|  |
| --- |
| HỌC VIỆN KỸ THUẬT MẬT MÃ  **KHOA AN TOÀN THÔNG TIN**  Logo HvKTMM  ĐỒ ÁN MÔN HỌC  **KỸ THUẬT LẬP TRÌNH**  **ĐỀ TÀI**  **TÌM HIỂU SSH BOTNET**  **Ngành:** An toàn thông tin  *Sinh viên thực hiện:*  **Nguyễn Văn Hải –AT170415**  **Lâm Tăng Thành – AT170346**  **Đàm Ngọc Nam – AT170435**  **Trần Gia Hiển – AT170317**  *Giảng viên hướng dẫn:* **ThS. Bùi Việt Thắng**  Hà Nội, 4 - 2023 |

**MỤC LỤC**

[LỜI NÓI ĐẦU 4](#_Toc132395905)

[DANH MỤC CÁC CHỮ VIẾT TẮT 6](#_Toc132395906)

[DANH MỤC CÁC BẢNG 7](#_Toc132395907)

[CHƯƠNG I . TÌM HIỂU VỀ DDOS 10](#_Toc132395908)

[1.1. Khái niệm về DDOS 10](#_Toc132395909)

[1.2. Phân loại 11](#_Toc132395910)

[1.3. Cách phòng chống DDOS 14](#_Toc132395911)

[CHƯƠNG II. TỔNG QUAN VỀ SSH BOTNET 16](#_Toc132395912)

[2.1. Tổng quan SSH 16](#_Toc132395913)

[***2.1.1. Khái niệm*** 16](#_Toc132395914)

[***2.1.2. Phương thức hoạt động*** 16](#_Toc132395915)

[***2.1.3. Quy trình hoạt động của SSH*** 17](#_Toc132395916)

[***2.1.4. Chức năng chính*** 18](#_Toc132395917)

[***2.1.5. SSH sử dụng khi nào*** 19](#_Toc132395918)

[2.2. Giới thiệu Bonet 20](#_Toc132395919)

[***2.2.1. Khái niệm*** 20](#_Toc132395920)

[***2.2.2. Đặc điểm*** 20](#_Toc132395921)

[***2.2.3. Mục đích của Botnet*** 21](#_Toc132395922)

[***2.2.4. Phân loại Botnet*** 21](#_Toc132395923)

[***2.2.5. Cấu trúc của botnet*** 23](#_Toc132395924)

[***2.2.6. Các loại tấn công Botnet*** 26](#_Toc132395925)

[***2.2.7. Phòng chống Botnet*** 30](#_Toc132395926)

[2.3. SSH Bonet 30](#_Toc132395927)

[***2.3.1. Tìm hiểu về SSH Botnet*** 30](#_Toc132395928)

[***2.3.2. Cấu trúc SSH Botnet*** 31](#_Toc132395929)

[***2.3.3. Cách phòng chống SSH Botnet*** 31](#_Toc132395930)

[***2.3.4. Tìm hiểu module Paramiko trong Python*** 32](#_Toc132395931)

[***2.3.5. Dictionary mật khẩu SSH dùng Paramiko*** 32](#_Toc132395932)

[***2.3.6. Khai thác SSH thông qua khóa bí mật yếu*** 33](#_Toc132395933)

[***2.3.7. Dựng SSH Botnet với Python*** 34](#_Toc132395934)

[CHƯƠNG III. TRIỂN KHAI THỰC NGHIỆM 35](#_Toc132395935)

[3.1. Xây dựng class SSHBotnet 35](#_Toc132395936)

[3.2. Xây dựng class BotMaster 40](#_Toc132395937)

[3.3. Tấn công Dictionary tìm mật khẩu SSH 43](#_Toc132395938)

[3.4. Điều khiển nhiều botnet và thực hiện DDOS 48](#_Toc132395939)

[CHƯƠNG IV. KẾT LUẬN 58](#_Toc132395940)

[BẢNG PHÂN CÔNG CÔNG VIỆC 59](#_Toc132395941)

**LỜI NÓI ĐẦU**

Với sự phát triển vượt bậc của khoa học kỹ thuật và sự mở rộng của mạng máy tính, Internet đã mang lại nhiều lợi ích và thuận lợi cho xã hội. Tuy nhiên, cùng với những lợi ích đó là những mối đe dọa ngày càng tăng về an ninh mạng, bao gồm vi-rút, thư rác, tin tặc và các đối tượng tấn công khác…. Việt Nam hiện đứng đầu trong danh sách các quốc gia bị tấn công và đứng cuối cùng ở khu vực Đông Nam Á về chỉ số an ninh toàn cầu. Trong số các mối đe dọa này, Botnet và các mã độc được coi là mối đe dọa chính.

Vì vậy nhóm em đã quyết định nghiên cứu về SSH Botnet để tìm hiểu các khái niệm cơ bản, phương thức hoạt động, sự nguy hiểm của nó và cách phòng tránh các cuộc tấn công của SSH Botnet. Nghiên cứu này bao gồm các thử nghiệm như tương tác với SSH thông qua Paramiko, Dictionary mật khẩu SSH với Paramiko, khai thác SSH thông qua khóa bí mật yếu và xây dựng SSH với Python…(Paramiko thuộc thư viện ngôn ngữ Python ,hữu ích trong việc triển khai SSH Botnet)

**Mục tiêu của đồ án :**

Nghiên cứu, tìm hiểu khái niệm, tổng quan lý thuyết phương thức hoạt động của SSH Botnet (Botnet là gì, SSH là gì, tại sao SSH đã chiếm mọi vị trí của Telnet, các cuộc tấn công Botnet, phương pháp phóng tránh tấn công Botnet)

Qua đó có thêm kiến thức về lĩnh vực công nghệ hiểu hơn về SSH Botnet, biết được các tấn công SSH Botnet và các phòng tránh hiệu quả nhất.

**GIỚI THIỆU**

Báo cáo nhóm em gồm các nội dung chính:

**CHƯƠNG I. TÌM HIỂU VỀ DDOS**

1.1. Khái niệm DDOS

1.2. Phân loại

1.3. Cách phòng chống DDOS

**CHƯƠNG II. TỔNG QUAN VỀ SSH BOTNET**

2.1. Tổng quan SSH

2.2. Giới thiệu Botnet

2.3. SSH BotNet

**CHƯƠNG III. TRIỂN KHAI THỰC NGHIỆM**

3.1. Xây dựng class SSHBotnet

3.2. Xây dựng class BotMaster

3.3. Tấn công Dictionary tìm mật khẩu SSH

3.4. Điều khiển nhiều botnet và thực hiện DDOS

**CHƯƠNG IV. KẾT LUẬN**

Qua 4 chương này chúng em sẽ khái quát những vấn đề chung nhất liên quan đến SSH Botnet. Lý thuyết, đặc điểm, cấu tạo, phân loại, nguyên lý hoạt động, các phương thức tấn công và phòng thủ đối với SSH Botnet.

Tuy đã cố gắng tìm kiếm từ nhiều nguồn thông tin về đề tài này, nhưng vẫn còn những thiếu sót không thể tránh khỏi trong quá trình thực hiện đề tài. Rất mong thầy cô và các bạn đọc có thể đưa ra lời góp ý đề nhóm chúng em hoàn thiện tài liệu hơn.

Xin chân thành cảm ơn thầy và các bạn!

**DANH MỤC CÁC CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **DDOS** | **Distributed Denial of Service** |
| **IoT** | Internet of Things |
| **SSH** | Secure Shell |
| **TCP/IP** | Transmisson Control Protocol/Internet Protocol |
| **IRC** | Internet Relay Chat |
| **P2P** | Peer-to-Peer |
| **HTTP** | Hypertext Transfer Protocol |
| **FTP** | File Transfer Protocol |
| **C&C** | Command and Control |
| **UDP** | User Datagram Protocol |
| **SYN** | synchronize |
| **ICMP** | Internet Control Message Protocol |
| **NTP** | Network Time Protocol |
| **OSI** | Open Systems Interconnection |
| **POST** | Power-On Self-Test |
| **URL** | Uniform Resource Locator |
| **APDoS** | Advanced Persistent DoS |
| **IDS/IPS** | Intrusion Detection System/Intrusion Prevention System |

**DANH MỤC CÁC BẢNG**

|  |  |
| --- | --- |
| **Hình 1** | Giao diện Window SSH điển hình |
| **Hình 2** | Mô tả cơ chế hoạt động của SH |
| **Hình 3** | Dòng lệnh nhắc nhở |
| **Hình 4** | Cấu trúc Botnet |
| **Hình 5** | Mô hình mạng client – server |
| **Hình 6** | Mô hình mạng peer-to-peer (ngang hàng) |
| **Hình 7** | Các loại tấn công Botnet |
| **Hình 8** | Mô tả Spamming |
| **Hình 9** | Tấn công đơn giản Dictionary sử dụng Paramiko |

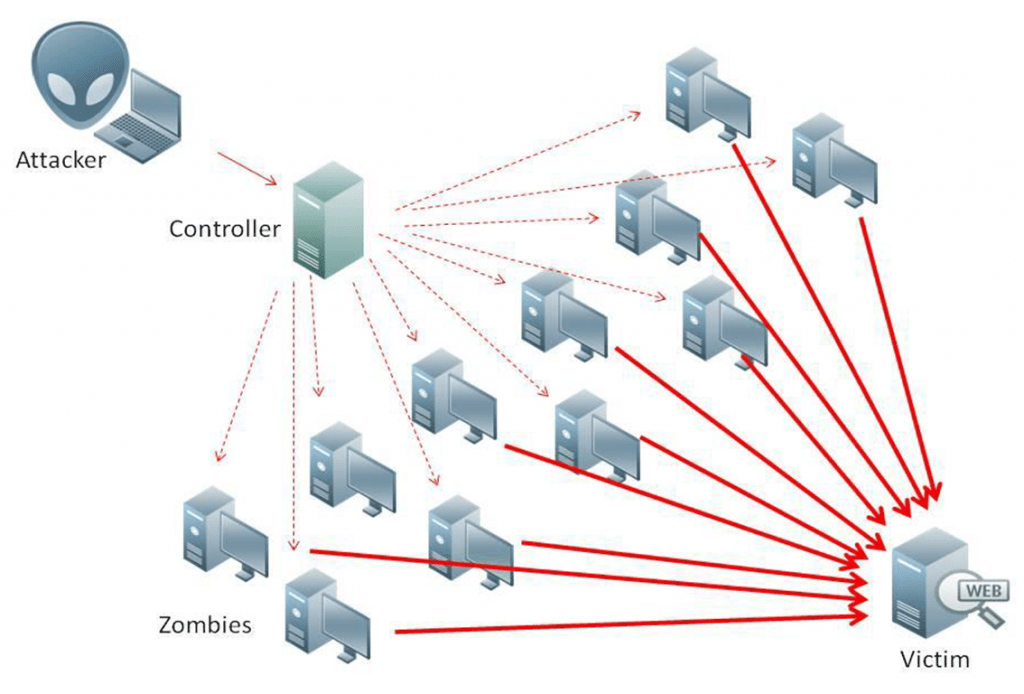
**TÌM HIỂU VỀ**

**SSH BOTNET**

# CHƯƠNG I . TÌM HIỂU VỀ DDOS

**1.1. Khái niệm về DDOS**

DDoS (Distributed Denial of Service) là một cuộc tấn công mạng nhằm vào một trang web, máy chủ, dịch vụ hoặc mạng bị tắt hoặc làm chậm bằng cách tạo ra một lượng lớn yêu cầu gửi tới mục tiêu. Kẻ tấn công thực hiện cuộc tấn công bằng cách sử dụng nhiều hệ thống máy tính để xâm nhập, mã hóa độc hại hoặc phần mềm độc hại để tạo ra lưu lượng truy cập giả mạo đến mục tiêu tấn công. Các thiết bị được khai thác bao gồm máy tính và các thiết bị có kết nối mạng khác, ví dụ như thiết bị IoT (Internet of Things). Cuộc tấn công DDoS có thể làm gián đoạn hoạt động trực tuyến và gây thiệt hại cho các tổ chức hoặc cá nhân bị tấn công.



