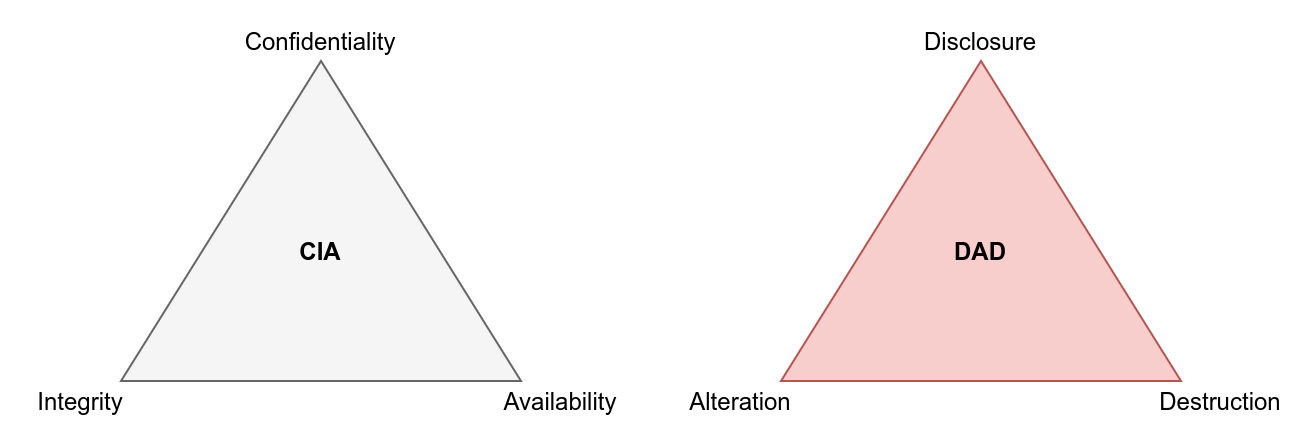
1. Sniffing Attack (thu thập gói tin mạng)

2. Man-in-the-Middle (MITM) Attack

3. Password Attack

4. Vulnerabilities

Mục đích chúng ta bảo vệ là: confidentiality, integrity, availability. Mục đích 1 cuộc tấn công là disclosure, alternation, destruction (DAD).



Những cuộc tấn công trực tiếp ảnh hưởng đến bảo mật của hệ thống. Ví dụ, thu thập gói tin mạng đã vi phạm confidentiality và dẫn đến sự rò rỉ thông tin. Một cuộc tấn công mật khẩu thành công có thể dẫn đến disclosure. Mặt khác, MITM làm vỡ tính toàn vẹn (integrity) cảu hệ thống vì nó có thể thay đổi dữ liệu.

Những lỗ hổng là 1 phạm vi rộng hơn và những lỗ hổng được khai thác có những ảnh hưởng khác nhau đến hệ thống đích. Ví dụ, khai thác lỗ hổng DoS có thể ảnh hưởng đến availability của hệ thống, trong khi khai thác RCE có thể dẫn đến những cuộc tấn công nghiêm trọng hơn.

**Sniffing Attack**

Cuộc tấn công Sniffing là dùng 1 công cụ thu thập gói tin mạng để tìm hiểu về đối tượng. Khi 1 giao thức giao tiếp ở dạng văn bản rõ ràng, dữ liệu được trao đổi có thể bị đánh cắp bởi bên thứ 3 để phân tích.

Tấn công sniffing có thể được thực hiện bằng card mạng Ethernet (802.3), miễn là người dùng có quyền thích hợp. Có rất nhiều công cụ để thu thập gói tin mạng như:

