SSH

SSH là gì và cách sử dụng SSH cho người mới bắt đầu

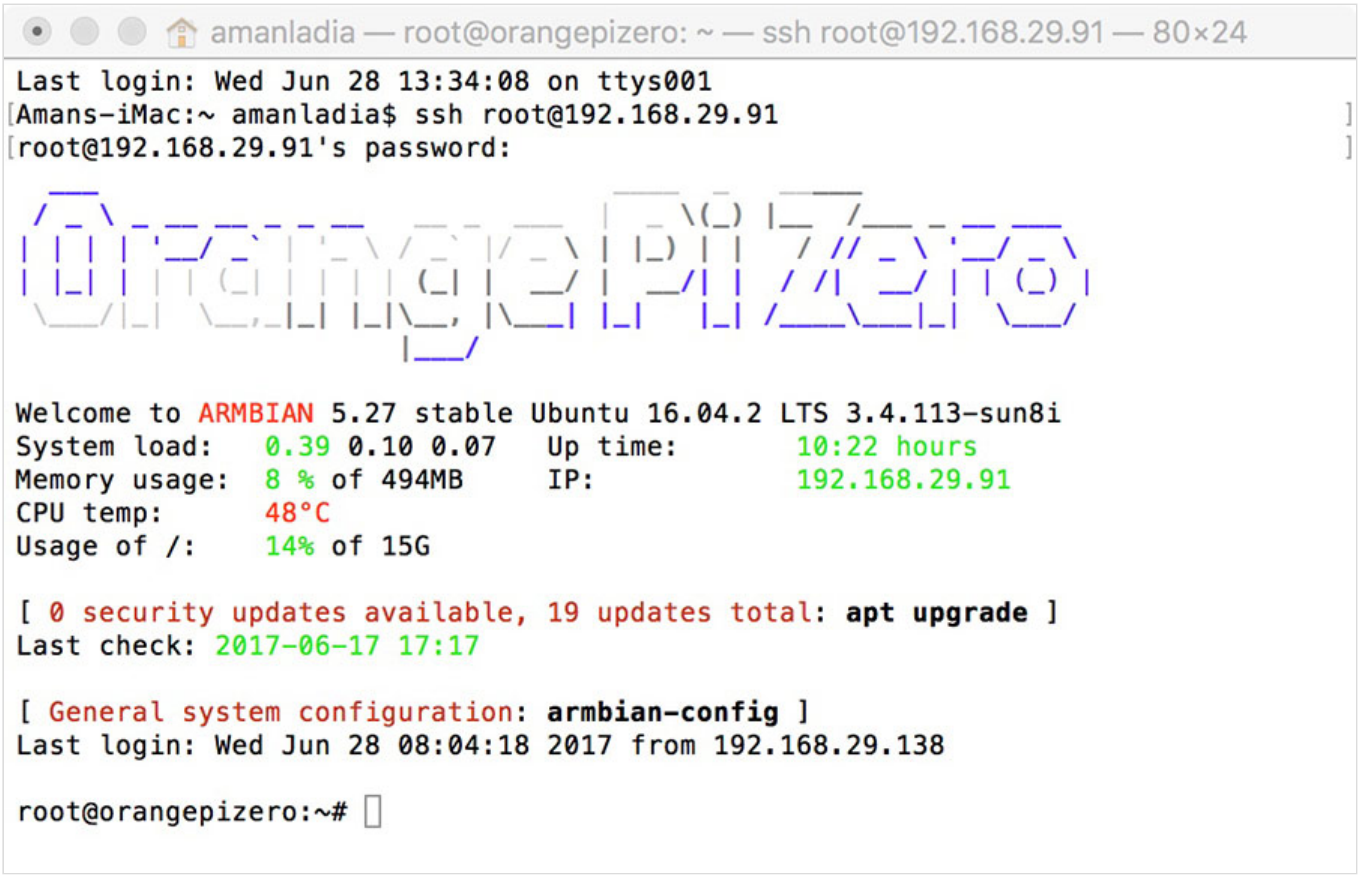
Last Updated on: Tháng Một 30th, 2018 Hai G. 13 Phút Đọc