*Hình 1: Mô hình cuộc tấn công DDOS*

Nạn nhân của tấn công DDoS thường là máy chủ, trang web hoặc dịch vụ của các tổ chức nổi tiếng như ngân hàng, doanh nghiệp thương mại, công ty truyền thông, tổ chức chính phủ, mạng xã hội… Mặc dù các cuộc tấn công này thường ít gây nguy hại đến các dữ liệu, tài sản quan trọng lưu trữ trên hệ thống, nhưng chúng có thể khiến nạn nhân tốn rất nhiều thời gian và tiền bạc để xử lý các hậu quả mà nó đã gây ra với cơ sở hạ tầng và khách hàng ,hay giảm độ uy tin của tổ chức doanh nghiệp .

**1.2. Phân loại**

Mặc dù DDoS cung cấp một chế độ tấn công ít phức tạp hơn các dạng tấn công mạng khác, nhưng chúng đang ngày càng mạnh mẽ và tinh vi hơn.

1. **Có ba loại tấn công cơ bản:**

* **Volume-based:** Sử dụng lưu lượng truy cập cao để làm tràn ngập băng thông mạng.
* **Protocol:** Tập trung vào việc khai thác các tài nguyên máy chủ.
* **Application:** Tập trung vào các ứng dụng web và được xem là loại tấn công tinh vi và nghiêm trọng nhất.

**B) Một vài tấn công từ chối dịch vụ phổ biến:**

* **SYN Flood**

SYN Flood khai thác điểm yếu trong chuỗi kết nối TCP, được gọi là bắt tay ba chiều. Máy chủ sẽ nhận được một thông điệp đồng bộ (SYN) để bắt đầu "bắt tay". Máy chủ nhận tin nhắn bằng cách gửi cờ báo nhận (ACK) tới máy lưu trữ ban đầu, sau đó đóng kết nối. Tuy nhiên, trong một SYN Flood, tin nhắn giả mạo được gửi đi và kết nối không đóng => dịch vụ sập.

* **UDP Flood**

User Datagram Protocol (UDP) là một giao thức mạng không session. Một UDP Flood nhắm đến các cổng ngẫu nhiên trên máy tính hoặc mạng với các gói tin UDP. Máy chủ kiểm tra ứng dụng tại các cổng đó nhưng không tìm thấy ứng dụng nào.

* **HTTP Flood**

HTTP Flood gần giống như các yêu cầu GET hoặc POST hợp pháp được khai thác bởi một [hacker](https://quantrimang.com/nhung-hieu-biet-co-ban-nhat-de-tro-thanh-hacker-phan-1-4650). Nó sử dụng ít băng thông hơn các loại tấn công khác nhưng nó có thể buộc máy chủ sử dụng các nguồn lực tối đa.

* **Ping of Death**

Ping of Death điều khiển các giao thức IP bằng cách gửi những đoạn mã độc đến một hệ thống. Đây là loại DDoS phổ biến cách đây hai thập kỷ nhưng đã không còn hiệu quả vào thời điểm hiện tại.

* **Smurf Attack**

Smurf Attack khai thác giao thức Internet (IP) và ICMP (Internet Control Message Protocol) sử dụng một chương trình phần mềm độc hại gọi là smurf. Nó giả mạo một địa chỉ IP và sử dụng ICMP, sau đó [ping](https://quantrimang.com/ping-la-gi-131855) các địa chỉ IP trên một mạng nhất định.

* **Fraggle Attack**

Fraggle Attack sử dụng một lượng lớn lưu lượng UDP vào mạng phát sóng của router. Nó giống như một cuộc tấn công Smurf, sử dụng UDP nhiều hơn là ICMP.

* **Slowloris**

Slowloris cho phép kẻ tấn công sử dụng nguồn lực tối thiểu trong một cuộc tấn công và các mục tiêu trên máy chủ web. Khi đã kết nối với mục tiêu mong muốn, Slowloris giữ liên kết đó mở càng lâu càng tốt với HTTP tràn ngập. Kiểu tấn công này đã được sử dụng trong một số DDoSing kiểu hacktivist (tấn công vì mục tiêu chính trị) cao cấp, bao gồm cuộc bầu cử tổng thống Iran năm 2009. Việc giảm thiểu ảnh hưởng với loại hình tấn công này là rất khó khăn.

* **Application Level Attacks**

Application Level Attacks khai thác lỗ hổng trong các ứng dụng. Mục tiêu của loại tấn công này không phải là toàn bộ máy chủ, mà là các ứng dụng với những điểm yếu được biết đến.

* **NTP Amplification**

NTPAmplification khai thác các máy chủ NTP (Network Time Protocol), một giao thức được sử dụng để đồng bộ thời gian mạng, làm tràn ngập lưu lượng UDP. Đây là reflection attack bị khuếch đại. Trong reflection attack bất kỳ nào đều sẽ có phản hồi từ máy chủ đến IP giả mạo, khi bị khuếch đại, thì phản hồi từ máy chủ sẽ không còn tương xứng với yêu cầu ban đầu. Vì sử dụng băng thông lớn khi bị DDoS nên loại tấn công này có tính phá hoại và volumne cao.

* **Advanced Persistent DoS (****APDoS)**

Advanced Persistent DoS (APDoS) là một loại tấn công được sử dụng bởi hacker với mong muốn gây ra những thiệt hại nghiêm trọng. Nó sử dụng nhiều kiểu tấn công được đề cập trước đó HTTP Flood, SYN Flood, v.v...) và thường nhắm tấn công theo kiểu gửi hàng triệu yêu cầu/giây. Các cuộc tấn công của APDoS có thể kéo dài hàng tuần, phụ thuộc vào khả năng của hacker để chuyển đổi các chiến thuật bất cứ lúc nào và tạo ra sự đa dạng để tránh các bảo vệ an ninh.

* **Zero-day DDoS Attacks**

Zero-day DDoS Attacks là tên được đặt cho các phương pháp tấn công DDoS mới, khai thác các lỗ hổng chưa được vá.

* **HTTP GET**

HTTP GET là một kiểu tấn công lớp ứng dụng (Application Layer attack), quy mô nhỏ hơn và được nhắm tới những mục tiêu hơn. Application Level Attacks khai thác lỗ hổng trong các ứng dụng. Mục tiêu của loại tấn công này không phải là toàn bộ máy chủ, mà là các ứng dụng với những điểm yếu được biết đến.

Kiểu tấn công này sẽ nhắm vào Lớp thứ 7 trong [mô hình OSI](https://quantrimang.com/kien-thuc-co-ban-ve-mang-phan-17-mo-hinh-osi-43991). Đây là lớp có lưu lượng mạng cao nhất, thay vì hướng vào lớp thứ 3 thường được chọn làm mục tiêu trong các cuộc tấn công Bulk Volumetric. HTTP GET khai thác quy trình trình của một trình duyệt web hoặc ứng dụng HTTP nào đó và yêu cầu một ứng dụng hoặc máy chủ cho mỗi yêu cầu HTTP, đó là GET hoặc POST.

HTTP Flood gần giống như các yêu cầu GET hoặc POST hợp pháp được khai thác bởi một hacker. Nó sử dụng ít băng thông hơn các loại tấn công khác nhưng nó có thể buộc máy chủ sử dụng các nguồn lực tối đa. Rất khó để chống lại kiểu tấn công này vì chúng sử dụng các yêu cầu [URL](https://quantrimang.com/url-la-gi-158090)tiêu chuẩn, thay vì các tập lệnh bị hỏng hoặc khối lượng lớn.

**1.3. Cách phòng chống DDOS**

**A. Dấu hiệu của tấn công DDoS**

Dưới đây là một số dấu hiệu của tấn công DDoS:

Tốc độ truy cập lớn: Một trong những dấu hiệu đầu tiên của tấn công DDoS là sự tăng đột biến trong lưu lượng truy cập đến hệ thống. Điều này có thể dẫn đến sự chậm trễ hoặc tạm dừng hoạt động của hệ thống.

Giảm tốc độ truy cập: Một số dạng tấn công DDoS có thể tạo ra lưu lượng truy cập giảm, thường là do tấn công vào cơ chế phân tán tài nguyên (DDoS amplification attacks), trong đó một số yêu cầu nhỏ được chuyển tiếp đến các thiết bị trung gian và cuối cùng đến hệ thống mục tiêu, tạo ra một lưu lượng truy cập lớn hơn.

Khối lượng lớn yêu cầu: Khi một tấn công DDoS đang diễn ra, hệ thống sẽ nhận được một lượng lớn yêu cầu truy cập đến cùng một thời điểm, vượt quá khả năng của hệ thống để xử lý.

Khối lượng lớn dữ liệu: Một số loại tấn công DDoS có thể gửi một lượng lớn dữ liệu đến hệ thống mục tiêu, gây áp lực lớn lên băng thông mạng của hệ thống.

Sự giảm đáng kể về hiệu suất hệ thống: Do lượng lưu lượng truy cập lớn, hệ thống có thể trở nên chậm hoặc ngưng hoạt động. Các ứng dụng có thể bị treo hoặc không phản hồi, và hệ thống có thể tạm dừng hoạt động hoặc không thể truy cập được.

**B. Phòng chống DDoS**

Dưới đây là một số giải pháp phòng chống DDoS:

* Sử dụng Firewall: Firewall là một công cụ mạnh mẽ để ngăn chặn các cuộc tấn công DDoS. Nó có thể được cấu hình để chặn các luồng dữ liệu không hợp lệ và các yêu cầu kết nối không hợp lệ đến mục tiêu.
* Sử dụng CDN: CDN (Content Delivery Network) là một giải pháp đám mây để phân phối tài nguyên web trên toàn thế giới. Với CDN, nội dung được lưu trữ trên nhiều máy chủ trên toàn cầu, điều này giúp giảm áp lực lên một máy chủ cụ thể và tăng khả năng chống lại các cuộc tấn công DDoS.
* Sử dụng Load Balancer: Load Balancer là một công cụ giúp phân phối tải đều cho nhiều máy chủ. Nó giúp giảm áp lực lên một máy chủ cụ thể và đảm bảo rằng mục tiêu không bị quá tải.
* Sử dụng Anti-DDoS Services: Các dịch vụ chống DDoS được cung cấp bởi các nhà cung cấp dịch vụ đám mây hoặc các công ty bảo mật. Chúng cung cấp giải pháp chuyên nghiệp để chống lại các cuộc tấn công DDoS.
* Tăng cường bảo mật hệ thống: Tăng cường bảo mật hệ thống bằng cách sử dụng các phần mềm chống virus, bảo vệ tường lửa và các công cụ bảo mật khác giúp giảm khả năng bị tấn công DDoS.

# CHƯƠNG II. TỔNG QUAN VỀ SSH BOTNET

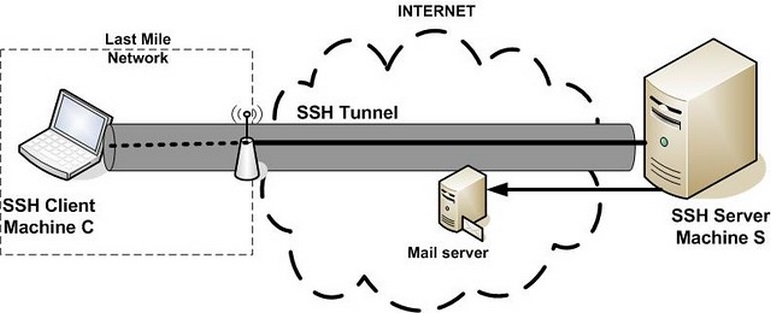
**2.1. Tổng quan SSH**

***2.1.1. Khái niệm***

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.