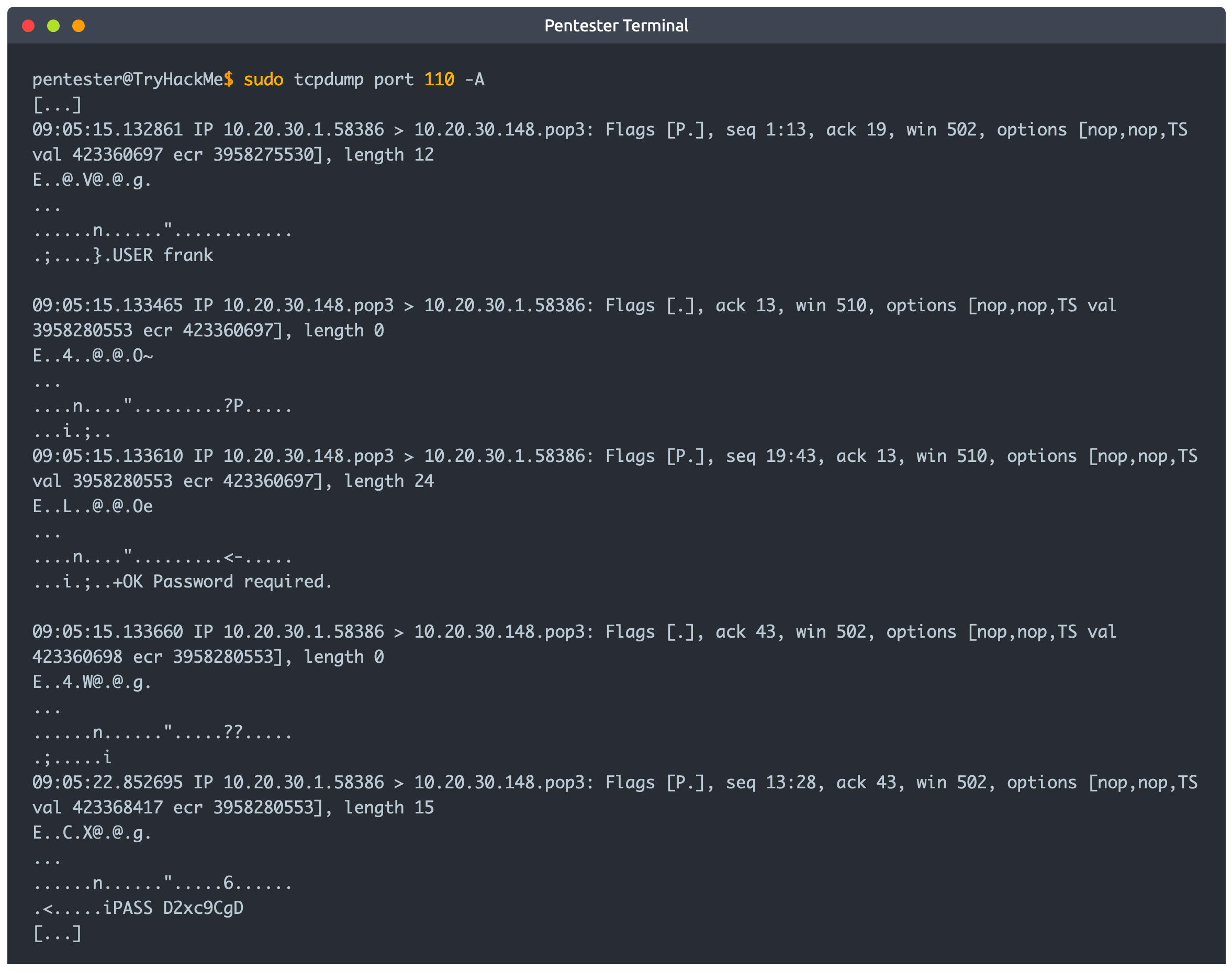
1. Tcpdump là 1 giao diện dòng lệnh (CLI) mã nguồn mở miễn phí có thể hoạt động trên nhiều hệ điều hành khác nhau.

2. Wireshark là 1 giao diện người dùng (GUI) mã nguồn mở miễn phí có thể hoạt động trên 1 số hệ điều hành như Linux, macOS, MS Windows

3. Tshark là 1 CLI thay thế của Wireshark.

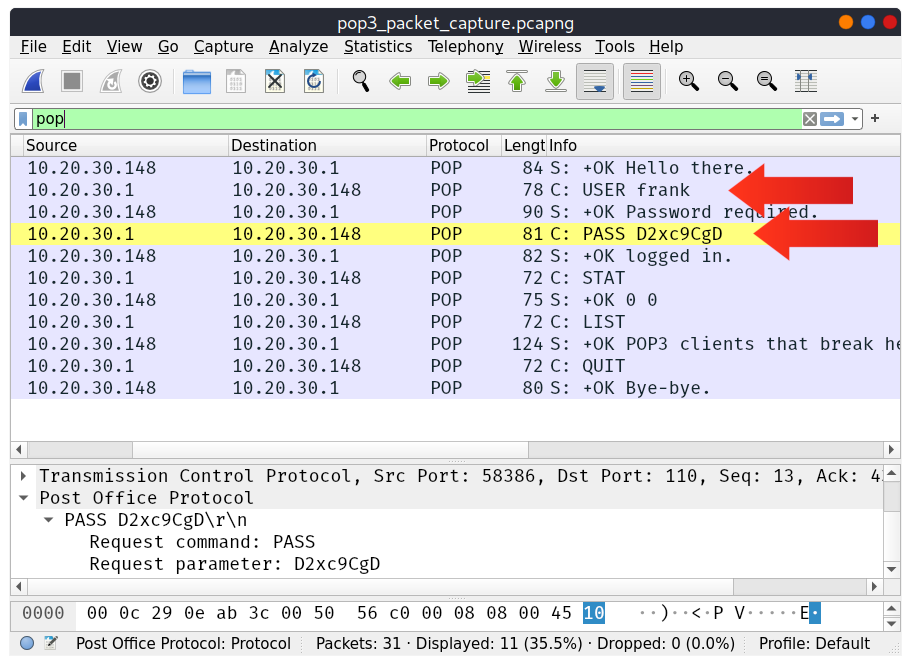
Xem xét 1 người dùng kiểm tra tin nhắn bằng việc dùng POP3. Đầu tiên, chúng ta sẽ dùng tcpdump để thu thập tên người dùng và mật khẩu. Chúng ta dùng lệnh sudo tcpdump port 110 -A. Loại tấn công này yêu cầu truy cập vào lưu lượng mạng, ví dụ qua wiretap hay switch với port mirroring. Ngoài ra, chúng ta có thể truy cập lưu lượng nếu tấn công MITM thành công.

Dùng sudo vì thu thập gói tin yêu cầu quyền root. Chúng ta muốn giới hạn số lượng gói tin được thu thập và hiển thị khi chúng trao đổi với máy chủ POP3. pop3 dùng cổng 110. Chúng ta muốn hiển thị nội dung ở dạng ASCII, thêm -A



Gói tin đầu tiên rõ ràng đã hiển thị “USER frank” và “PASS D2xc9CgD”.

