Hướng Dẫn Thực Hành VPN

Task 1 tạo kết nối host-to-host sử dụng đường hầm TUN/TAP

Công nghệ cho phép cho

TLS/ SSLVPNsisTUN / TAP, hiện đang triển khai rộng dãi các hệ điều hành hiện đại. TUN và TAP là hệ điều hành điều lõi để điều khiển mạng ảo; họ triển khai thiết bị mạng được hỗ trợ hoàn toàn bằng phần mềm TAP (như trong mạng lưới) mô phỏng một kết nối Ethernet và chạy trên các gói lớp 2 như khung Ethernet TUN (như trong networkTUNnel) mô phỏng một thiết bị lớp mạng và nó hoạt động với các gói lớp-3 như các gói IP. Với TUN / TAP, chúng ta có thể tạo mạng ảo trong các địa chỉ. Một chương trình không gian người dùng thường được gắn vào giao diện mạng ảo TUN / TAP. Các gói được gửi bởi một hệ điều hành thông qua giao diện mạng TUN / TAP được gửi đến chương trình không gian người dùng. Mặt khác, các gói được gửi bởi chương trình thông qua giao diện mạng TUN / TAP được đưa vào ngăn xếp mạng của hệ điều hành, với hệ điều hành, có vẻ như các gói tin đến từ một nguồn bên ngoài thông qua giao diện mạng ảo.

Khi một chương trình được gắn vào một giao diện TUN / TAP, các gói IP mà máy tính gửi đến giao diện này sẽ được đưa vào chương trình; mặt khác, các gói IP mà chương trình gửi đến giao diện sẽ được đưa vào máy tính, như thể chúng đến từ bên ngoài thông qua giao diện mạng ảo này. Chương trình có thể sử dụng các cuộc gọi hệ thống read () và write () tiêu chuẩn để nhận các gói tin từ hoặc gửi các gói tin tới giao diện ảo.

Davide Brini đã viết một bài hướng dẫn tuyệt vời về cách sử dụng TUN / TAP để tạo một đường hầm giữa hai máy. URL của hướng dẫn là http://waldner.netsons.org/d2-tuntap.php. Hướng dẫn này cung cấp một chương trình gọi là simpletun, kết nối hai máy tính bằng cách sử dụng kỹ thuật đường hầm TUN. Học sinh nên đọc hướng dẫn này. Khi bạn biên dịch mã mẫu từ hướng dẫn, nếu bạn thấy thông báo lỗi liên quan đến linux / if.h, hãy thử thay đổi "<linux / if.h>" thành "<net / if.h>" trong câu lệnh hướng dẫn.

Để thuận tiện cho phòng thí nghiệm này, chúng tôi đã sửa đổi chương trình đơn giản của Brini và liên kết mã trong trang web của phòng thí nghiệm. Học sinh chỉ cần tải xuống chương trình C này và chạy lệnh sau để biên dịch nó. Chúng tôi sẽ sử dụng simpletun để tạo đường hầm trong phòng thí nghiệm này: $ gcc -o simpletun simpletun.c

Tạo đường hầm host to host : Quy trình sau đây cho thấy cách tạo đường hầm lưu trữ trên máy chủ bằng chương trình simpletun. Chương trình simpletun có thể chạy như một máy khách và máy chủ. Khi nó đang chạy với -s-ag, nó hoạt động như một máy chủ; khi nó đang chạy với -c fl ag, nó hoạt động như một máy khách.

1. Khởi động hai máy ảo. Đối với nhiệm vụ này, chúng tôi sẽ khởi chạy hai máy ảo này trên cùng một máy chủ. Địa chỉ IP cho hai máy lần lượt là 192.168.10.5 và 192.168.20.5 (bạn có thể chọn bất kỳ địa chỉ IP nào bạn muốn). Xem sự xác định trong Hình 1