***2.1.2. Phương thức hoạt động***

SSH gồm cả giao thức mạng và bộ tiện ích cơ bản để triển khai chính giao thức đó. Cụ thể, SSH ứng dụng ảnh mô hình client-server, kết nối với vùng hiển thị Session và vùng Session chạy.

******

*Hình 2: Mô tả cơ chế hoạt động của SH*

Khi triển khai SSH hệ thống sẽ hỗ trợ cả giao thức ứng dụng, sử dụng cho trình giả lập Terminal hoặc truyền file. Trong thực tế, người ta còn sử dụng SSH để phát triển tunnel bảo mật cho các giao thức ứng dụng.

Mục đích SSH được tạo ra là để thay thế cho trình giả lập Terminal, cơ chế đăng nhập không an toàn (Telnet, Rlogin). Giao thức SSH hỗ trợ tính năng đăng nhập, khởi chạy Terminal Session thông qua hệ thống điều khiển từ xa.

Chức năng cơ bản nhất của giao thức SSH là liên kết với một host từ xa, ứng với một phiên Terminal bằng dòng lệnh “ssh server.example.org”. Dòng lệnh này có thể liên kết Client với một máy chủ server.example.com thông qua ID người dùng UserName.

Trường hợp đó là lần kết nối đầu tiên giữa của Server và Host, người dùng phải được thông báo mã khóa của Host. Đồng thời, nhắc nhở qua dòng lệnh dưới đây.

******

*Hình 3: Dòng lệnh nhắc nhở*

Nếu như lựa chọn đồng ý Yes, phiên để tiếp diễn và lưu lại Host Key trong file known\_hosts. Đây là file ẩn, lưu trữ mặc định. Khi Client liên kết trực tiếp với Server trong lần tiếp theo, Host Key lập tức tiến hành xác thực và kết nối.

***2.1.3. Quy trình hoạt động của SSH***

*Bước 1:*

* Định danh host – xác định định danh của hệ thống tham gia phiên làm việc SSH. Đơn giản hơn, đây là bước khởi tạo kết nối SSH. Kết nối này là một kênh giao tiếp bảo mật giữa server và client.
* Xác định host.
* Việc định danh host được thực hiện qua việc trao đổi khoá. Mỗi máy tính điều khiển SSH có một khoá định danh duy nhất. Chúng bao gồm hai thành phần: khoá riêng tư và khoá công cộng. Khoá công cộng được sử dụng khi cần trao đổi giữa các máy chủ. Trong khi khóa riêng tư thì được dùng để giải mã riêng.
* Khi hai hệ thống bắt đầu một phiên làm việc SSH, máy chủ sẽ gửi khoá công cộng của nó cho máy khách. Máy khách sinh ra một khoá phiên ngẫu nhiên. Sau đó nó mã hoá khoá này bằng khoá công cộng của máy chủ. Cuối cùng nó được gửi lại cho máy chủ. Máy chủ sẽ giải mã khoá phiên này bằng khoá riêng của mình và nhận được khoá phiên. Quá trình này được xem như các bước nhận diện máy chủ và máy khách.

*Bước 2:* Mã hoá dữ liệu – thiết lập kênh làm việc mã hoá. Sau khi Client xác định được định danh của Server 1 kết nối bảo mật đối xứng được hình thành giữa 2 bên.

*Bước 3:* Chứng thực và giải mã – xác thực người sử dụng có quyền đăng nhập hệ thống. Kết nối này sẽ được sử dụng để Server xác thực Client.

### ***2.1.4. Chức năng chính***

Giao thức đảm nhiệm khá nhiều chức năng trong hệ thống điều khiển, liên kết máy chủ. Các chức năng cơ bản phải kể đến như:

* Hỗ trợ truy cập từ xa vào những hệ thống, thiết bị ứng dụng giao thức SSH.
* Cho phép dịch chuyển file an toàn.
* Thực thi lệnh bảo mật, an toàn trên hệ thống điều khiển từ xa.
* Quản lý an toàn và hiệu quả thành phần hạ tầng mạng.

SSH có thể kết hợp với Terminal Session thay thế cho những chương trình Telnet có tính bảo mật thấp.

### ***2.1.5. SSH sử dụng khi nào***

Trong mô hình TCP / IP, giao thức SSH sẽ hoạt động tại tầng thứ 4. Đối tượng nhiệm vụ tương tác duy trì kết nối giữa máy chủ và máy khách. Thông qua cơ chế mã hóa chuyên biệt, giao thức này có thể đảm bảo dữ liệu truyền đi an toàn. So với Telnet, Rlogin đã khắc phục yếu điểm tốt về mặt bảo mật.

**- *Sử dụng trong mọi datacenter***

Giao thức SSH có mặt trong hầu hết datacenter và luôn đi kèm phần lớn Server Unix, Mac và Linux. Kết nối SSH hiệu ứng dụng phổ biến để xây dựng giao thức bảo mật cho hệ thống mấy cục bộ, host từ xa.

Chẳng hạn như ứng dụng xây dựng quyền truy cập an toàn từ xa vào hệ thống tài nguyên, cập nhật phần mềm,.. Ngoài chức năng tạo đường dẫn an toàn cho máy cục bộ và Host từ xa, SSH cũng hỗ trợ quản lý Router, phần cứng máy chủ, một số nền tảng ảo hóa.

**- *Kết nối hệ thống server***

SSH có phải là kết nối hệ thống server. Từ đó thực hiện các thay đổi, nâng cấp thông qua công cụ hoặc Terminal. SSH Key thử sử dụng để truy cập tập vào server trong cách tự động, ứng dụng chủ yếu trong Script, backup, công cụ quản lý cấu hình.

**- *Ứng dụng cho hệ thống đăng nhập một lần***

Giao thức SSH hỗ trợ hoàn hảo để ứng dụng vào hệ thống đăng nhập một lần SSO. Theo đó, người dùng sẽ đăng nhập nhanh chóng, di chuyển qua lại giữa tài khoản mà không cần tốn thời gian nhập mật khẩu.

**- *Mã hóa dữ liệu***

Không chỉ hỗ trợ xác thực kết nối mã hóa mà SSH traffic còn tham gia trực tiếp vào quá trình lão hóa. Khi người dùng chuyển file, duyệt web, nhập lệnh hay những tác vụ khác đều diễn ra an toàn.

**- *Xác thực thông tin***

SSH kết hợp với ID người dùng và mật khẩu khi xác thực thông tin. Tuy vậy hiện nay, giao thức này chủ yếu hỗ trợ xác thực giữa hệ thống Host với nhau. Như vậy người dùng cá nhân vẫn phải sử dụng ID user và mật khẩu khi cần kết nối với Host từ xa.

Quá trình kết nối để thực hiện thông qua việc tạo một Public Key Pair ứng với từng Host. Trong đó, mỗi Session lại yêu cầu hai Public Key Pair. Một Key phục vụ xác thực máy cục bộ từ xa, Key còn lại làm nhiệm vụ xác thực máy cục bộ trực tiếp.

**2.2. Giới thiệu Bonet**

***2.2.1. Khái niệm***

Từ **“botnet”** là sự kết hợp của hai từ, **“robot”** và **“network”**.Botnet là một mạng lưới các máy tính được cài phần mềm để làm 1 công việc nào đó. Người ta thường nói về botnet với nghĩa xấu hơn là tốt vì botnet hay được hacker mũ đen sử dụng để tấn công một website hay một dịch vụ online nào đó. Botnet không chỉ được xài để gây hại cho người khác mà còn cho chính máy tính / điện thoại bị lây nhiễm bởi vì nó có thể mang theo những phần mềm mã độc khác, ví dụ như một ransomeware mã hóa dữ liệu của bạn và đòi tiền chuộc để mở khóa chẳng hạn.

→ Chúng có thể không bị phát hiện và bạn thậm chí không biết thiết bị của mình có phải là một phần của botnet hay không.

***2.2.2. Đặc điểm***

Ở đây, một tên tội phạm mạng thực hiện vai trò của một “botmaster” sử dụng virus [Trojan](https://quantrimang.com/phan-biet-virus-trojan-worm-va-rootkit-107894) để xâm phạm bảo mật của một số máy tính và kết nối chúng vào mạng vì mục đích xấu. Mỗi máy tính trên mạng hoạt động như một “bot”, và được kẻ xấu kiểm soát để lây truyền malware, spam hoặc nội dung độc hại nhằm khởi động cuộc tấn công. Botnet còn được gọi là đội quân zombie vì các máy tính liên quan đang được điều khiển bởi một người khác không phải chủ sở hữu của chúng.

Nguồn gốc của botnet chủ yếu là để phục vụ như một công cụ trong các kênh Internet Relay Chat (IRC). Cuối cùng, những kẻ xấu sẽ khai thác các lỗ hổng có trong mạng IRC và các bot được phát triển. Điều này được cố ý tạo ra để thực hiện các hoạt động độc hại như ghi nhật ký về hành động nhấn phím, đánh cắp mật khẩu, v.v...

***2.2.3. Mục đích của Botnet***

Tấn công botnet có thể được dùng cho nhiều mục đích khác nhau. Do đặc thù mạng botnet tập hợp rất nhiều máy tính, cho nên tin tặc có thể sử dụng bonet để thực hiện các cuộc tấn công DDoS vào một máy chủ web nào đó. Theo đó, hàng trăm ngàn máy tính sẽ cùng lúc truy cập vào một website mục tiêu, khiến cho lưu lượng truy cập vào website đó bị quá tải. Kết quả là gây ra tình trạng bị nghẽn mạng, treo máy, không truy cập được.

Ngoài ra, mục đích của tấn công bonet có thể là:

* Gửi mail spam. Đây là cách thức kiếm tiền phổ biến của các Spammer. Bên cạnh đó, botnet cũng được sử dụng để tạo các website gian lận chèn bổ sung quảng cáo chạy trên nền web, khi người dùng click vào link quảng cáo sẽ đem lại lợi nhuận cho hacker.
* Botnet còn được sử dụng để đào Bitcoin nhằm mang lại tiền bán bitcoin cho kẻ tấn công. Botnet cũng tạo và phát tán các loại virut, malware, phần mềm độc hại đến máy tính bạn, sau đó tiếp tục lây lan sang các máy tính khác để tạo một mạng lưới Botnet lớn rộng nhằm thu được nhiều lợi nhuận hơn.

***2.2.4. Phân loại Botnet***

Botnet được phân loại theo cách thức liên hệ với C&C Server. Do đó, có 5 loại botnet chính sau:

* **IRC Botnet**

IRC là một giao thức thường được dùng trong việc chat. Như chúng ta hay chat trên facebook thì IRC Botnet cũng vậy. Bot master ra lệnh cho các bot của mình thông qua 1 channel chat (giống như việc chat nhóm vậy).

* **P2P Botnet**

Khác với IRC Botnet, mô hình của P2P sẽ là mô hình phân tán vì các kết nối của các bot với C&C là các kết nối P2P - kết nối ngang hàng. Với mô hình này, rất khó để shut-down vì ở mô hình phân tán sẽ không còn client-server mà mỗi bot để là một client và cũng đều là một server.

Bot master share command fiel với search key đến mỗi bot trong mạng P2P botnet. Các bot sẽ nhận lệnh bằng việc liên lạc với nhau và nhận lệnh với các search key tương ứng.

* **HTTP Botnet**

Mô hình của HTTP Botnet cũng là mô hình tập trung client-server như IRC Botnet nhưng thay vì nhận lệnh thông qua kênh chat thì HTTP Botnet sẽ sử dụng giao thức HTTP để gửi request và nhận lệnh từ các bot đến Bot master. Như vậy, HTTP Botnet không thể nhận lệnh realtime như P2P Botnet được mà sẽ gửi request liên tục hoặc qua x thời gian nào đó để cập nhật tình hình.