Contents

SSH là gì

SSH, hoặc được gọi là Secure Shell, là một giao thức điều khiển từ xa cho phép người dùng kiểm soát và chỉnh sửa server từ xa qua Internet. Dịch vụ được tạo ra nhằm thay thế cho trình Telnet vốn không có mã hóa và sử dụng kỹ thuật cryptographic để đảm bảo tất cả giao tiếp gửi tới và gửi từ server từ xa diễn ra trong tình trạng mã hóa. Nó cung cấp thuật toán để chứng thực người dùng từ xa, chuyển input từ client tới host, và relay kết quả trả về tới khách hàng.

Hình bên dưới thể hiện một giao diện Windows SSH điển hình. Bất kể user Linux hoặc macOS nào cugn4 đều có thể SSH tới server từ xa trực tiếp từ cửa sổ terminal. Windows users có thể sử dụng những SSH clients như là Putty. Bạn có thể thực thi lệnh shell cũng như việc bạn đang thực sự vận hành máy tính vật lý.



Bạn đã biết SSH là gì vậy hãy tiếp tục tìm hiểu về cách thức hoạt động của SSH,bên cạnh việc tìm hiểu về công nghệ được sử dụng để đảm bảo tính an toàn cho các kết nối từ xa. Nó sẽ gồm nhiều lớp và loại mã hóa được sử dụng, tùy thuộc vào mục đích của từng layer.

SSH hoạt động như thế nào

Để hiểu SSH là gì thì trước tiên bạn cần phải biết nó hoạt động như thế nào. Nếu bạn đang sử dụng Linux hoặc Mac, sử dụng SSH rất đơn giản. Nếu bạn sử dụng Windows, bạn chỉ cần sử dụng những SSH client để mở kết nối SSH. Những trình SSH client phổ biến là Putty, bạn có thể xem thêm tại đây.

Đối với người dùng xài MAC và Linux, hãy mở **terminal** và làm theo hướng dẫn sau:

Lệnh SSH có 3 phần:

```
ssh {user}@{host}
```

SSH key command cho hệ thống biết là bạn muốn mở một kết nối được mã hóa Secure Shell Connection. **{user}** đại diện cho tài khoản người dùng bạn muốn dùng để truy cập. Ví dụ, bạn muốn truy cập user **root**, thì thay root tại đây. User root là user quản trị hệ thống với toàn quyền để chỉnh

sửa bất kỳ điều gì trên hệ thống. **{host}** đại diện cho máy tính bạn muốn dùng để truy cập. Nó có thể là một địa chỉ IP **(ví dụ 244.235.23.19)** hoặc một tên miền (ví dụ, www.xyzdomain.com).

Khi bạn nhấn enter, nó sẽ hỏi bạn nhập mật khẩu tương ứng cho tài khoản. Khi bạn gõ, bạn sẽ không thấy bất kỳ dấu hiệu nào trên màn hình, nhưng nếu bạn gõ đúng mật khẩu và nhấn enter, bạn sẽ vào được hệ thống và nhận thông báo đăng nhập thành công.