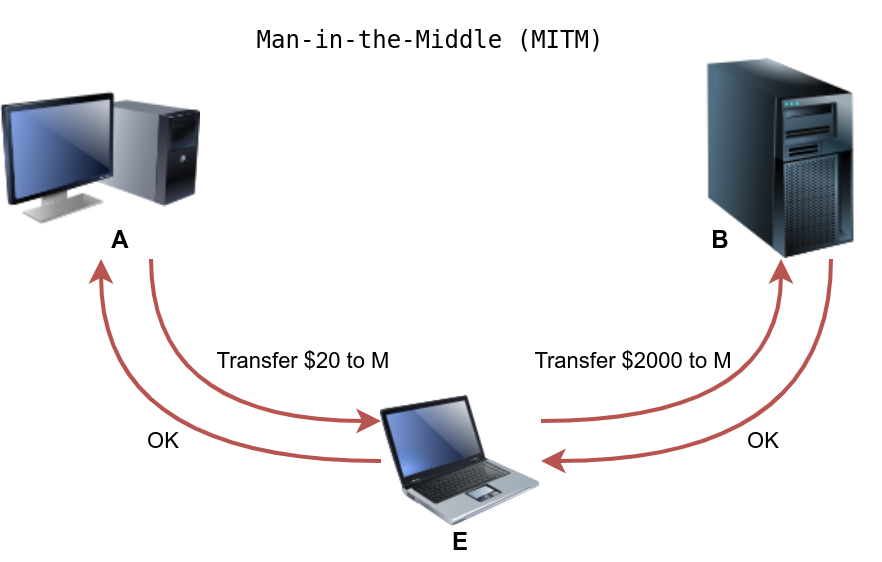
Chúng ta cũng có thể dùng wireshark để thu được kết quả tương tự. Nhập pop ở mục filter để lọc những lưu lượng chúng ta thích.



Bất cứ giao thức nào dùng văn bản rõ ràng đều nhạy cảm với loại tấn công này. Yêu cầu duy nhất để thành công là phải truy cập vào 1 hệ thống giữa hai hệ thống đang giao tiếp. Yêu cầu sự chú ý; biện pháp giảm thiểu nằm ở việc thêm 1 lớp mã hóa lên trên bất kỳ giao thức mạng nào. Nhìn chung TLS đã được thêm vào HTTP, FTP, SMTP, POP3, IMAP, … Để truy cập từ xa,telnet đã được thay thế bằng SSH.

**Man-in-the-Middle Attack**

MITM xảy ra khi nạn nhân tin tưởng họ đang giao tiếp với điểm đến hợp pháp (B) nhưng không biết đang giao tiếp với kẻ tấn công (E). Trong hình dưới, A yêu cầu chuyển 20$ cho M; Tuy nhiên, E thay thế tin nhắn đó với 1 cái mới và gửi cho B.



Tấn công này rất dễ thực hiện nếu hai bên không kiểm tra tính xác thực và toàn vẹn của mỗi tin nhắn. Trong một vài trường hợp, giao thức được chọn không xác thực an toàn và kiểm tra tính toàn vẹn; Rất nhạy cảm với loại tấn công này.

Bất cứ lúc nào duyệt qua HTTP, bạn rất nhạy cảm với tấn công MITM. Và điều đáng sợ là bạn không nhận ra nó. Nhiều công cụ sẽ hỗ trợ bạn thực hiện 1 cuộc tấn công như vậy như Eltercap và Bettercap.

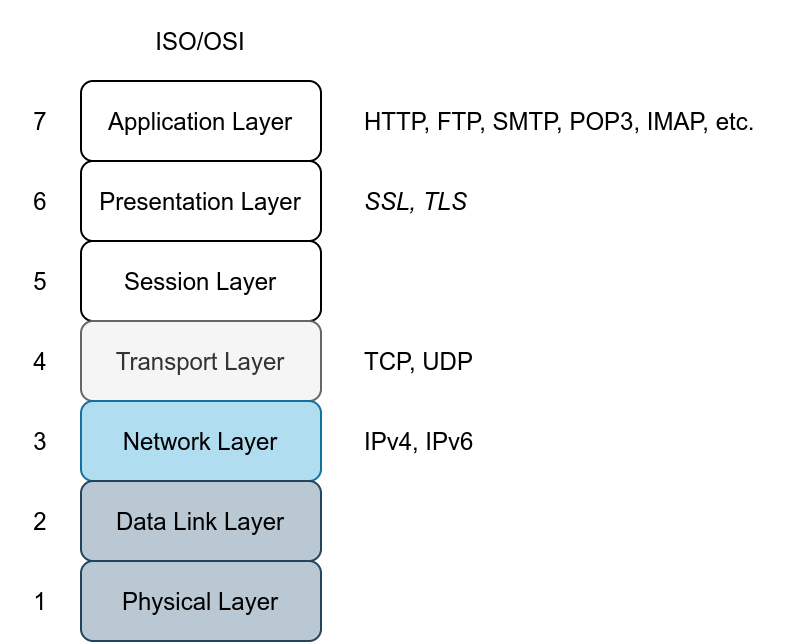
MITM có thể ảnh hưởng đến những giao thức có văn bản rõ ràng như FTP, SMTP và POP3. Giảm thiểu loại tấn công này yêu cầu mã hóa. Biện pháp dựa trên xác thực phù hợp cùng với mật mã hóa hoặc ký tên các tin nhắn trao đổi. Với sự hỗ trợ của Public Key Infrastructure (PKI) và những chứng chỉ root tin cậy, TLS bảo vệ khỏi những cuộc tấn công MITM.