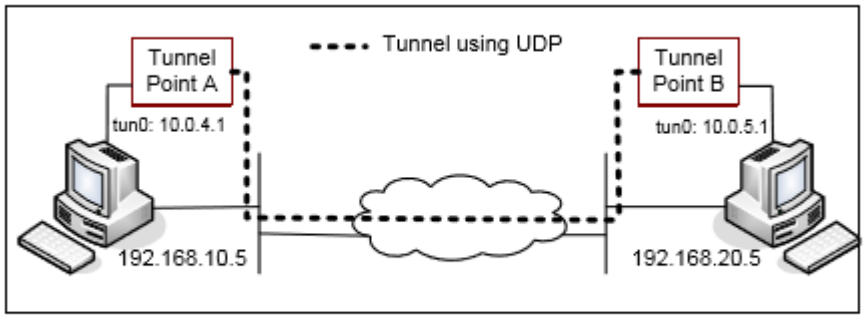

Figure 1: Host-to-Host Tunnel

2. Tunnel Point A: chúng ta sử dụng Tunnel Point A làm phía máy chủ của đường hầm. Điểm A trên máy 192.168.10.5 (xem Hình 1). Cần lưu ý rằng khái niệm máy khách / máy chủ chỉ có ý nghĩa khi thiết lập kết nối giữa hai đầu. Khi đường hầm được thiết lập, không có sự khác biệt giữa máy khách và máy chủ; chúng chỉ đơn giản là hai đầu của một đường hầm. Chúng ta chạy lệnh sau đây (-d-ag yêu cầu chương trình in ra thông tin gỡ lỗi):

Trên máy 192.168.10.5:
./simpletun -i tun0 -s -d

Sau bước trên, máy tính của bạn bây giờ có nhiều giao diện mạng, một là giao diện thẻ Ethernet riêng của nó, và một là giao diện mạng ảo được gọi là tun0. Giao diện mới này chưa được cấu hình, vì vậy chúng ta cần phải cấu hình nó bằng cách gán một địa chỉ IP. Chúng tôi sử dụng địa chỉ IP từ không gian địa chỉ IP dành riêng (10.0.0.0/8). Cần lưu ý rằng lệnh trên sẽ chặn và chờ
các kết nối, vì vậy, chúng ta cần tìm một cửa sổ khác để cấu hình giao diện tun0. Chạy các lệnh sau đây (lệnh đầu tiên sẽ gán một địa chỉ IP cho giao diện "tun0", và lệnh thứ hai sẽ đưa lên giao diện):

Trên máy 192.168.10.5:
ip addr thêm 10.0.4.1/24 dev tun0
ifconfig tun0 up

3. Tunnel Point B: chúng ta sử dụng Tunnel Point B như phía client của đường hầm. Điểm B là trên máy 192.168.20.5 (xemFigure1). Chúng tôi chạy lệnh sau trên máy này (Lệnh đầu tiên sẽ kết nối với chương trình máy chủ chạy trên 192.168.10.5, đó là máy chạy Tunnel Point A. Lệnh này cũng sẽ chặn, vì vậy chúng ta cần tìm một cửa sổ khác cho lệnh thứ hai và thứ ba):
On Machine 192.168.20.5:

./simpletun -i tun0 -c 192.168.10.5 –d
ip addr add 10.0.5.1/24 dev tun0
ifconfig tun0 up

4. Đường dẫn định tuyến: Sau hai bước trên, đường hầm sẽ được thiết lập. Trước khi chúng ta có thể sử dụng đường hầm, chúng ta cần phải thiết lập đường dẫn định tuyến trên cả hai máy để hướng luồng đi ra dự định thông qua đường hầm. Mục bảng định tuyến sau đây hướng tất cả các gói đến mạng 10.0.5.0/24 (mạng 10.0.4.0/24 cho lệnh thứ hai) thông qua phần tử tunface tun0, từ đó gói tin sẽ được chuyển qua đường hầm.

On Machine 192.168.10.5:
route add -net 10.0.5.0 netmask 255.255.255.0 dev tun0
On Machine 192.168.20.5:
route add -net 10.0.4.0 netmask 255.255.255.0 dev tun0
5. UsingTunnel: Bây giờ chúng ta có thể truy cập 10.0.5.1 từ 192.168.10.5 (và tương tự access10.0.4.1 từ 192.168.20.5). Chúng ta có thể kiểm tra đường hầm bằng cách sử dụng ping và ssh (lưu ý: đừng quên khởi động máy chủ ssh đầu tiên):

On Machine 192.168.10.5:
$ ping 10.0.5.1
$ ssh 10.0.5.1
On Machine 192.168.20.5:
$ ping 10.0.4.1
$ ssh 10.0.4.1

UDPTunnel: Kết nối được sử dụng trong chương trình simpletun là kết nối TCP, đường hầm VPN của bạn cần sử dụng UDP. Vì vậy, bạn cần phải sửa đổi simpletun và biến đường hầm TCP thành đường hầm UDP. Bạn cần phải suy nghĩ về lý do tại sao nó là tốt hơn để sử dụng UDP trong đường hầm, thay vì TCP. Vui lòng viết câu trả lời của bạn trong báo cáo phòng thí nghiệm.