Nếu bạn muốn tìm hiểu thêm về lệnh SSH, hãy tham khảo tại đây

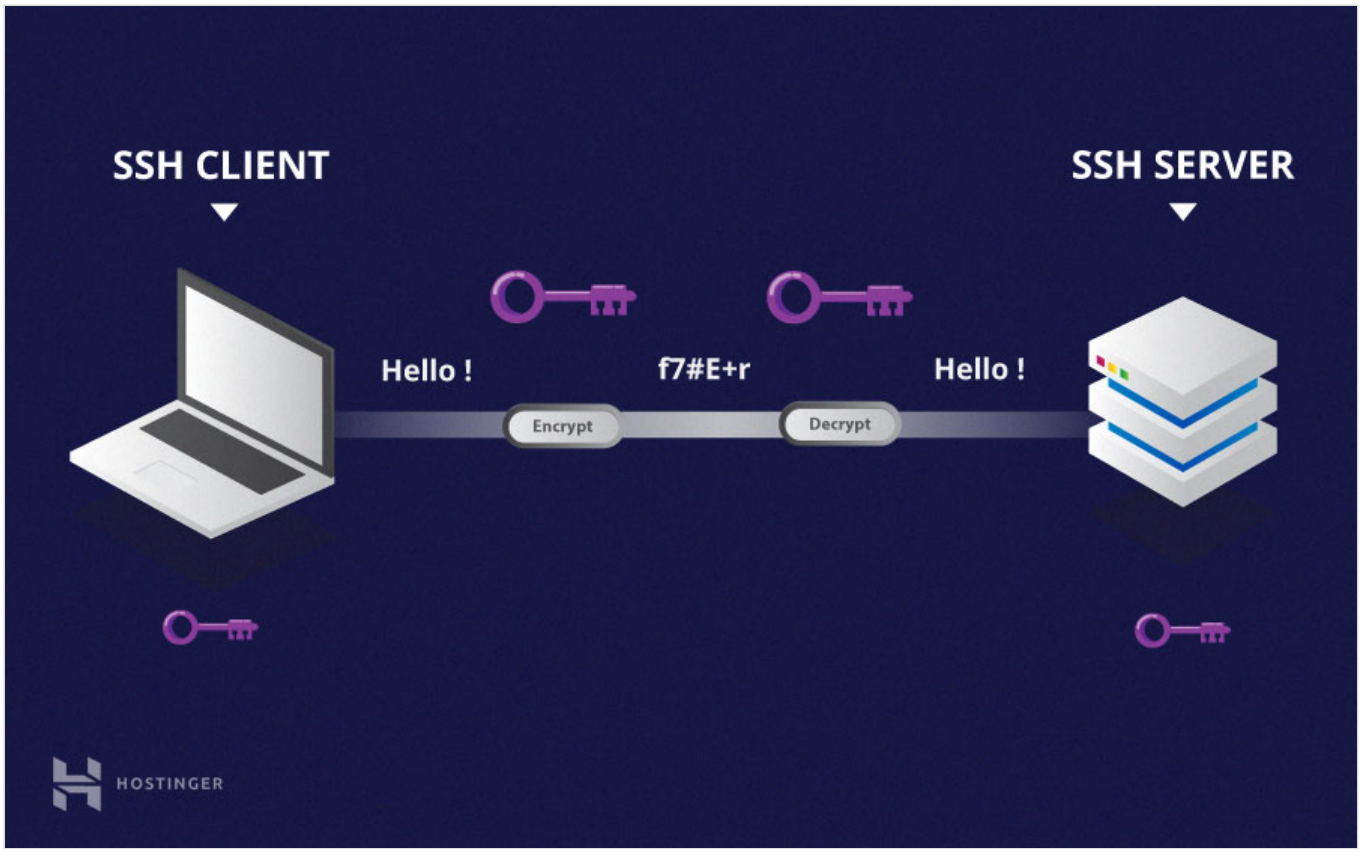
Hiểu về nhiều kỹ thuật mã hóa khác nhau

Lợi điểm khiến SSH hơn hẳn những giao thức cũ là khả năng mã hóa và truyền tải dữ liệu an toàn giữa host và client. **Host** đại diện cho máy chủ từ xa bạn muốn kết nối tới và **client** là máy tính của bạn dùng để truy cập tới host. Có 3 cách khác nhau để mã hóa qua SSH:

1. Symmetrical encryption
2. Asymmetrical encryption
3. Hashing.

Symmetric Encryption

Symmetric encryption là một dạng mã hóa sử dụng **secret key** ở cả 2 chiều mã hóa và giải mã tin nhắnb bởi cả host và client. Có nghĩa là ai nắm được khóa đều có thể giải mã tin nhắn trong quá trình chuyền.