**Transport Layer Security (TLS)**

Trong bài này, chúng ta học về 1 biện pháp tiêu chuẩn để bảo vệ confidentiality và integrity của những gói tin được trao đổi. Cách tiếp cận này có thể bảo vệ những cuộc tấn công sniffing và MITM.

SSL (secure sockets layer) bắt đầu khi thế giới web chứng kiến những ứng dụng mới như mua sắm trực tuyến và thanh toán online. Netscape giới thiệu SSL năm 1994 với SSL bản 3.0 được phát hành năm 1996. Nhưng cuối cùng, cần phải bảo mật nhiều hơn và giao thức TLS được giới thiệu năm 1999. Hãy xem chúng phù hợp với mô hình mạng như thế nào?

Những giao thức phổ biến thường gửi dữ liệu ở văn bản rõ ràng; Bất cứ ai có thể truy cập mạng cũng thu thập, lưu trữ và phân tích những tin nhắn đã trao đổi được. Hình dưới là mô hình mạng OSI; chúng ta có thể thêm mật mã hóa cho những giao thức của ta ở lớp hiển thị (presentation). Do đó, dữ liệu sẽ được hiển thị ở dạng mã hóa thay vì văn bản rõ ràng.



Vì sự liên hệ gần gũi giữa SSL và TLS, nó có thể dùng thay thế cho nhau. Tuy nhiên TLS bảo mật hơn SSL và thực tế nó đã thay thế SSL.

Một giao thức văn bản rõ ràng có thể được chỉnh sửa để sử dụng mã hóa qua SSL/TLS.

| **Protocol** | **Default Port** | **Secured Protocol** | **Default Port with TLS** |
| --- | --- | --- | --- |
| HTTP | 80 | HTTPS | 443 |
| FTP | 21 | FTPS | 990 |
| SMTP | 25 | SMTPS | 465 |
| POP3 | 110 | POP3S | 995 |
| IMAP | 143 | IMAPS | 993 |

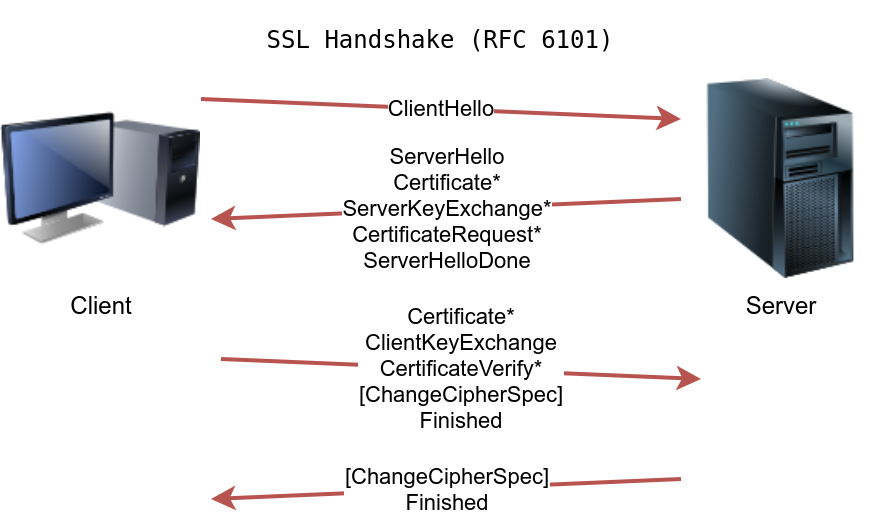
Xem xét trường hợp HTTP. Ban đầu, để truy suất trang web qua http, trình duyệt web cần thực hiện ít nhất hai bước sau:

1. Thiết lập kết nối TCP với máy chủ web từ xa
2. Gửi yêu cầu http đến máy chủ web như GET hay POST.

Https yêu cầu thêm 1 bước nữa để mã hóa lưu lượng. Bước mới xảy ra sau khi thiết lập kết nối TCP và trước khi gửi yêu cầu Get/Post.

1. Thiết lập kết nối TCP với máy chủ web từ xa
2. Thiết lập kết nối SSL/TLS
3. Gửi yêu cầu http đến máy chủ web như GET hay POST.