* **Botnet Cloud**

Botmaster sử dụng cách dịch vụ cloud. Với Botnet loại này rất khó để nhận diện vì nó sử dụng các dịch vụ cloud với tính năng bảo mật cao và có sẵn. Ví dụ như sử dụng dịch vụ của Google chẳng hạn.

Cùng với sự phát triển của các dịch vụ cloud thì việc deploy một kiến trúc để thực thi botnet nhanh hơn rất nhiều so với botnet truyền thống chưa kể việc nó luôn luôn online, sẵn sàng thực thi các hành động.

* **Mobile Botnet**
* **SMS Based**

Cùng với sự phát triển nhanh chóng của công nghệ và sự bùng nổ các thiết bị smart (có gắn CPU) nhất là smart phone. Botnet đã có một biến thể mới, lợi dụng dịch vụ SMS trên các smart phone và các thiết bị di động khác để gửi, nhận lệnh.

* **Bluetooth based**

Là loại botnet sử dụng giao thức Bluetooth để thực hiện các hành vi độc hại. Nhưng loại botnet này có điểm yếu là khi đi ra ngoài vùng phủ sóng của Bluetooth thì không thể hoạt động được.

* **Hybrid Botnet**

Hybrid Botnet là sự kết hợp của IRC Botnet, P2P Botnet và HTTP Botnet.

***2.2.5. Cấu trúc của botnet***

Diagram

Description automatically generated

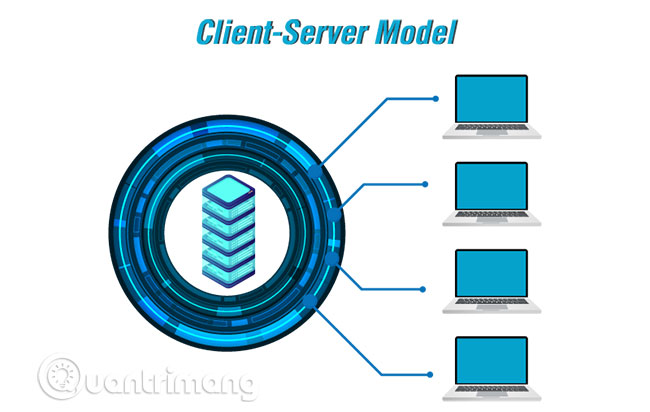
*Hình 4: Cấu trúc botnet*

Cấu trúc của botnet thường có một trong hai dạng:

1. Mô hình client - server (máy khách - máy chủ)
2. Mô hình peer-to-peer (ngang hàng).

**Mô hình client - server (máy khách - máy chủ)**

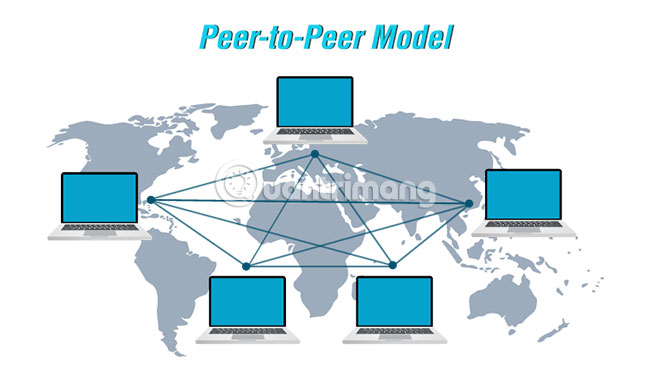
Trong cấu trúc client - server, một mạng cơ bản được thiết lập với một máy chủ, hoạt động như botmaster. Botmaster kiểm soát việc truyền thông tin từ mỗi client để thiết lập lệnh và điều khiển các thiết bị khách. Mô hình client - server hoạt động với sự trợ giúp của phần mềm đặc biệt và cho phép botmaster duy trì quyền kiểm soát. Mô hình này có một vài nhược điểm như nó có thể được định vị dễ dàng và chỉ có một điểm kiểm soát. Trong mô hình này, nếu máy chủ bị phá hủy, botnet sẽ "chết".



*Hình 5: Mô hình mạng client – server*

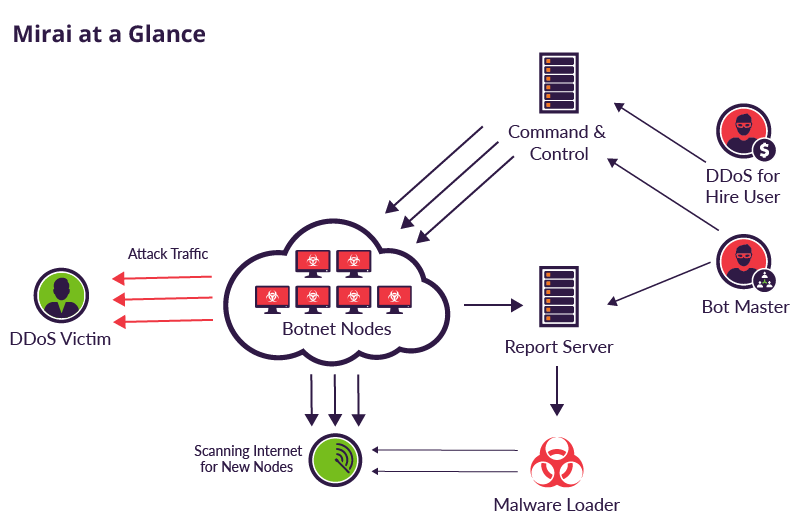
**Mô hình peer-to-peer (ngang hàng)**

Để khắc phục nhược điểm của việc dựa vào một máy chủ tập trung, các botnet đã phát triển. Các botnet mới được kết nối với nhau dưới dạng cấu trúc ngang hàng. Trong mô hình botnet P2P, mỗi thiết bị được kết nối hoạt động độc lập như một client và server, phối hợp với nhau để cập nhật và truyền thông tin qua lại. Cấu trúc botnet P2P mạnh hơn do không có một nơi điều khiển tập trung duy nhất.



*Hình 6: Mô hình mạng peer-to-peer (ngang hàng)*

***2.2.6. Các loại tấn công Botnet***



*Hình 7: Các loại tấn công Botnet*

**Distributed Denial of Operations Service (DDoS)**

Một botnet có thể được sử dụng cho một [cuộc tấn công Distributed Denial of Operations Service (DDoS - từ chối dịch vụ phân tán)](https://quantrimang.com/tim-hieu-ve-tan-cong-tu-choi-dich-vu-dos-34926) để phá hủy kết nối và dịch vụ mạng. Điều này được thực hiện bằng cách làm quá tải tài nguyên tính toán (computational resource) hoặc tiêu tốn băng thông của nạn nhân.

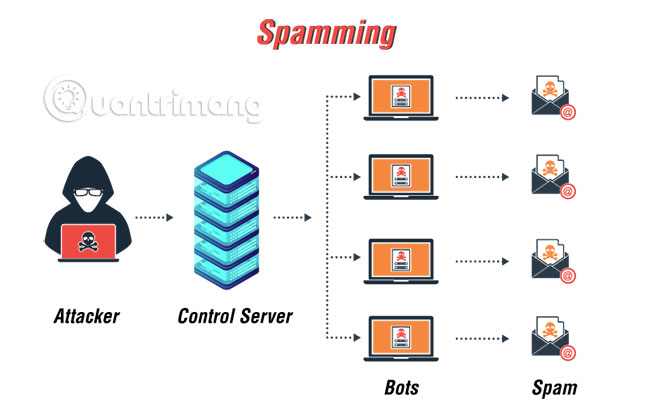
Các cuộc tấn công được thực hiện phổ biến nhất là những cuộc tấn TCP SYN và UDP flood. Các cuộc tấn công DDoS không chỉ giới hạn ở những máy chủ web, mà có thể nhắm mục tiêu đến bất kỳ dịch vụ nào được kết nối với Internet. Mức độ nghiêm trọng của cuộc tấn công có thể được tăng lên bằng cách sử dụng HTTP flood trên trang web của nạn nhân. Hình thức này gọi là spidering được tiến hành để tăng tải hiệu quả.

Một trong những cuộc tấn công botnet DDoS lớn nhất liên quan đến [IoT](https://quantrimang.com/internet-of-things-iot-hay-mang-luoi-van-vat-ket-noi-la-gi-131856) và sử dụng virus botnet Mirai. Virus đã nhắm mục tiêu và kiểm soát hàng chục ngàn thiết bị Internet được bảo vệ lỏng lẻo, sau đó biến chúng thành bot để khởi động một cuộc tấn công DDoS. Mirai tiếp tục mở rộng, khiến cuộc tấn công trở nên phức tạp hơn.

**Spamming (phát tán thư rác) và giám sát lưu lượng**

Một bot có thể được sử dụng để xác định sự hiện diện của dữ liệu nhạy cảm trong các zombie hoặc máy tính bị nhiễm. Nó cũng có thể xác định vị trí các botnet đối thủ, nếu được cài đặt trong cùng một máy. Một số bot có thể mở proxy SOCKS v4/v5 (giao thức proxy chung cho mạng dựa trên TCP/IP). Khi proxy SOCKS được kích hoạt trên máy bị xâm nhập, nó có thể được sử dụng cho nhiều mục đích khác nhau như spamming (phát tán thư rác). Bot sử dụng [packet sniffer](https://quantrimang.com/packet-sniffer-la-gi-118706) để theo dõi thông tin hoặc dữ liệu được truyền bởi máy bị xâm nhập. Sniffer có thể truy xuất thông tin nhạy cảm như tên người dùng và mật khẩu.

Grum là loại thư rác khó phát hiện, vì nó lây nhiễm những file được sử dụng bởi các registry Autorun. Mạng botnet này đã thu hút các nghiên cứu vì nó tương đối nhỏ, chỉ có 600.000 thành viên nhưng chiếm tới 40 tỷ email spam mỗi ngày, tương đương khoảng 25% tổng số email spam.



*Hình 8: Mô tả Spamming*

**Keylogging**

Với sự trợ giúp của [keylogger](https://quantrimang.com/tim-hieu-ve-keylogger-va-cach-phong-tranh-108996), việc botmaster lấy thông tin nhạy cảm và đánh cắp dữ liệu trở nên dễ dàng. Sử dụng chương trình keylogger, kẻ tấn công có thể thu thập các phím được nhập trong PayPal, Yahoo, v.v...

Một loại phần mềm gián điệp được xác định là OSX/XSLCmd, có thể chuyển từ Windows sang OS X, bao gồm khả năng keylogging và chụp màn hình.

**Đánh cắp danh tính hàng loạt**

Các loại bot khác nhau có thể được kết hợp để thực hiện hành vi trộm cắp danh tính quy mô lớn, đây là một trong những hành vi phạm tội phát triển nhanh nhất. Email spam được gửi bởi bot để hướng lưu lượng truy cập đến các trang web giả mạo, đại diện cho bot để thu thập dữ liệu cá nhân. Bot có thể được sử dụng để xuất hiện như một công ty hợp pháp và yêu cầu người dùng gửi thông tin cá nhân, như mật khẩu tài khoản ngân hàng, chi tiết thẻ tín dụng, chi tiết thuế, v.v... Hành vi trộm cắp danh tính hàng loạt diễn ra bằng cách sử dụng email [phishing](https://quantrimang.com/cach-thuc-hoat-dong-cua-phishing-69841) để lừa nạn nhân nhập thông tin đăng nhập trên trang web như eBay, Amazon, hoặc thậm chí là ngân hàng.

**Lạm dụng việc trả tiền cho mỗi lần nhấp**

Chương trình AdSense của Google cho phép các trang web hiển thị quảng cáo Google và từ đó kiếm tiền từ chúng. Google trả tiền cho chủ sở hữu trang web trên cơ sở số lần nhấp mà quảng cáo của họ thu thập được. Máy bị nhiễm được sử dụng để tự động nhấp vào một trang web, làm tăng số lần nhấp được gửi đến công ty bằng quảng cáo.