Task 2 tạo đường hầm host to gateway

Bây giờ bạn đã thành công trong việc thiết lập đường hầm trên hai máy ảo trong một máy chủ duy nhất, bạn nên thiết lập một đường hầm tương tự trên hai máy ảo trên hai máy chủ khác nhau. Bạn có thể sử dụng cổng chuyển tiếp cho mục đích này. Vui lòng xem hướng dẫn về chuyển tiếp cổng. Trong nhiệm vụ này, bạn cần tạo một đường hầm giữa máy tính và cổng, cho phép máy tính truy cập vào mạng riêng được kết nối với cổng. Để chứng minh điều này, bạn cần hai máy tính vật lý. Trên một máy tính, bạn chạy một vài máy ảo trong máy tính để thiết lập cổng và mạng riêng. Sau đó, bạn sử dụng máy ảo trong máy tính khác để liên lạc với máy chủ trên mạng riêng. Vui lòng tham khảo phần hướng dẫn để xem cách thiết lập cổng và mạng riêng. Vì bạn cần hai máy tính vật lý, bạn có thể hợp tác với một sinh viên khác nếu bạn chỉ có một máy tính. Tuy nhiên, bạn phải làm công việc của bạn một cách độc lập. Hợp tác chỉ dành cho mục đích trình diễn.

Task 3 tạo đường hầm gateway to gateway

Trong nhiệm vụ này, bạn cần phải đi thêm một bước để thiết lập một đường hầm giữa hai cổng của các mạng riêng khác nhau. Với đường hầm này, bất kỳ máy chủ nào từ một mạng riêng có thể liên lạc với các máy chủ trên mạng riêng khác bằng đường hầm. Thiết lập cho một đường hầm gateway-to-gateway như được mô tả trong Hình 2.

Task 4 tạo 1 mạng riêng ảo (VPN)

Tại thời điểm này, bạn đã học được cách tạo một đường hầm mạng. Bây giờ, nếu bạn có thể bảo vệ đường hầm này, về cơ bản bạn sẽ nhận được một VPN. Đây là những gì chúng ta sẽ đạt được trong nhiệm vụ này. Để đảm bảo đường hầm này, chúng ta cần đạt được hai mục tiêu, tính quyết định và tính toàn vẹn. Tính xác thực được thực hiện bằng cách sử

dụng mã hóa, i.e. tức là nội dung đi qua đường hầm được mã hóa. Tại thời điểm này, bạn đã học được cách tạo một đường hầm mạng. Bây giờ, nếu bạn có thể bảo vệ đường hầm này, về cơ bản bạn sẽ nhận được một VPN. Đây là những gì chúng ta sẽ đạt được trong nhiệm vụ này. Để đảm bảo đường hầm này, chúng ta cần đạt được hai mục tiêu, tính quyết định và tính toàn vẹn. Tính xác thực được thực hiện bằng cách sử dụng mã hóa, tức là nội dung đi qua đường hầm được mã hóa. Một phần mềm VPN thực thường hỗ trợ một số thuật toán mã hóa khác nhau. MiniVPN trong phòng thí nghiệm này, chúng tôi chỉ cần hỗ trợ thuật toán Sencryption AE và chúng tôi sử dụng chế độ Chấm khối mã hóa (CBC). Mục tiêu tính toàn vẹn đảm bảo rằng không ai có thể làm xáo trộn đường đi trong đường hầm hoặc khởi động một cuộc tấn công phát lại. Tính toàn vẹn có thể đạt được bằng các phương pháp khác nhau. Trong phòng thí nghiệm này, chúng tôi chỉ cần hỗ trợ phương thức Mã xác thực thư (MAC). Thuật toán mã hóa AES và thuật toán HMAC-SHA256