Để thiết lập kết nối SSL/TLS, máy khách cần thực hiện handshake phù hợp với máy chủ. Dựa trên RFC 6101, thiết lập kết nối SSL sẽ như sau:



Kết nối SSL/TLS có thể nhìn phức tạp dựa vào kiến thức về mật mã của bạn, nhưng chúng ta có thể xác định 4 bước sau:

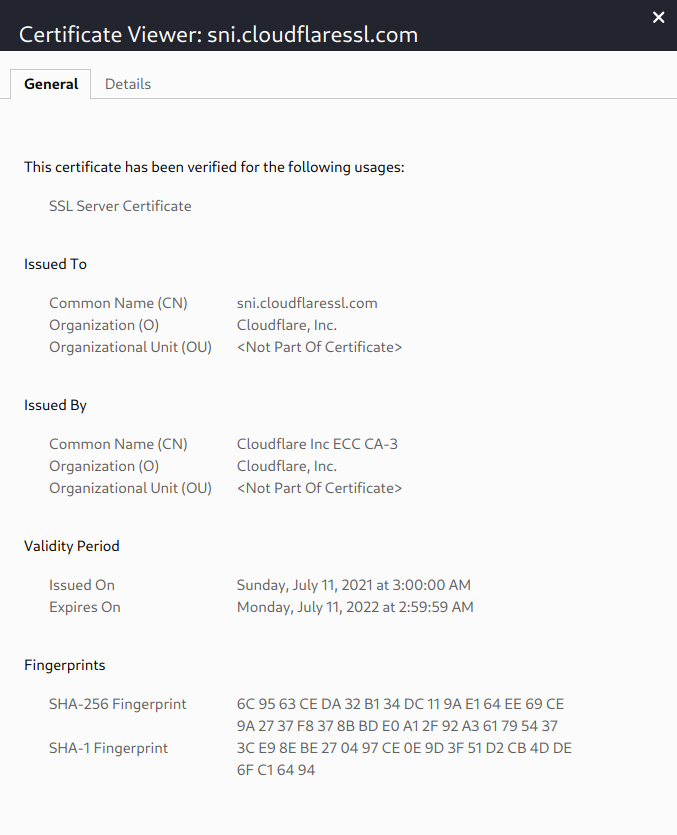
1. Máy khách gửi 1 ClientHello đến máy chủ để chỉ định thuật toán.
2. Máy chủ phản hồi với ServerHello, chỉ ra những tham số kết nối. Máy chủ cung cấp chứng chỉ nếu xác thực được yêu cầu. Chứng chỉ là 1 tập tin số ; Nó được đánh dấu bởi bên thứ 3. Hơn nữa, nó có thể gửi thêm thông tin cần thiết để tạo 1 key chính, trong tin nhắn ServerKeyExchange, trước khi gửi tin nhắn ServerHelloDone.
3. Máy khách phản hồi với ClientKeyExchange, chứa thêm thông tin cần thiết để tạo 1 key chính. Hơn nữa, nó điều chỉnh để sử dụng mã hóa và thông báo đến máy chủ việc sử dụng tin nhắn ChangeCipherSpec.
4. Máy chủ điều chỉnh để sử dụng mã hóa và thông báo với máy khách trong tin nhắn ChangeCipherSpec.

Nếu nó vẫn còn phức tạp, đừng lo lắng; chúng ta chỉ cần ý chính của nó. Máy khách có thể đồng ý chìa khóa bí mật với máy chủ có chứng chỉ công cộng. Key bí mật này được tạo ra rất bảo mật để mà bên thứ ba theo dõi kênh sẽ không thể khám phá ra. Hơn nữa, giao tiếp giữa máy khách và máy chủ sẽ được mã hóa bằng việc dùng key được tạo ra.

Do đó, khi handshake SSL/TLS được thiết lập, những yêu cầu http và dữ liệu trao đổi không thể truy cập với bất cứ ai xem kênh giao tiếp.

Lưu ý cuối cùng, để SSL/TLS được hiệu quả, đặc biệt khi duyệt web qua https, chúng ta dựa vào các chứng chỉ công khai được ký bởi cơ quan chứng chỉ được các hệ thống tin tưởng.

Nói cách khác, khi chúng ta duyệt Tryhackme qua https, trình duyệt sẽ mong máy chủ Tryhackme cung cấp 1 chứng chỉ đã được chứng nhận bởi cơ quan có thẩm quyền. Như vậy, trình duyệt sẽ đảm bảo nó đang giao tiếp với đúng máy chủ và tấn công MITM không thể xảy ra.



Trong hình trên, chúng ta có thể thấy những thông tin sau:

Giấy chứng nhận được cấp cho ai? Tên của công ty sẽ sử dụng chứng chỉ đó.

Ai đã phát hành chứng chỉ? Đó là tác giả đã phát hành chứng chỉ

Thời gian hợp lệ. Bạn không muốn dùng 1 chứng chỉ đã hết hạn.

May mắn, chúng ta không phải kiểm tra chứng chỉ mỗi khi chúng ta thăm 1 trang web; Trình duyệt sẽ làm điều đó. Nó sẽ đảm bảo rằng chúng ta đang giao tiếp với đúng máy chủ và đảm bảo quá trình giao tiếp của ta an toàn, nhờ vào chứng chỉ của máy chủ.