Symmetrical encryption thường được gọi là **shared key** hoặc **shared secret** encryption. Vì có một khóa được sử dụng, hoặc một cặp khóa (pair key) mà một khóa có thể được tính ra từ khóa kia.

Symmetric keys được sử dụng để mã hóa toàn bộ liên lạc trong phiên giao dịch SSH. Cả client và server tạo chung một key bí mật như là một phương thức thỏa thuận, và key đó không được tiết lộ cho bên thứ ba. Quá trình tạo symmetric key được thực hiện bởi **key exchange algorithm**.

Điều khiến cho thuật toán an toàn là vì key không được truyền giữa client và host. Thay vào đó, cả 2 máy tính chia sẽ thông tin chung và sau đó sử dụng chúng để tính ra khóa bí mật.Kể cả có máy khác bắt được thông tin chung, nó cũng không thể tính ra key bí mật vì không biết được thuật toán tạo key.

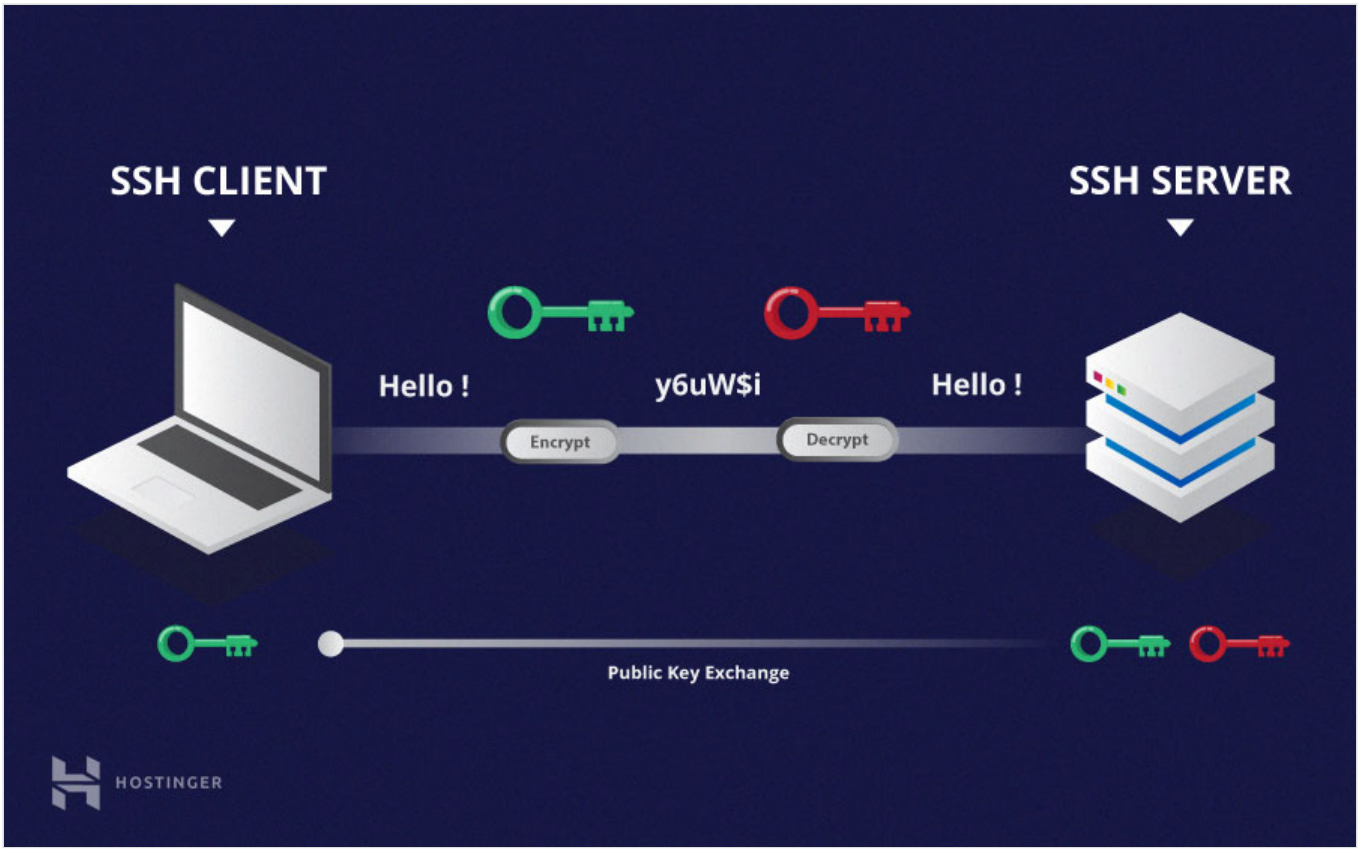
Cũng phải lưu ý rằng, tuy nhiên secret token được sử dụng cho một phiên SSH nhất định, và được tạo bởi chứng thực của client. Khi key đã được tạo, tất cả packets truyền giữa 2 máy phải được mã hóa bởi private key. Việc này bao gồm cả mật khẩu điền vào bởi user, vì vậy mật khẩu cũng có thể được bảo vệ khỏi những "lính bắn tỉa packet" trên mạng.