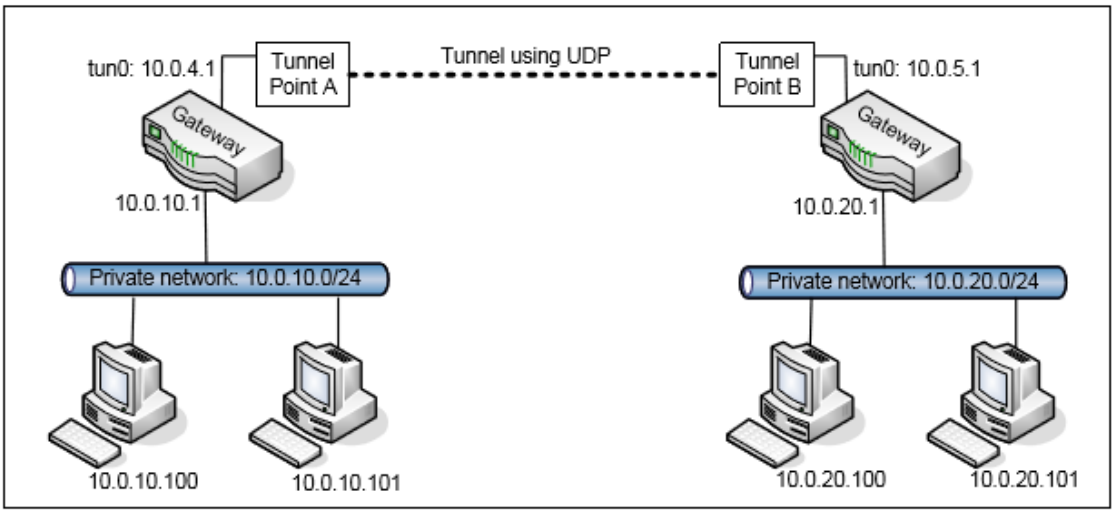

Figure 2: Gateway to Gateway

đều được triển khai trong thư viện OpenSSL. Có rất nhiều tài liệu trực tuyến giải thích cách sử dụng thư viện mã hóa của OpenSSL. Cả mã hóa và MAC đều cần một khóa bí mật. Mặc dù các khóa có thể khác nhau cho mã hóa và MAC, vì mục đích đơn giản, chúng tôi giả định rằng cùng một khóa được sử dụng. Khóa này phải được cả hai bên của VPN đồng ý. Đối với nhiệm vụ này, chúng tôi giả định rằng khóa đã được cung cấp. Đồng ý về chìa khóa sẽ được thực hiện trong nhiệm vụ tiếp theo. Để mã hóa, máy khách và máy chủ cũng cần phải đồng ý với một Vector ban đầu (IV). Vì mục đích bảo mật, bạn không nên mã hóa mã IV trong mã của mình. IV nên được tạo ngẫu nhiên cho mỗi đường hầm VPN. Đồng ý về IV cũng sẽ được thực hiện trong nhiệm vụ tiếp theo.

Task 5 xác thực và trao đổi khóa

Trước khi VPN được thiết lập, máy khách VPN phải xác thực máy chủ VPN, đảm bảo rằng máy chủ không phải là một máy chủ lừa đảo. Mặt khác, máy chủ VPN phải xác thực máy khách (tức là người dùng), đảm bảo rằng người dùng có quyền tạo đường hầm VPN như vậy. Sau khi xác thực được thực hiện, máy khách và máy chủ sẽ đồng ý với khóa phiên cho đường hầm VPN. Khóa phiên này chỉ được biết với máy khách và máy chủ. Quá trình phát sinh khóa phiên này được gọi là trao đổi khóa.

Bước 1: Xác thực máy chủ VPN Một cách điển hình để xác thực máy chủ là sử dụng chứng chỉ khóa công khai. Máy chủ VPN cần có một chứng nhận khóa công khai từ một Tổ chức cấp chứng chỉ (CA), chẳng hạn như Verisign. Khi máy khách kết nối với máy chủ

VPN, máy chủ sẽ sử dụng chứng chỉ máy chủ đó là máy chủ dự định. Giao thức HTTPS trong Web sử dụng một cách tương tự để xác thực máy chủ web, đảm bảo rằng bạn đang nói chuyện với một máy chủ web dự định, không phải là một giao diện giả mạo. Sau bước này, bạn nên có một ý tưởng rõ ràng về cách xác thực trong HTTPS hoạt động. Trong phòng thí nghiệm này, MiniVPN nên sử dụng phương thức như vậy để xác thực máy chủ VPN. Bạn có thể thực hiện một giao thức xác thực (chẳng hạn như SSL) từ đầu, sử dụng các thư viện mã hóa trong OpenSSL để xác minh các chứng chỉ. Hoặc bạn có thể sử dụng các chức năng SSL của OpenSSL để trực tiếp tạo kết nối SSL giữa máy khách và máy chủ, trong trường hợp đó, việc xác nhận các chứng chỉ sẽ được thực hiện tự động bởi Chức năng SSL. Bạn có thể tìm thấy các hướng dẫn về cách tạo kết nối như vậy trong phần tiếp theo.