**Lây lan botnet**

Botnet cũng được sử dụng để lan truyền các botnet khác bằng cách thuyết phục người dùng tải xuống chương trình cụ thể và chương trình được thực thi thông qua email, HTTP hoặc FTP. Thường thì virus sử dụng botnet này sẽ được phát tán qua email. Hai nhà nghiên cứu bảo mật trong tháng 1 năm 2017, đã phát hiện ra botnet Twitter “Star Wars”, bao gồm 350.000 tài khoản bot đã tweet các trích dẫn ngẫu nhiên từ một sê-ri phim. Các bot như vậy nếu tiếp tục tồn tại có thể tạo ra những chủ đề thịnh hành giả mạo gây xôn xao dư luận, gửi thư rác không mong muốn, phát động các cuộc tấn công mạng và hơn thế nữa.

**Phần mềm quảng cáo**

[Phần mềm quảng cáo](https://quantrimang.com/lich-su-spyware-adware-va-co-che-phat-tan-17124) được sử dụng để thu hút người dùng bằng cách quảng cáo trên các trang web hoặc ứng dụng. Chúng xuất hiện mà không có sự cho phép của người dùng, với quảng cáo gốc bị thay thế bởi phần mềm quảng cáo lừa đảo, lây nhiễm vào hệ thống của bất kỳ người dùng nào nhấp vào nó.

Phần mềm quảng cáo trông giống như quảng cáo vô hại nhưng sử dụng phần mềm gián điệp để thu thập dữ liệu trình duyệt. Để thoát khỏi phần mềm quảng cáo, cần phải có phần mềm [chặn quảng cáo](https://quantrimang.com/cach-chan-quang-cao-khi-luot-web-116984). Mặc dù có sẵn nhiều phiên bản phần mềm chặn quảng cáo miễn phí và trả phí, nhưng tốt nhất bạn nên sử dụng một tùy chọn có giấy phép. Nhiều gói quét virus cũng đi kèm với chương trình chống phần mềm độc hại.

Botnet có thể bị trục xuất hoặc ngừng xâm nhập vào máy tính bằng cách sử dụng chương trình chống phần mềm độc hại, có thể phát hiện việc lây nhiễm trên ổ cứng hoặc lưu lượng mạng và xử lý chúng ngay lập tức. Mặt khác, cách tiếp cận hiệu quả nhất sẽ là tìm hiểu để nhận thức được toàn diện về cách chống lại botnet.

***2.2.7. Phòng chống Botnet***

**Cấp độ cá nhân**

1. Cài đặt [phần mềm diệt virus](https://quantrimang.com/4-phan-mem-diet-virus-mien-phi-tot-nhat-cho-windows-80614) hoặc chống thư rác và giữ cho chúng được cập nhật thường xuyên.
2. Bật cài đặt [tường lửa](https://quantrimang.com/cach-tat-bat-windows-firewall-trong-windows-7-68908) và hạn chế truy cập không mong muốn.
3. Hãy chắc chắn rằng hệ điều hành được cập nhật theo thời gian.
4. Không tải xuống nội dung bất hợp pháp như nhạc, game, file lậu, v.v... từ Internet.
5. Không nhấp vào file đính kèm hoặc liên kết từ các email không xác định.

**Cấp độ mạng**

1. Có tường lửa, hệ thống IDS/IPS và tính năng lọc nội dung.
2. Giám sát lưu lượng truy cập tăng bất thường.
3. Có tính năng bảo vệ DDoS tại chỗ.
4. Nếu hệ thống người dùng cá nhân nghi ngờ bị nhiễm botnet, hãy thử xóa phần mềm độc hại ngay lập tức. Nếu điều này không được thực hiện kịp thời, các hệ thống khác trong mạng cũng có thể bị lây nhiễm.
5. Đảm bảo rằng tất cả các cá nhân trong công ty đã có phần mềm được cập nhật trên hệ thống của họ.
6. Giám sát nhật ký tường lửa để xác định các trung tâm điều khiển và lệnh botnet.
7. Nếu bất kỳ việc lây nhiễm nào được xác định, hãy thông báo cho các nhà cung cấp phần mềm diệt virus ngay lập tức.

**2.3. SSH Bonet**

***2.3.1. Tìm hiểu về SSH Botnet***

SSHBotnet là một thuật ngữ được sử dụng để chỉ một tập hợp các máy tính đã bị xâm nhập và được kiểm soát từ xa thông qua giao thức SSH. Các máy tính này có thể bao gồm các thiết bị IoT, máy tính cá nhân và máy chủ có cài đặt các dịch vụ SSH.

Các máy tính trong mạng botnet này được kiểm soát bởi một máy tính trung tâm được gọi là Command and Control Server (C&C Server). Kẻ tấn công có thể sử dụng botnet này để thực hiện các cuộc tấn công mạng, bao gồm DDoS (Distributed Denial of Service), spam email, phát tán phần mềm độc hại, đánh cắp thông tin và các hành động khác.

Các kẻ tấn công có thể sử dụng các phương pháp khác nhau để xâm nhập vào các thiết bị có kết nối internet và cài đặt mã độc để kiểm soát từ xa. Một trong những phương pháp phổ biến nhất là tấn công brute-force vào giao thức SSH để đoán mật khẩu đăng nhập. Khi tấn công thành công, kẻ tấn công có thể sử dụng các lỗ hổng bảo mật trên thiết bị để cài đặt mã độc hoặc phần mềm độc hại khác.

***2.3.2. Cấu trúc SSH Botnet***

Cấu trúc của một SSH Botnet thường được thiết kế với mục đích tấn công một mục tiêu cụ thể. Nó thường bao gồm các thành phần sau:

1. Command and Control (C&C) server: Đây là trung tâm điều khiển của botnet, nơi tất cả các bot được kết nối để nhận lệnh tấn công.
2. Botnet controller: Là phần mềm độc hại được cài đặt trên máy tính của các nạn nhân và có thể được điều khiển từ xa bởi C&C server.
3. Bot agent: Là phần mềm độc hại được cài đặt trên các máy chủ khác nhau và được điều khiển bởi botnet controller. Bot agent có thể là các tệp script, malware hoặc phần mềm độc hại khác.
4. Proxy server: Các bot có thể sử dụng các proxy server để giấu danh tính của chúng khi tấn công. Proxy server cũng có thể được sử dụng để giảm thiểu mối đe dọa về bảo mật cho C&C server.
5. Payload: Là một phần mềm độc hại được sử dụng để tấn công mục tiêu. Payload có thể là bất kỳ phần mềm độc hại nào, bao gồm keyloggers, ransomware, backdoor, v.v.

***2.3.3. Cách phòng chống SSH Botnet***

Để phòng chống SSH botnet, có thể thực hiện các biện pháp sau:

1. Sử dụng các phần mềm bảo mật tường lửa và phần mềm chống virus để chặn các truy cập trái phép vào hệ thống.
2. Điều chỉnh cấu hình hệ thống SSH, chẳng hạn như tắt chức năng đăng nhập bằng mật khẩu và sử dụng khóa SSH thay thế.
3. Sử dụng các phần mềm giám sát hệ thống để theo dõi lưu lượng mạng và hoạt động của các tiến trình hệ thống. Nếu có hoạt động bất thường, bạn cần thực hiện các biện pháp bảo vệ ngay lập tức.
4. Cập nhật hệ thống và các ứng dụng thường xuyên để đảm bảo các lỗ hổng bảo mật được vá kịp thời.
5. Sử dụng các công cụ quản lý đăng nhập như Fail2ban để chặn các địa chỉ IP có nhiều lần đăng nhập không thành công.
6. Giáo dục người dùng về cách bảo mật mật khẩu và tránh sử dụng các mật khẩu yếu hoặc dễ đoán.
7. Sử dụng các giải pháp giám sát và quản lý mạng để phát hiện và ngăn chặn các cuộc tấn công mạng trên hệ thống.
8. Thực hiện các biện pháp kiểm tra an ninh mạng thường xuyên, bao gồm kiểm tra định kỳ hệ thống và quản lý các chính sách bảo mật mạng.
9. Tạo và quản lý các lớp mạng khác nhau trong hệ thống để giảm thiểu rủi ro an ninh mạng và ngăn chặn các cuộc tấn công tràn lan trên toàn bộ hệ thống.
10. Thực hiện các bản sao lưu định kỳ và lưu trữ chúng ngoài hệ thống để đảm bảo tính toàn vẹn của dữ liệu khi xảy ra cuộc tấn công.

***2.3.4. Tìm hiểu module Paramiko trong Python***

Paramiko là một thư viện Python được sử dụng để kết nối, quản lý và tương tác với các máy chủ thông qua giao thức SSH (Secure Shell). Thư viện này cung cấp các lớp và phương thức cho phép xác thực, kết nối, truyền dữ liệu, quản lý phiên, và thực thi các lệnh trên máy chủ từ xa.

Paramiko cung cấp một cách tiện lợi và bảo mật cho việc tự động hóa các tác vụ trên các máy chủ từ xa, như kiểm tra tình trạng máy chủ, cài đặt phần mềm, sao lưu dữ liệu, v.v. Với paramiko, các lập trình viên có thể tạo ra các ứng dụng Python mạnh mẽ và linh hoạt cho việc quản lý hệ thống từ xa.

***2.3.5. Dictionary mật khẩu SSH dùng Paramiko***

Tấn công từ điển mật khẩu SSH là một phương pháp thường được sử dụng để đánh cắp thông tin đăng nhập vào các hệ thống. Đối với SSH Botnet, phương pháp này thường được sử dụng để tìm ra các mật khẩu đăng nhập vào các máy chủ mục tiêu.

Trong Python, thư viện Paramiko cũng có thể được sử dụng để thực hiện các tấn công từ điển mật khẩu SSH. Để thực hiện tấn công này, ta cần tạo một kết nối SSH đến máy chủ mục tiêu bằng thư viện Paramiko. Sau đó, ta sẽ thực hiện lần lượt các yêu cầu đăng nhập với các mật khẩu khác nhau cho đến khi đăng nhập thành công hoặc đã thử hết tất cả các mật khẩu trong từ điển.

Dưới đây là một ví dụ đơn giản về cách sử dụng thư viện Paramiko để thực hiện tấn công từ điển mật khẩu SSH:

Graphical user interface, text, application

Description automatically generated

*Hình 9: Tấn công đơn giản Dictionary sử dụng Paramiko*

Trong đoạn mã trên, ta sử dụng phương thức paramiko.SSHClient() để thiết lập kết nối SSH đến máy chủ mục tiêu. Sau đó, ta sử dụng một danh sách các mật khẩu để thực hiện tấn công từ điển mật khẩu.

Với mỗi mật khẩu trong danh sách, ta thực hiện yêu cầu đăng nhập bằng cách gọi phương thức ssh.connect() với tham số là mật khẩu đăng nhập. Nếu đăng nhập thành công, ta in ra thông báo và thoát khỏi vòng lặp. Nếu đăng nhập thất bại, ta tiếp tục thử với mật khẩu tiếp theo trong danh sách.

***2.3.6. Khai thác SSH thông qua khóa bí mật yếu***

Mật khẩu cung cấp một phương pháp xác thực tới máy chủ SSH nhưng đây không phải là phương pháp duy nhất. Ngoài ra, SSH cung cấp các phương tiện để xác thực bằng cách sử dụng mật mã khóa công khai. Trong trường hợp này, máy chủ biết khóa công khai và người dùng biết khóa riêng. Sử dụng thuật toán RSA hoặc DSA, máy chủ tạo ra các khóa này để đăng nhập vào SSH. Thông thường, điều này cung cấp một phương pháp tuyệt vời để xác thực. Với khả năng tạo các khóa 1024 bit, 2048 bit hoặc 4096 bit, quá trình xác thực này gây khó khăn cho việc sử dụng brute force như chúng tôi đã làm với mật khẩu yếu.