Một số loại symmetrical encryption ciphers đã tồn tại, bao gồm, những không giới hạn AES (Advanced Encryption Standard), CAST128, Blowfish etc. Trước khi thiết lập kết nối an toàn client và host sẽ đồng ý loại cipher nào được sử dụng, bằng cách xuất bản danh sách cyphers được hỗ trợ để tham khảo. Cypher thích hợp nhất ở phía client sẽ hiển thị trong danh sách của host như là một bidirectional cypher.

Ví dụ, nếu 2 máy Ubuntu 14.04 LTS liên lạc với nhau qua SSH, nó sẽ sử dụng **aes128-ctr** làm cipher mặc định.

Asymmetric Encryption

Không giống với symmetrical encryption, asymmetrical encryption sử dụng 2 khóa khác nhau để mã hóa và giải mã. 2 khóa này được gọi là **public key** và **private key**. Cả 2 hình thành nên một cặp khóa là **public-private key pair**.



Khóa public, như tên gọi của nó sẽ được công khai cho tất cả các bên liên quan. Mặc dù nó liên quan mật thiết đến private key về chức năng, nhưng private key không thể được tính toán ra từ một public key. Sự liên quan này rất phức tạp: thư được mã hóa bởi public key của một máy, và chỉ có thể được giải mã bởi private key của chính máy đó. Sự liên quan một chiều này có nghĩa là public key không thể giải mã chính thư của nó, hoặc không thể giải mã bất kỳ thứ gì được mã hóa bằng private key.