Bước 2: Xác thực Trình khách VPN (tức là Người dùng) Có hai cách phổ biến để xác thực người dùng. Một là sử dụng chứng chỉ khóa công khai. Cụ thể, người dùng cần có giấy chứng nhận khóa công khai của riêng họ. Khi họ cố gắng tạo ra một VPN với máy chủ, họ cần phải gửi Cates fi chứng chỉ của họ đến máy chủ, mà sẽ xác minh xem họ có quyền cho một VPN như vậy. Chức năng SSL của OpenSSL cũng hỗ trợ tùy chọn này nếu bạn chỉ định rằng xác thực ứng dụng khách là bắt buộc. Vì người dùng thường không có Cates fi chứng chỉ khóa công khai của họ, một cách phổ biến hơn để xác thực người dùng là sử dụng tên người dùng truyền thống và cách tiếp cận mật khẩu. Cụ thể, sau khi máy khách và máy chủ đã thiết lập kết nối TCP an toàn giữa mình, máy chủ có thể yêu cầu máy khách nhập tên người dùng và mật khẩu, sau đó máy chủ quyết định có cho phép người dùng tiến hành tùy thuộc vào tên người dùng hay không và mật khẩu khớp với thông tin trong cơ sở dữ liệu người dùng của máy chủ. Trong phòng thí nghiệm này, bạn có thể chọn một trong số chúng để triển khai

Bước 3: Trao đổi khóa. Nếu bạn sử dụng các chức năng SSL của OpenSSL, sau khi xác thực, một kênh bảo mật sẽ được thiết lập tự động (bởi các chức năng OpenSSL). Tuy nhiên, chúng tôi sẽ không sử dụng kết nối TCP này cho đường hầm của chúng tôi, vì đường hầm VPN của chúng tôi sử dụng UDP. Do đó, chúng tôi sẽ xử lý kết nối TCP này làm kênh điều khiển giữa máy khách và máy chủ. Trong kênh điều khiển này, máy khách và máy chủ sẽ đồng ý với khóa phiên cho kênh dữ liệu (i.e..tức là đường hầm VPN). Họ cũng có thể sử dụng kênh điều khiển cho các chức năng khác, chẳng hạn như cập nhật khoá phiên, trao đổi Vector ban đầu (IV), chấm dứt đường hầm VPN, v.v. Ở cuối bước này, bạn sẽ có thể sử dụng khóa phiên để bảo vệ đường hầm. Nói cách khác, bạn sẽ có thể kiểm tra Nhiệm vụ 4 và Nhiệm vụ 5 cùng nhau

Bước 4: Tái cấu hình động. Bạn nên thực hiện một số lệnh ở phía máy khách, để cho phép máy khách thực hiện các thao tác sau:
• Thay đổi khoá phiên trên đầu máy khách và thông báo cho máy chủ thực hiện thay đổi tương tự. • Thay đổi IV trên đầu máy khách và thông báo cho máy chủ thực hiện thay đổi tương tự.
• Phá vỡ đường hầm VPN hiện tại. Máy chủ cần được thông báo, vì vậy nó có thể giải phóng tài nguyên tương ứng.
Bạn được khuyến khích triển khai các tính năng khác cho MiniVPN của mình. Điểm thưởng sẽ được trao cho các tính năng hữu ích. Tuy nhiên, bất kỳ tính năng nào bạn thêm vào triển khai, bạn cần đảm bảo rằng bảo mật không bị xâm phạm.

Task 6 hỗ trợ nhiều đường hầm VPN

Trong thế giới thực, một máy chủ VPN thường hỗ trợ nhiều đường hầm VPN. Cụ thể, máy chủ VPN cho phép nhiều hơn một máy khách kết nối với nó đồng thời; mỗi khách hàng có đường hầm VPN riêng của mình với máy chủ và các khóa phiên được sử dụng trong các đường hầm khác nhau nên có sự khác biệt. Máy chủ VPN của bạn sẽ có thể hỗ trợ nhiều máy khách. Bạn không thể giả định rằng chỉ có một đường hầm và một khóa phiên. Khi một gói tin đến máy chủ VPN thông qua một đường hầm VPN, máy chủ cần phải tìm ra từ đường hầm VPN mà gói tin đến từ đó. Nếu không có thông tin này, máy chủ không thể biết khóa giải mã nào (andIV) nên sử dụngtrong bộ mã hóa, sử dụng khóa sai sẽ làm cho gói tin bị loại bỏ, vì HMAC sẽ không khớp. Bạn có thể xem giao thức IPSec và nghĩ về cách IPSec có thể hỗ trợ nhiều đường hầm. Bạn có thể sử dụng ý tưởng tương tự