**Secure Shell (SSH)**

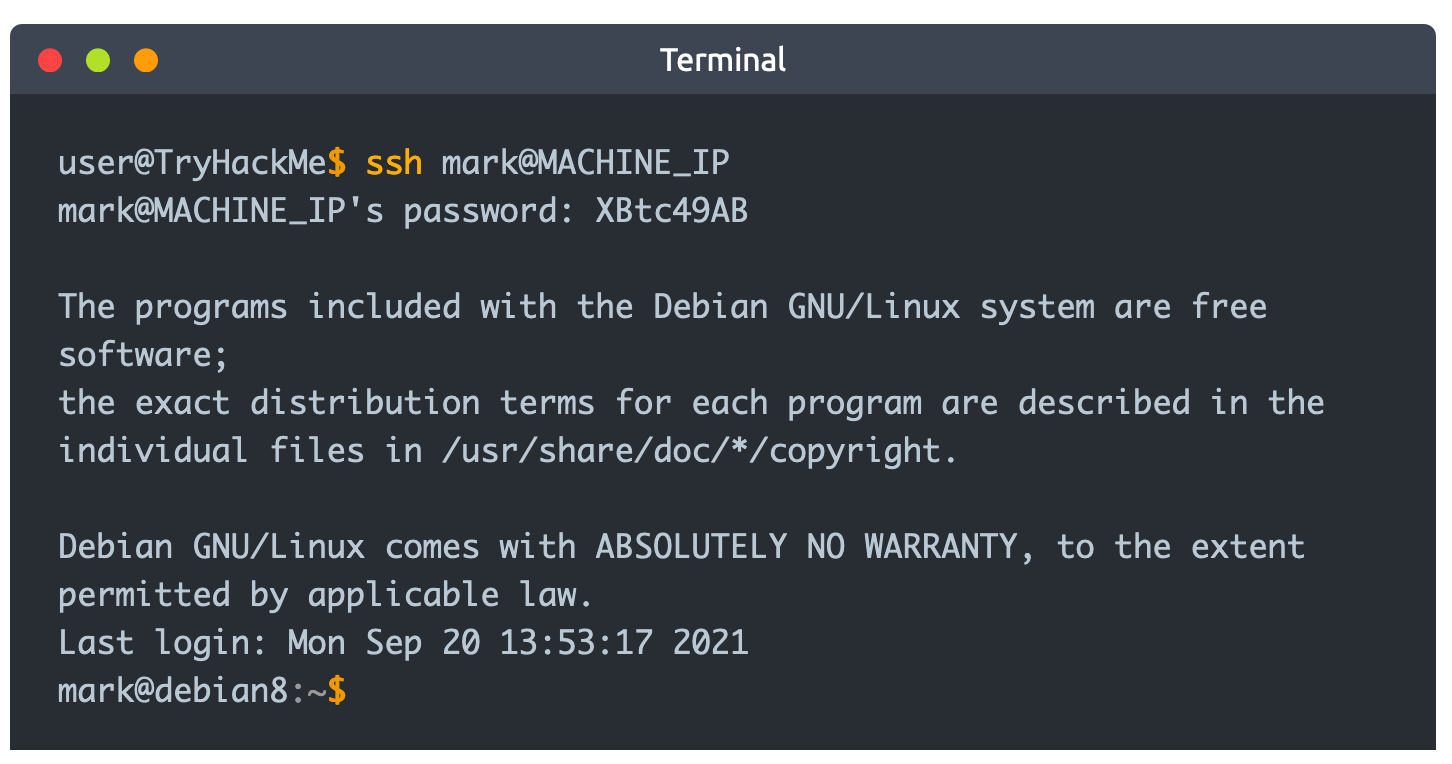
Secure Shell được tạo ra để cung cấp 1 cách bảo mật hơn để quản trị từ xa. Nói cách khác, nó để bạn kết nối đến hệ thống khác qua mạng và thực thi lệnh.Nói đơn giản, S trong SSH là secure có thể được tổng hợp như sau:

1. Bạn có thể xác nhận danh tính của máy chủ từ xa
2. Những tin nhắn trao đổi được mã hóa và chỉ có thể được giải mã bởi người nhận
3. Cả hai bên có thể phát hiện sự điều chỉnh trong tin nhắn

Để dùng SSH, bạn cần máy chủ SSH và máy khách SSH. Máy chủ lắng nghe trên cổng 22. Máy khách có thể xác thực bằng:

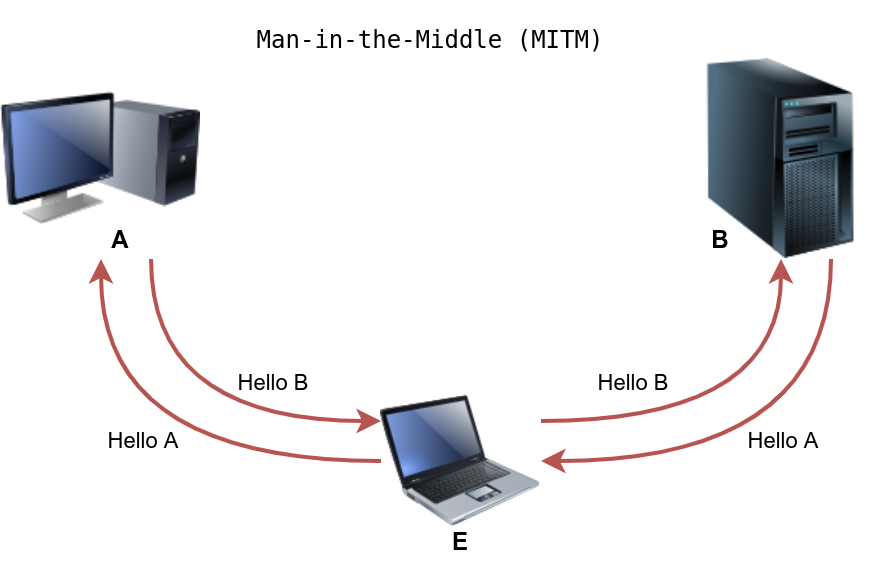
* Tên người dùng và mật khẩu
* Key riêng và chung (sau khi máy chủ SSH được cấu hình để nhận ra key chung)

Ở Linux, macOS và MS Windows sau 2018, bạn có thể kết nối đến máy chủ SSH bằng lệnh sau: ssh username@MACHINE\_IP. Lệnh này sẽ cố gắng kết nối đến máy chủ có địa chỉ IP trên với tên đăng nhập username. Nếu máy chủ SSH đang lắng nghe trên cổng mặc định, nó sẽ yêu cầu mật khẩu. Khi xác thực thành công, người dùng có thể truy cập terminal của hệ thống từ xa.