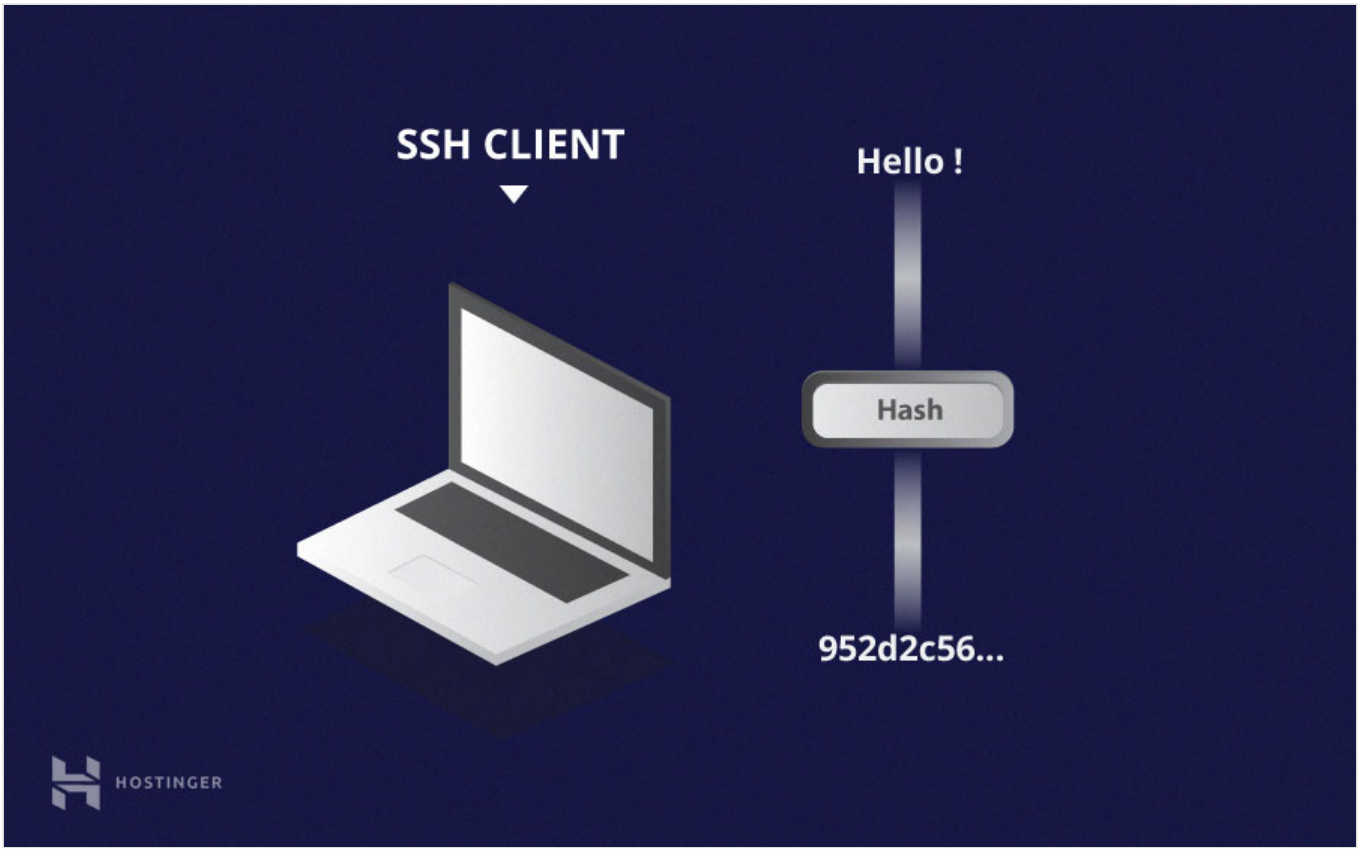
Private key phải luôn luôn được đảm bảo an toàn, ví dụ, kết nối an toàn, không có bên thứ 3 biết. Sức mạnh của cả chu trình kết nối phụ thuộc vào việc private key có bị tiết lộ hay không, vì chỉ có nó mới có khả năng giải mã thư được truyền đi mà được mã hóa bởi public key. Vì vậy, bất kỳ bên nào có thể giải mã thư được ký bởi public key có nghĩa là bên đó đang sở hữu private key tương ứng.

Không giống với quan niệm thông thường, asymmetrical encryption không được dùng để mã hóa toàn bộ phiên SSH. Thay vào đó, nó chỉ được sử dụng trong quá trình trao đổi thuật toán của khóa của symmetric encryption. trước khi bắt đầu một phiên giao dịch an toàn, cả 2 đồng ý tạo ra một cặp public-private key tạm, chia sẽ private keys để tạo một khóa secret key chung.