Tuy nhiên, vào năm 2006, một điều thú vị đã xảy ra với bản phân phối Debian Linux. Một nhà phát triển đã nhận xét về một dòng mã được tìm thấy bởi một bộ công cụ phân tích phần mềm tự động. Dòng mã cụ thể đảm bảo entropy trong ý tưởng sáng tạo của khóa SSH. Bằng cách nhận xét về dòng mã cụ thể, kích thước của không gian khóa có thể tìm kiếm giảm xuống còn 15 bit entropy. Không chỉ có 15 bit entropy, điều này có nghĩa là chỉ có 32.767 khóa tồn tại cho mỗi thuật toán và kích thước.

***2.3.7. Dựng SSH Botnet với Python***

Bây giờ đã chứng minh rằng có thể kiểm soát một máy chủ thông qua SSH, hãy mở rộng nó để kiểm soát nhiều máy chủ đồng thời. Những kẻ tấn công thường sử dụng các bộ sưu tập máy tính bị xâm nhập cho các mục đích xấu. Chúng tôi gọi đây là một mạng botnet vì các máy tính bị xâm nhập hoạt động giống như các bot để thực hiện các hướng dẫn. Để xây dựng mạng botnet sẽ phải đưa ra một khái niệm mới - một lớp (a class). Khái niệm về một lớp đóng vai trò là cơ sở cho một mô hình lập trình có tên là lập trình hướng đối tượng. Trong hệ thống này, khởi tạo các đối tượng riêng lẻ bằng các phương thức liên kết. Đối với mạng botnet, mỗi bot hoặc ứng dụng client riêng lẻ sẽ yêu cầu khả năng kết nối và ra lệnh.

# CHƯƠNG III. TRIỂN KHAI THỰC NGHIỆM

**3.1. Xây dựng class SSHBotnet**

Graphical user interface, text, application

Description automatically generated

Định nghĩa một class có tên là SSHBotnet, được sử dụng để thiết lập kết nối SSH với một máy chủ từ xa và thực hiện các hoạt động trên đó bằng cách sử dụng thư viện Paramiko.

Hàm \_\_init\_\_ được gọi khi khởi tạo một đối tượng SSHBotnet, với các tham số là hostname, username, và password để thiết lập kết nối SSH. Hàm này thực hiện các công việc sau:

* Lưu trữ hostname, username, và password trong các biến riêng tư (self.\_\_hostname, self.\_\_username, self.\_\_password).
* Tạo một đối tượng SSHClient từ thư viện Paramiko (self.\_\_ssh).
* Đặt chính sách thêm khóa host tự động (paramiko.AutoAddPolicy()).
* Kết nối đến hostname bằng username và password sử dụng self.\_\_ssh.connect().
* Mở một kết nối SFTP tới đường dẫn hiện tại trên máy chủ từ xa (self.\_\_sfpt).
* Lưu trữ đường dẫn hiện tại (self.\_\_current\_dir).

Hàm \_\_exit\_\_ được sử dụng để đóng kết nối SSH và SFTP khi đối tượng SSHBotnet bị thu hồi. Các kết nối được đóng bằng cách gọi phương thức close() của đối tượng self.\_\_sfpt và self.\_\_ssh.

Text

Description automatically generated

Phương thức format\_permissions nhận một tham số là giá trị quyền truy cập và chuyển đổi nó sang chuỗi biểu diễn của các quyền truy cập. Phương thức này trả về một chuỗi ba ký tự cho mỗi quyền truy cập:

* 'r' cho quyền đọc
* 'w' cho quyền ghi
* 'x' cho quyền thực thi.

Ví dụ, quyền truy cập 755 sẽ được chuyển đổi thành chuỗi "rwxr-xr-x".

Phương thức format\_date nhận một tham số là timestamp (thời điểm dựa trên giây kể từ thời điểm bắt đầu của Epoch) và trả về một chuỗi đại diện cho thời điểm đó dưới định dạng "mm/dd/yyyy hh:mm AM/PM".

Để làm điều này, phương thức này sử dụng thư viện datetime trong Python để chuyển đổi timestamp thành đối tượng datetime, sau đó sử dụng phương thức strftime() để định dạng chuỗi theo định dạng đã cho.

Graphical user interface, text, application

Description automatically generated

Đoạn code này định nghĩa phương thức list\_dir() để liệt kê các tập tin và thư mục trong thư mục hiện tại trên máy chủ SSH. Đầu tiên, phương thức này sử dụng phương thức listdir\_attr() của đối tượng SFTPClient để lấy danh sách các tập tin và thư mục trong thư mục hiện tại và trả về một danh sách các đối tượng SFTPAttributes.

Sau đó, vòng lặp for được sử dụng để lặp qua danh sách các đối tượng SFTPAttributes này và hiển thị thông tin về các tập tin và thư mục đó bao gồm: tên, loại, kích thước, quyền truy cập và thời gian sửa đổi gần nhất.

Phương thức format\_permissions() được sử dụng để định dạng quyền truy cập của tập tin hoặc thư mục từ số octal sang chuỗi "rwxrwxrwx" tương ứng với quyền đọc, ghi và thực thi của chủ sở hữu, nhóm và người dùng khác.

Phương thức format\_date() được sử dụng để định dạng thời gian sửa đổi của tập tin hoặc thư mục từ dạng timestamp sang chuỗi ngày/tháng/năm giờ:phút giây AM/PM.

Graphical user interface, text, application

Description automatically generated

Phương thức cd\_new\_dir được sử dụng để thay đổi thư mục hiện tại trên máy chủ SSH

Graphical user interface, text, application

Description automatically generated

Phương thức execute\_command trong class SSHBotnet được sử dụng để thực thi các lệnh trên remote server thông qua giao thức SSH.

Nếu lệnh bắt đầu bằng “more”, phương thức sẽ mở tệp tin có tên được chỉ định và in nội dung của tệp tin đó ra màn hình.

Nếu không phải là lệnh “more”, phương thức sử dụng exec\_command của thư viện paramiko để thực thi lệnh trên remote server và lấy ra kết quả đầu ra và lỗi (nếu có) từ đối tượng stdout và stderr của kết nối SSH. Kết quả đầu ra và lỗi được in ra màn hình để người dùng xem. Nếu lệnh không hợp lệ, thông báo lỗi sẽ được in ra màn hình.

Text

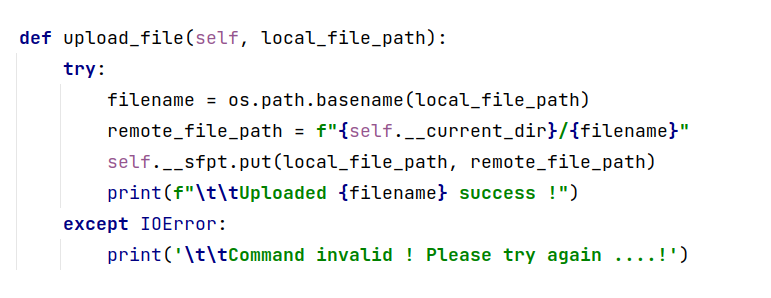
Description automatically generated

Phương thức download\_file được sử dụng để tải về một tệp từ máy chủ từ xa và lưu trữ nó trên máy tính cục bộ của người dùng.

Để thực hiện việc này, hàm sẽ xác định tên tệp cần tải về từ command, sau đó sử dụng phương thức get của đối tượng SFTPClient để tải tệp từ đường dẫn từ xa và lưu trữ nó trên máy tính cục bộ.

Khi tải xong, hàm sẽ in ra thông báo "Downloaded {filename} success !" để thông báo cho người dùng biết tệp đã được tải về thành công.

Nếu có lỗi xảy ra trong quá trình thực hiện, hàm sẽ in ra thông báo lỗi "Command invalid ! Please try again ....!".



Phương thức upload\_file trong lớp SSHBotnet, nó cho phép tải lên một tệp từ máy tính cục bộ của người dùng lên máy chủ từ xa thông qua SFTP.

Cụ thể, phương thức này nhận đầu vào là đường dẫn của tệp cục bộ mà người dùng muốn tải lên. Sau đó, nó lấy tên tệp từ đường dẫn và xác định đường dẫn tới tệp từ xa trên máy chủ từ xa.

Sau đó, nó sử dụng phương thức put của thư viện paramiko để tải tệp lên máy chủ từ xa.Nếu tải lên thành công, nó sẽ in ra thông báo "Uploaded {filename} success !". Nếu có lỗi xảy ra trong quá trình tải lên, phương thức sẽ in ra thông báo "Command invalid ! Please try again ....

Graphical user interface, text, application

Description automatically generated

Chức năng tấn công mạng DDOS vào một địa chỉ website đã được chỉ định. Khi hàm này được gọi, địa chỉ website được trích xuất từ câu lệnh command, sau đó thêm tiền tố "https://" để tạo thành URL của trang web đó.

Một lệnh curl sẽ được tạo ra để kiểm tra trạng thái của trang web, và sau đó một vòng lặp sẽ chạy 10 lần với mỗi lần chạy gửi yêu cầu curl đến địa chỉ URL.

Kết quả sau mỗi lần yêu cầu curl được in ra màn hình bao gồm tên máy chủ, tên đăng nhập và mật khẩu của tài khoản đang chạy, và các thông tin về yêu cầu được ghi nhận là tấn công DDOS và trạng thái trả về từ máy chủ web. Nếu có lỗi xảy ra trong quá trình tấn công, nó sẽ được bỏ qua.

**3.2. Xây dựng class BotMaster**

Class BotMaster sẽ thực thi lệnh với nhiều list botnet đã chiếm quyền .

**A picture containing text

Description automatically generated**

Phương thức khởi tạo init() được gọi khi một đối tượng BotMaster được tạo ra. Nó khởi tạo danh sách botnets và gọi phương thức scan\_botnets() để quét các botnet.

Phương thức start() chứa một vòng lặp vô hạn và hiển thị menu lựa chọn cho người dùng. Người dùng được yêu cầu nhập một số nguyên tương ứng với các tùy chọn menu. Các tùy chọn này bao gồm liệt kê danh sách botnets còn sống, thực thi lệnh và tấn công DDOS.

Nếu người dùng chọn 0, chương trình thoát khỏi vòng lặp và kết thúc.

Nếu người dùng nhập số không hợp lệ, chương trình tiếp tục chạy vòng lặp và yêu cầu người dùng chọn lại.

Graphical user interface, text, application

Description automatically generated

Phương thức \_\_scan\_botnets trong class BotMaster được sử dụng để quét các botnet trong tệp văn bản dictionary.txt.

Đầu tiên, phương thức đọc dữ liệu từ tệp và lưu trữ vào biến datas. Sau đó, phương thức sử dụng module threading để tạo một luồng cho mỗi bot trong datas, chạy phương thức connect\_botnet với các tham số tương ứng là data[0] (hostname), data[1] (username) và data[2] (password).

Cuối cùng, phương thức sử dụng thread.join() để đợi cho tất cả các luồng đã tạo kết thúc. Nếu có lỗi xảy ra trong quá trình thực thi, phương thức sẽ bỏ qua nó.

Graphical user interface, text, application, email

Description automatically generated

Phương thức attack\_ddos là một phương thức tĩnh (staticmethod), nhận vào một tham số botnet và cmd. Phương thức này sẽ gọi phương thức execute của đối tượng botnet để thực hiện lệnh cmd trên botnet đó.

Phương thức ddos sẽ yêu cầu người dùng nhập vào một lệnh, sau đó tạo ra một danh sách các luồng (threads) và cho mỗi botnet trong danh sách botnet, tạo ra một luồng mới để gọi phương thức attack\_ddos với đối số là botnet và cmd. Sau đó chạy tất cả các luồng và đợi cho tất cả các luồng kết thúc bằng cách gọi phương thức join() cho mỗi luồng. Kết quả là tất cả các botnet trong danh sách botnet sẽ thực hiện lệnh được cung cấp bởi người dùng.