SSH rất đáng tin cậy để quản trị từ xa vì tên người dùng và mật khẩu gửi đi đã được mã hóa; Hơn nữa, tất cả những lệnh chúng ta thực thi trên hệ thống cũng sẽ được gửi qua kênh mã hóa.

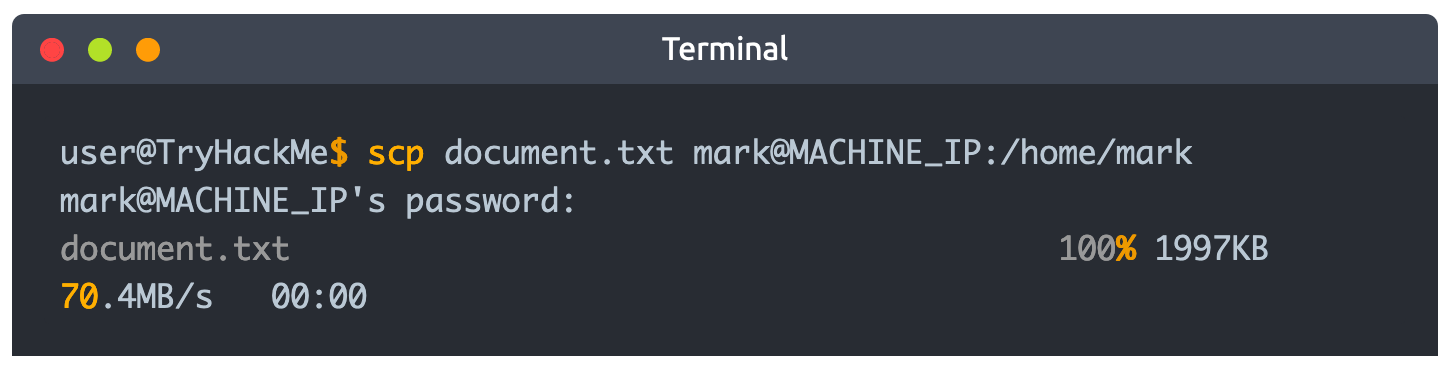
Lưu ý nếu đây là lần đầu tiên ta kết nối đến hệ thống đó, chúng ta sẽ cần kiểm tra key chung của máy chủ SSH để tránh những cuộc tấn công MITM. Trong trường hợp của SSH, chúng ta không thường có bên thứ ba để kiểm tra key chung có hợp lệ không, vì vậy chúng ta làm thủ công.



Chúng ta có thể dùng SSH để vận chuyển tập tin bằng SCP (Secure Copy Protocol).

scp mark@MACHINE\_IP:/home/mark/archive.tar.gz ~. Lệnh này sẽ sao chép 1 tập tin tên archive.tar.gz từ hệ thống đích đặt ở thư mục /home/mark đến ~, thư mục home của root.