Khi kết nối symmetrict an toàn đã được thiết lập, server sử dụng public key của client để tạo và challenge và truyền nó tới client để chứng thực. Nếu client có thể giải mã tin nhắn, có nghĩa là nó đang giữa đúng private key cần thiết cho kết nối. Phiên giao dịch SSH bắt đầu.

Hashing

Hashing một chiều là một dạng mã hóa khác sử dụng trong Secure Shell Connections. Hash một chiều khác với cả 2 phương thức mã hóa trên ở chỗ nó không được sinh ra để giải mã. Chúng tạo ra một giá trị duy nhất với độ dài nhất định cho mỗi lần nhập liệu mà không có hướng nào khác để khai thác. Điều này khiến nó dường như không thể quay ngược lại giải mã.



Rất dễ để tạo một cryptographic hash từ một lần input, nhưng không thể tạo ra lần input đó từ một hash. Có nghĩa là nếu client giữ đúng input đó, client có thể tạo ra một crypto-graphic hash giống như vậy và so sánh nó với giá trị ở đầu bên kia để xác định cả 2 bên nhập giống input.

SSH sử dụng hashes để xác nhận tính xác thực của tin nhắn. Nó được thực hiện bởi HMACs, hoặc **H**ash-based **M**essage **A**uthentication **C**odes. Việc này đảm bảo lệnh không bị giả mạo bởi bất kỳ phương thức nào.

Trong khi thuật toán symmetrical encryption được chọn, một thuật toán xác thực tin nhắn phù hợp cũng được chọn. Nó hoạt động tương tự việc cipher được chọn như thế nào, như bên trên mình đã giải thích trong phần symmetric encryption.

Mỗi tin nhắn được truyền đi phải chứa MAC, được tính bởi symmetric key, packet sequence number, và nội dung tin nhắn. Nó truyền ra ngoài một gói dữ liệu được mã hóa symmetric như là một phần của communication packet.

SSH xử lý như thế nào với những kỹ thuật này

Sau khi bạn đã biết SSH là gì và biết các loại mã hóa, chúng ta đi tiếp về việc nó hoạt động như thế nào. SSH hoạt động bằng mô hình client-server cho phép chứng thực an toàn giữa 2 máy từ xa và mã hóa dữ liệu được truyền giữa chúng.

SSH vận hành trên TCP port 22 mặc định (có thể được thay đổi nếu cần). Host (server) nghe port 22 (hoặc bất kỳ port nào SSH được gán vào) cho nhưng kết nối tới. Nó sẽ thiết lập kết nối an toàn khi chứng thực giữa client và môi trường shell đang mở thành công.

Client phải bắt đầu kết nối SSH bằng cách tạo ra TCP handshake với server, đảm bảo có thể thiết lập kết nối symmetric, xác thực thông tin của server có khớp dữ liệu cũ không (thông thường được trong RSA key store file), và so sánh thông tin đăng nhập của user kết nối để xác thực đúng kết nối.

Có 2 giai đoạn để thiết lập kết nối: trước tiên cả 2 bên đồng ý chuẩn mã hóa để bảo vệ giao tiếp trong tương, thừ 2, user phải được xác thực. Nếu thông tin đăng nhập khớp, user có quyền truy cập.

Session Encryption Negotiation

Khi client cố kết nối tới server qua TCP, server sẽ trình ra encrytpion protocal và những phiên bản liên quan nó hỗ trợ. Nếu client cũng có protocol tương ứng và phiên bản đúng như vậy, một thỏa thuật sẽ được đặt ra và kết nối bát đầu tiếp nhận protocol. Server cũng sử dụng một symmetric public key mà client có thể dùng để xác thực tính chính xác của server.