**3.3. Tấn công Dictionary tìm mật khẩu SSH**

**Chuẩn bị :**

- 1 máy Kali ,2 máy Windowns 10

- 1 file ‘passwords.txt’, nội dung chứa mật khẩu yếu ,người dùng hay đặt

Text

Description automatically generated

* IP máy và tên người dùng máy Windowns (máy mục tiêu)

**Code triển khai :**

Graphical user interface, text, application

Description automatically generated

Phương thức read\_file(filename) dùng để đọc tất cả các dòng trong tệp có tên được chỉ định trong tham số filename và trả về một danh sách các chuỗi các dòng đã được bỏ đi ký tự xuống dòng (\n).

Phương thức get\_argument() được sử dụng để lấy đối số khi chạy chương trình từ dòng lệnh. Nó sử dụng thư viện optparse để định nghĩa các tùy chọn (options) và đối số (arguments) được truyền vào khi chạy chương trình. Các tùy chọn được định nghĩa bao gồm -i hoặc --ip, -u hoặc --user, và -c hoặc --command.

Nếu không truyền vào tùy chọn -i hoặc không cung cấp một địa chỉ IP hợp lệ, chương trình sẽ báo lỗi. Hàm get\_argument() trả về tất cả các tùy chọn được cung cấp khi chạy chương trình.

Graphical user interface, text, application

Description automatically generated

Đoạn code này định nghĩa hàm connect\_botnet() để kết nối đến botnet thông qua SSH.

Hàm này có bốn đối số: hostname, username, key, và command. hostname là địa chỉ IP của botnet, username là tên người dùng để đăng nhập vào botnet, key là đường dẫn đến tệp khóa để đăng nhập và command là lệnh cần thực thi trên botnet.

Trong hàm, một kết nối SSH được tạo bằng cách sử dụng các thông tin hostname, username và key. Sau đó, nếu kết nối được thiết lập thành công, hàm sẽ in ra thông tin của botnet và thực thi lệnh được cung cấp trong command. Nếu kết nối không thành công, hàm sẽ trả về False.

Hàm trả về True nếu kết nối được thiết lập thành công và lệnh được thực thi thành công.

Graphical user interface, text, application

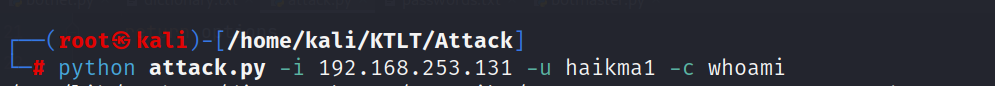
Description automatically generated

Đoạn code này định nghĩa hàm attack(file\_name), hàm này sử dụng một danh sách các mật khẩu trong tệp tin file\_name để kết nối đến các botnet và thực thi một lệnh trên đó. Nếu kết nối được thiết lập thành công với bất kỳ mật khẩu nào trong danh sách, quá trình kết nối và thực thi lệnh sẽ dừng lại.

Trong phương thức attack(file\_name), đầu tiên nó đọc danh sách các mật khẩu từ tệp tin file\_name bằng cách sử dụng hàm read\_file(). Sau đó, nó lấy các thông số địa chỉ IP, tên đăng nhập và lệnh cần thực thi từ dòng lệnh sử dụng hàm get\_argument(). Cuối cùng, vòng lặp duyệt qua danh sách các mật khẩu, mỗi lần kết nối đến một botnet với một mật khẩu. Nếu kết nối được thiết lập thành công, quá trình kết nối và thực thi lệnh sẽ dừng lại bằng cách break. Nếu không có mật khẩu nào hoạt động, hàm sẽ không có hành động gì.

Cuối cùng, nếu đoạn code được chạy như một chương trình chính (bằng cách sử dụng if \_\_name\_\_ == '\_\_main\_\_':), nó sẽ gọi hàm attack('passwords.txt') để thực thi tấn công với danh sách mật khẩu được cung cấp trong tệp tin 'passwords.txt'.

**Thực thi triển khai code :**

****

**Chạy terminal, chỉ định :**

* IP mục tiêu là 192.168.253.131
* Người dùng là haikma1
* Câu lệnh thực khi sau khi tìm thấy là ‘whoami’

Kết quả:

Text

Description automatically generated

Tìm thấy password có trong từ điển : ‘passwords.txt’ là 12345678

Thực thi câu lệnh ‘whoami’ trả về kết quả : desktop-bnpe2tr\haikma1

**Bổ sung:**

SSH còn kết nối thông qua khóa bí mật ,nhưng khóa bí mật nếu yếu ,thì có thể bị tấn công được ,gọi là **Khai thác SSH thông qua khóa bí mật yếu**. Phần này tương tự và triển khai theo **Tấn công Dictionary lấy mật khẩu SSH** ở trên

Người đọc có thể tham khảo các tạo khóa bí mật ở link này:

<https://vietnix.vn/tao-ssh-key/>

**3.4. Điều khiển nhiều botnet và thực hiện DDOS**

Trong phần 3.2 ,em đã thiết kế ra 1 class BotMaster ,class này sẽ là trung tâm ,điều khiển và đưa ra câu lệnh thực thi cho các botnet

**Chuẩn bị:**

* + 1 máy kali
  + 2 máy windowns 10
  + 1 file ‘dictionary.txt’ chứa nội dung IP ,Usernane,Password như hình dưới đây :

**Text

Description automatically generated**

**Thực thi class BotMaster**

**Graphical user interface, text

Description automatically generatedText

Description automatically generated**

Kết quả

Trả về một tool như hình dưới đây :

A screenshot of a computer

Description automatically generated with medium confidence

Bấm 1 thì ta có thể xem danh sách botnet khả dụng ,đang bị chiếm quyền :

Text

Description automatically generated

Kết quả trả về đó là 2 botnet :

[192.168.253.129,haikma2,87654321]

[192.168.253.131,haikma1,12345678]

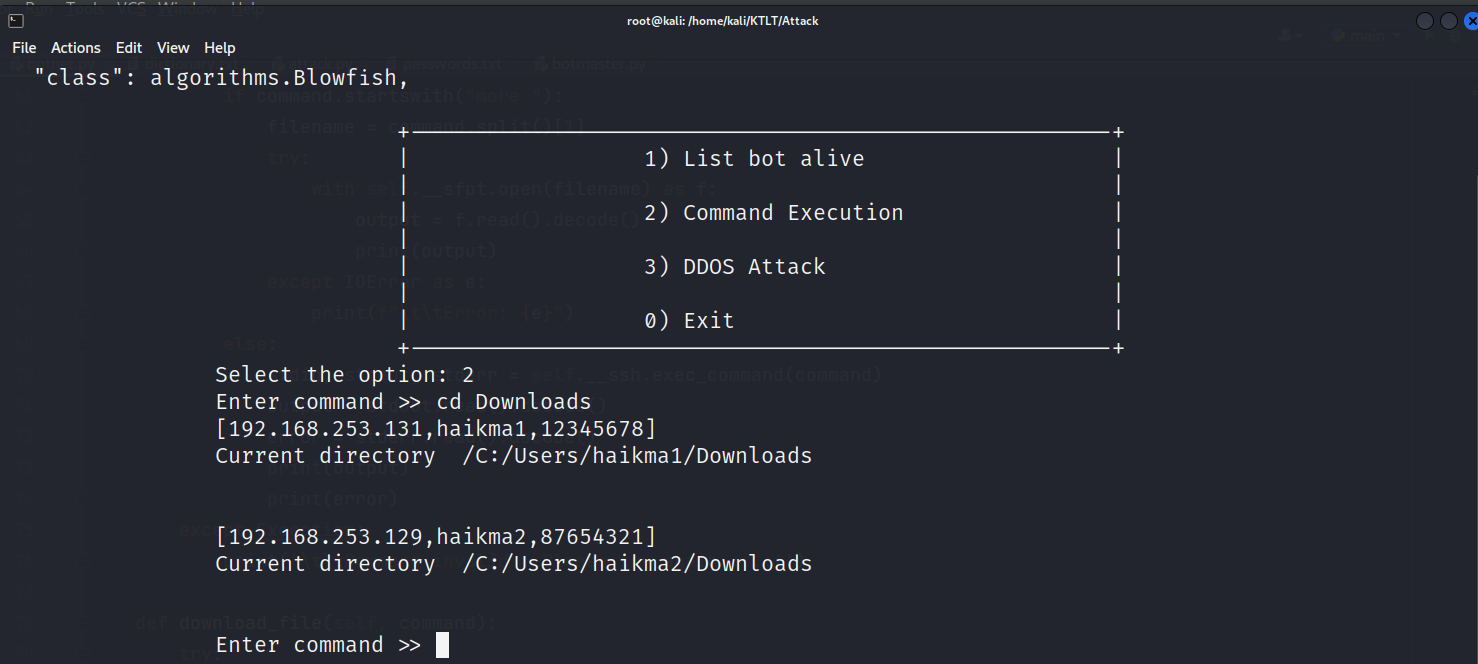
**Thực thiện điều khiển nhiều Botnet**

Bấm 2 ,thực hiện điều khiển nhiều botnet cùng lúc :

Text

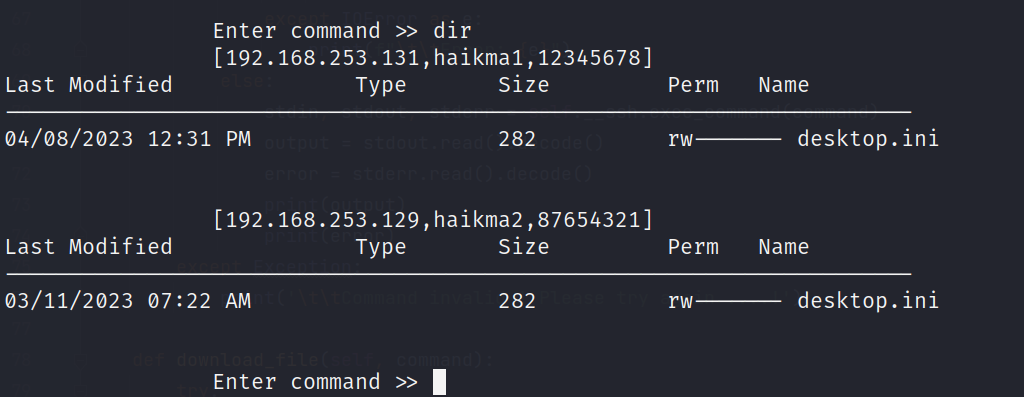
Description automatically generated

Thực hiện câu lệnh ‘cd Downloads’ và được kết quả :