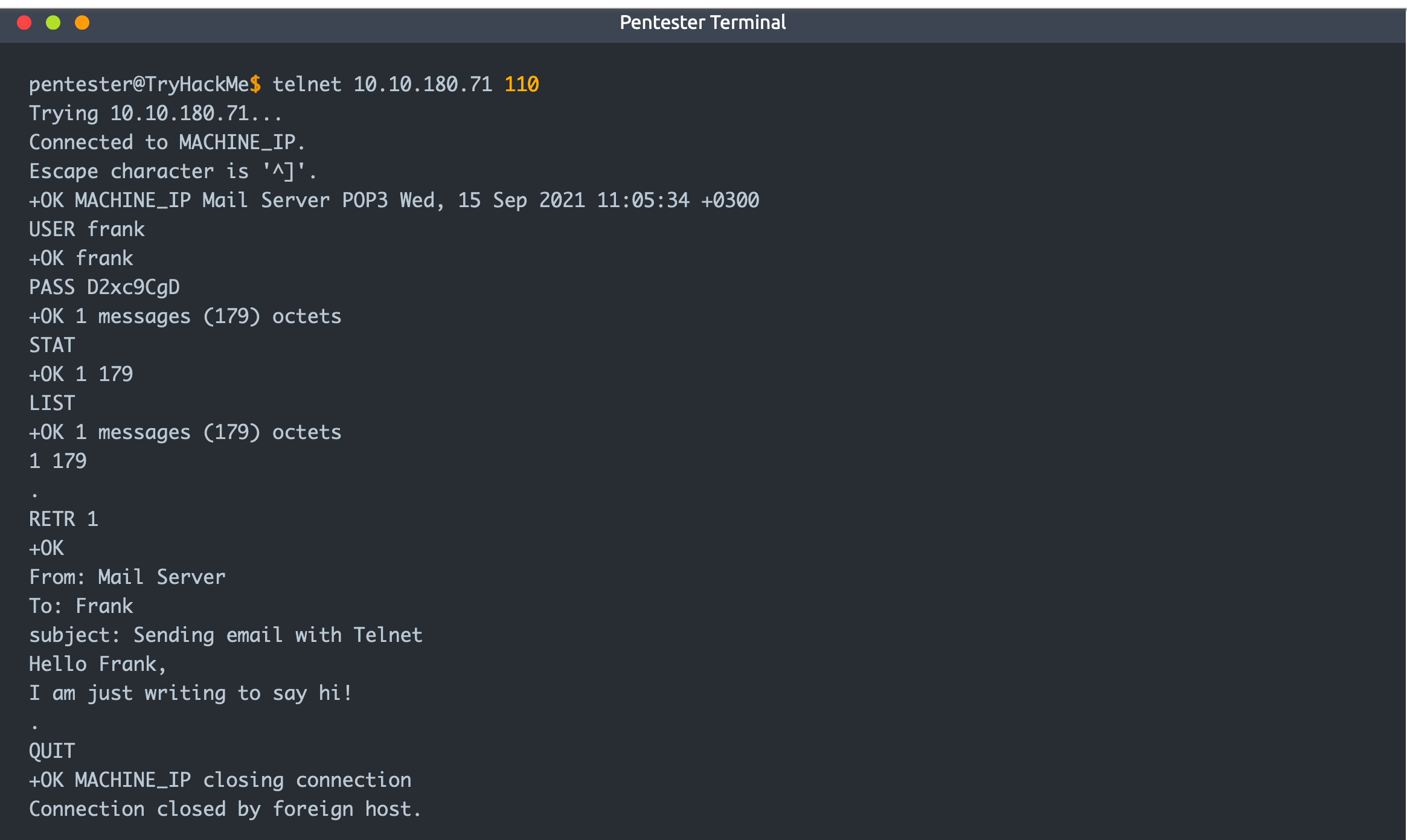
Một lệnh khác là: scp backup.tar.bz2 mark@MACHINE\_IP:/home/mark. Lệnh này sẽ sao chép tập tin backup.tar.bz2 từ hệ thống cục bộ đến thư mục /home/mark trên hệ thống đích.



FTP có thể được bảo mật với SSL/TLS bằng việc dùng giao thức FTPS sử dụng cổng 990. Điều đáng lưu ý là FTP cũng có thể được bảo mật bằng giao thức SSH là giao thức SFTP. Mặc định dịch vụ này lắng nghe trên cổng 22 như SSH.

**Tấn công mật khẩu:**

Rất nhiều giao thức yêu cầu bạn xác thực. Xác thực để chứng minh bạn là ai.



Xác thực dựa vào rất nhiều yếu tố sau đây:

1. Một vài thứ bạn biết: mật khẩu, mã PIN

2. Một vài thứ bạn có: SIM card, RFID card

3. Một vài thứ bạn là: dấu vân tay

Bài này sẽ tập trung vào những cuộc tấn công chống lại mật khẩu. Dựa trên 150 triệu tài khoản và mật khẩu bị rò ri từ Adobe năm 2013, 10 mật khẩu thông dụng nhất là:

* 123456
* 123456789
* password
* adobe123
* 12345678
* qwerty
* 1234567
* 111111
* photoshop
* 123123

Những cuộc tấn công chống lại mật khẩu được thực hiện bằng:

* Đoán mật khẩu
* Thư viện tấn công
* Tấn công dồn dập

Hãy tập trung vào thư viện tấn công. Qua nhiều thời gian, những hackers đã biên dịch 1 danh sách mật khẩu từ những vụ rò rỉ dữ liệu.

Hydra là 1 công cụ tự động thử những mật khẩu phổ biến từ các danh sách, hỗ trợ nhiều giao thức như FTP, POP3, IMAP, SMTP, SSH và tất cả những phương thức liên quan đến HTTP

Cú pháp lệnh chung là:

hydra -l username -P wordlist.txt server service

* -l username: -l nên đứng trước tên người dùng, tên đăng nhập của đối tượng.
* -P wordlist.txt: là 1 tập tin chứa danh sách các mật khẩu bạn muốn thử với tên người dùng được cung cấp.
* server là tên máy chủ hoặc địa chỉ IP
* service chỉ ra dịch vụ bạn muốn thực hiện cuôc tấn công từ điển

Xem xét các ví dụ cụ thể sau:

* hydra -l mark -P /usr/share/wordlists/rockyou.txt 10.10.180.71 ftp
* hydra -l mark -P /usr/share/wordlists/rockyou.txt ftp://10.10.180.71
* hydra -l frank -P /usr/share/wordlists/rockyou.txt 10.10.180.71 ssh

Đây là một số lựa chọn bạn có thể thêm:

-s PORT

-V hay -vV chi tiết hơn, làm cho Hydra hiển thị tên người dùng và mật khẩu được kết hợp.

-t n : n là số lượng kết nối song song đến đối tượng. -t 16 sẽ tạo 16 luồng để kết nối với đối tượng.

-d: để debug, để có thêm thông tin chi tiết về cái gì sắp diễn ra. Đầu ra có thể làm bạn bớt thất vọng. Ví dụ, nếu Hydra cố gắng kết nối đến cổng đã đóng và hết hạn, -d sẽ tiết lộ điều này ngay lập tức.