Khi đã được thiết lập, cả 2 bên sử dụng một thuật toán được biết là Diffie-Hellman Key Exchange Algorithm để tạo symmetrical key. Thuật toán này cho phép cả client và server có cùng một key chung được dùng để mã hóa toàn bộ liên lạc sau này.

Đây là cách thuật oán hoạt động về cơ bản:

1. Cả client và server đồng ý dựa trên một số nguyên lớn, dĩ nhiên là không có bất kỳ tính chất chung nào. Số này được gọi là **seed value**.
2. Tiếp theo, cả 2 bên đồng ý một cách mã hóa được tạo ra từ seed value bằng một dạng thuật toán nhất định. Những cơ chế này là nguồn tạo mã hóa, hoạt đông lớn trên seed value. Ví dụ như là generator llà AES (Advanced Encryption Standard).

3. Cả 2 bên độc lập tạo một số khác. Nó được dùng như là một private key bí mật cho tương tác.
4. Key private mới tạo này, với số chung và thuật toán mã hóa ở trên (AES) được dùng để tạo ra một key public được phân phối cho máy còn lại.
5. 2 bên sau đó sử dụng private key của chính nó, public key của máy còn lại và số nguyên ban đầu để tạo ra một key chung cuối cùng. Key này độc lập được tính toán bởi cả 2 máy nhưng sẽ tạo ra một key mã hóa giống nhau trên cả 2.
6. Bây giờ cả 2 đã có shared key, chúng có thể tạo mã hóa symmetric cho cả phiên SSH. Một key chung được sử dung để mã hóa và giải mã tin nhắn (đọc lại mục: symmetrical encryption).

Bây giờ phiên giao dịch được mã hóa symmetric đã được thiết lập, chứng thực cho user sẽ được tiến hành.

Chứng thực người dùng

Bước cuối là khi user được cấp quyền truy cập vào server xác thực chính thông tin đang nhập đó. Để làm vậy, hầu hết SSH user sử dụng mật khẩu. Người dùng được hỏi để nhập username, tiếp theo là mật khẩu. Những thông tin đăng nhập này được chuyển an toàn qua một đường hầm bảo mật symmetric, vì vậy không có cách nào chúng bị lấy cắp từ bên thứ 3.

Mặc dù mật khẩu đã được mã hóa, chúng tôi vẫn không khuyên sử dụng mật khẩu để thiết lập kết nối. Lý do là vì bằng thủ thuật tấn công brute fore, mật khẩu mặc định hoặc dễ đoán có thể được lần ra và bạn sẽ bị chiếm quyền tài khoản. Vì vậy, cách tốt nhất là sử dụng SSH Key Pairs.

Đây là một bộ khóa asymmetric được dùng để chứng thực thành viên mà không đòi hỏi phải nhập mật khẩu.

Kết luận

Hiểu rõ về SSH là gì và làm thế nào SSH hoạt động được có thể giúp bạn hiểu thêm về công nghệ bảo mật. Hầu hết mọi người tưởng quá trình này là phức tạp và không tài nào hiểu nổi, nhưng nó đơn giản hơn mọi người nghĩ nhiều. Nếu bạn không biết một máy tính mất bao lâu để tính ra một hash và chứng thực user, thì trên thực tế nó chỉ mất ít hơn một giây. Lượng thời gian trên internet chủ yếu là do việc truyền dữ liệu từ xa.

Hy vọng với bài hướng dẫn SSH này, chúng tôi đã giúp bạn có cái nhìn khác về công nghệ và nó là thành tố chính để bạn tạo một hệ thống mạnh mẽ và bảo mật. Cũng vì lẽ đó, bạn đã hiểu vì sao Telnet đã là quá khứ và vì sao SSH đã chiếm lấy mọi chỗ đứng của nó.

Để biết thêm về thủ thuật Linux, hãy xem qua khu vực hướng dẫn cho VPS tutorials của chúng tôi

Về tác giả