Thông báo đường dẫn của 2 máy nạn nhân

Giờ thực hiện câu lệnh ‘dir’ để xem danh sách file với thư mục hiện có



Kết quả trả về một danh sách gồm file cùng với các quyền,thời gian …

**Chú ý** ,2 máy hiện chưa có file nào là ‘car.jpg’

Giờ thực hiện câu lệnh em thiết kế ,tải file ‘car.jpg’ lên tất cả máy botnet

Text

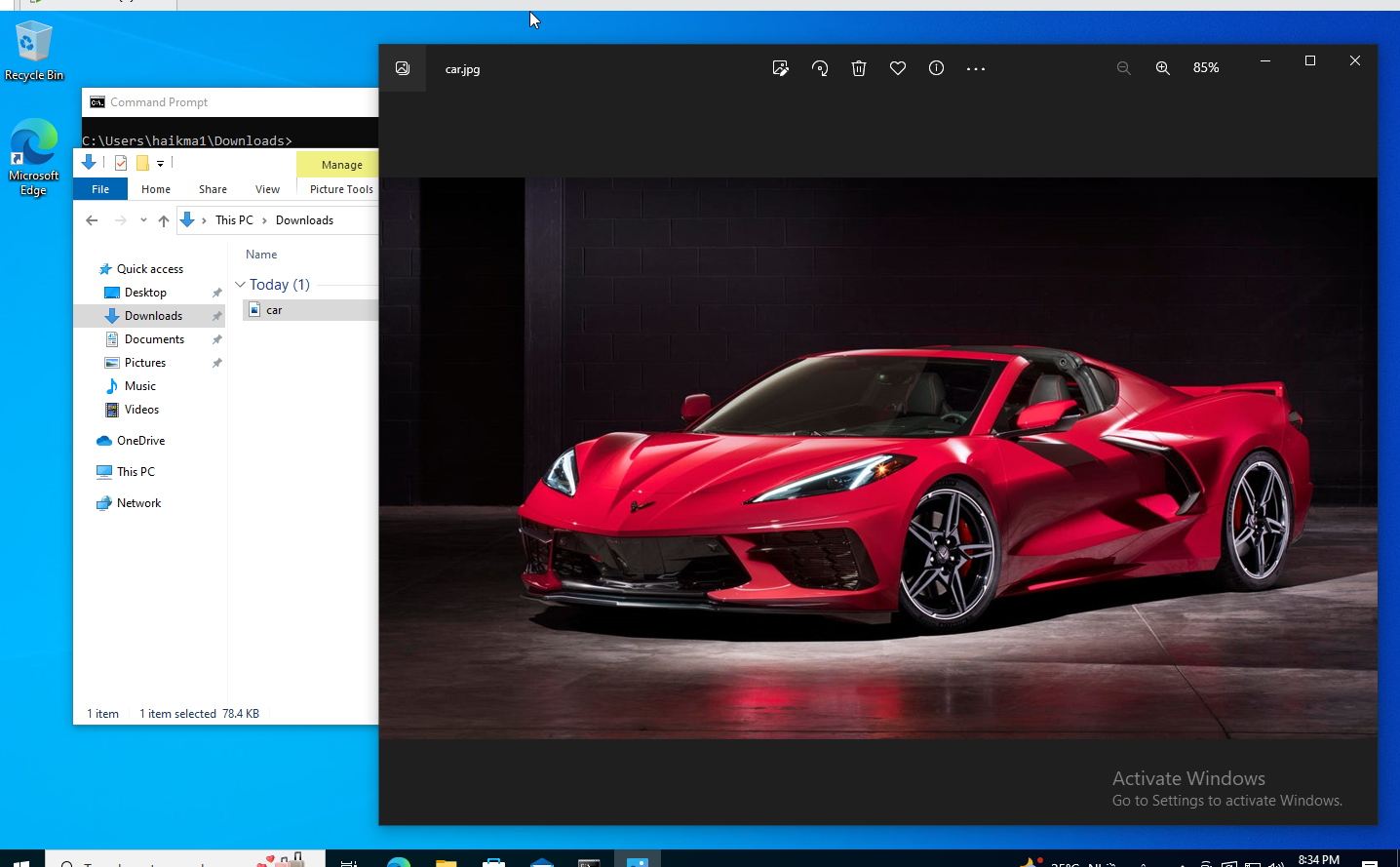
Description automatically generated

Thông báo kết quả tải file thành công ,giờ xem kết quả ,gõ ‘dir’

A screenshot of a computer

Description automatically generated with medium confidence

Kết quả cho thấy ,file ‘car.jpg’ đã được upload thành công .Giờ xem máy nạn nhân ,có xem được file không nào ?



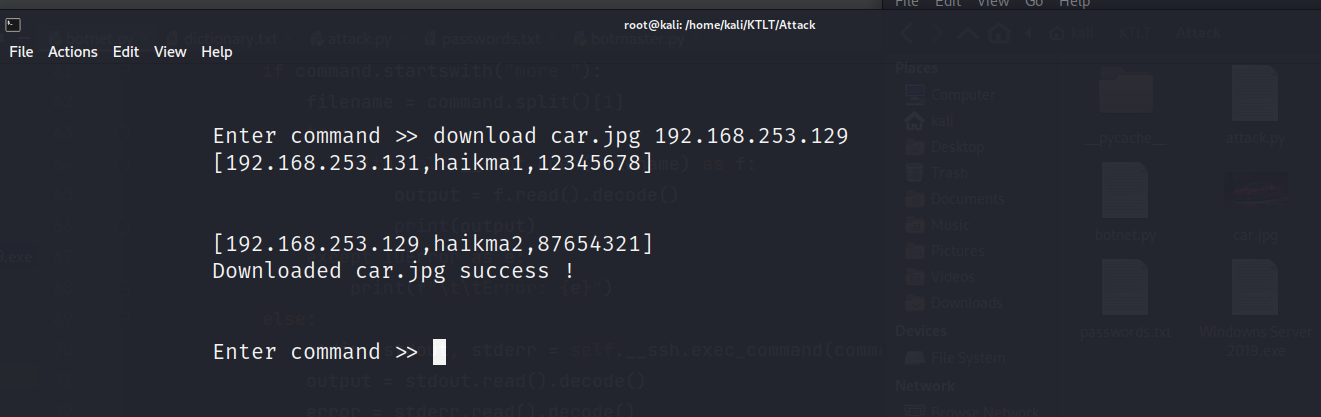
Và đây là kết quả trả về !

Trong code em còn viết chức năng ‘download file’ có nghĩa là download load bất kì file nào từ máy nạn nhân về máy của mình. Giờ thực hiện :

Graphical user interface

Description automatically generated

Để ý thưc mục này chưa có file nào là ‘car.jpg’ ,giờ thực thi câu lệnh



Download file car.jpg từ máy có địa chỉ ip 192.168.253.129

Thông báo từ máy botnet là download thành công

A screenshot of a computer

Description automatically generated with medium confidence

Kết quả là download về máy điều khiển thành công ,và xem được file

Vậy ,giả dụ tải lên một malware ,một keylogger ,một trojan … thì mọi hoạt động của máy nạn nhân ta sẽ biết. Đánh cắp được thông tin quan trọng và gồm nhiều thứ khác nữa …

**Thực hiện tấn công DDOS**

Màn hình tool trên ,giờ bấm 3 để các máy thực hiện DDOS

Giời list bot có 2 máy đó là 192.168.253.129 và 192.168.253.131

Thực tế ,phải cần nhiều máy thực hiện để tấn công thành công ,dưới đây em xin demo như sau :



Thực hiện câu lệnh ,gửi yêu cầu đến trang web : dantri.com.vn

Text

Description automatically generated

Kết quả trả về gửi yêu cầu thành công ,giờ kiểm thử lại bằng WireShrak ,xem các máy có gửi gói tin và nhận gói tin không

A picture containing table

Description automatically generated

Nhìn thấy ,hai máy có địa chỉ ip là 192.168.253.129 và 192.168.253.131 có gửi gói tin yêu cầu đến trang web : dantri.com.vn (42.113.206.26) chứng tỏ ,thực nghiệm đã thành công.

Thực tế có nhiều botnet gửi yêu cầu ,thì tình trạng tắc nghẽn của trang web xảy ra.

# CHƯƠNG IV. KẾT LUẬN

Với đề tài SSH Botnet, sau quá trình tìm hiểu, thực hành và báo cáo, phần nào hiểu được khái niệm, nguyên lý hoạt động, đặc điểm, sự nguy hiểm của tấn công Botnet, nhóm em nhận thấy sự phát triển nhanh chóng của công nghệ đặc biệt là trong lĩnh vực Internet, điều đó mang đến rất nhiều lợi ích to lớn trong cuộc sống. Song hành cùng những lợi ích vô cùng to lớn đó là những rủi ro về nguy cơ bị tấn công bởi hacker, bị theo dõi và đánh cắp thông, tin và nguy hiểm hơn cả là bị lợi dùng để làm những điều trái pháp luật. SSH Botnet có thể mang đến những lợi ích vô cùng to lớn có thể kể đến như việc nghiên cứu về mã gen của con người, nếu như một máy phải làm trong 10000 giờ thì 1000 máy con số ấy sẽ chỉ là 10 giờ - chỉ ngang một giấc ngủ. Tuy nhiên nếu như điều ấy được ứng dụng trong việc tấn công và làm những việc sai trái thì hậu quả sữ vô cùng khủng khiếp. Qua quá trình nghiên cứu đề tài, nhóm đã có dịp tiếp cận những khái niệm, công cụ, phần mềm, được thực nghiệm trên chính máy tính của mình. Điều đó giúp nhóm hiểu và biết nhiều về các hình thức tấn công, nguyên lý hoạt động để qua đó có kinh nghiệm phòng tránh.

**Về cơ bản nhóm đã được những mục tiêu đề ra khi triển khai đề tài:**

* Hiểu được cơ bản về SSH Botnet (SSH, Botnet)
* Làm quen được ngôn ngữ lập trình Python
* Làm quen với môi trường và tiến hành thiếp lập môi trường SSH Botnet
* Được tiếp xúc và tìm hiều các cách thức tấn công khác nhau
* Được tìm hiểu chung về mạng, cổng mạng
* Hiểu sâu hơn về khái niệm DDOS
* Được vận dụng những kiến thức đã được học đề thực hành, tạo cơ sở, kỹ năng phục vụ làm việc sau này
* Các nhóm nâng cao ký năng làm việc nhóm, lắng nghe và tiếp thu, tổng hợp ý kiến, nâng cao khả năng về xử lý word, powperpoint, lập trình,..
* Tạo mỗi quan hệ với những người lạ, làm việc với người lạ
* Có trách nghiệm với công việc đươc giao, phải cố gắng hoàn thành và rút ra được những bài học riêng cho bản thân.

# TÀI LIỆU THAM KHẢO ĐỂ LÀM ĐỀ CƯƠNG

[1]. <https://quantrimang.com/botnet-hoat-dong-nhu-the-nao-36773>

[2]. <https://vietnetco.vn/botnet-la-gi-giai-phap-phong-ve-botnet-fortiguard/4999.html>

[3].<https://fptcloud.com/ssh-la-gi/>

[4].<https://anonyviet.com/cach-tao-botnet-ddos-don-gian-bang-py-botnet/>

[5] <https://www.tutorialspoint.com/build-your-own-botnet>

[6] <https://www.techtarget.com/searchsecurity/definition/botnet>

[7] <https://pexpect.readthedocs.io/en/stable/overview.html>

[8]<https://www.youtube.com/watch?v=WELvXrXJeUk&t=207s>

[9]<https://pexpect.readthedocs.io/en/stable/>

[10]<https://stackoverflow.com/questions/6188970/how-to-make-a-ssh-connection-with-python>

[11] <https://vietnix.vn/tao-ssh-key/>

[12] <https://docs.paramiko.org/en/stable/api/client.html>

[13] <https://vinasupport.com/python-3-ket-noi-ssh-chay-command-tren-linux-server/>

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| **MSV** | **Họ và tên** | **Phân chia công việc** | **Đánh giá** |
| AT170415 | Nguyễn Văn Hải | * Làm báo cáo word * Tìm hiểu nội dung * Viết code thực nghiệm * Chỉnh sửa PPT | Đã Hoàn Thành |
| AT170346 | Lâm Tăng Thành | * Sửa báo cáo word * Tìm hiểu nội dung * Viết code thực nghiệm * Chỉnh sửa PPT | Đã Hoàn Thành |
| AT170435 | Đàm Ngọc Nam | * Sửa báo cáo word * Tìm hiểu nội dung * Viết code thực nghiệm * Chỉnh sửa PPT | Đã Hoàn Thành |
| AT170317 | Trần Gia Hiển | * Sửa báo cáo word * Tìm hiểu nội dung * Viết code thực nghiệm * Làm PPT | Đã Hoàn Thành |