Analysis of commands of Telnet logs illegally connected to IoT devices

Toshihiro Yamauchi
Graduate School of Natural Science and Technology
Okayama University, Okayama, Japan / JST PRESTO

Ryota Yoshimoto, Takahiro Baba
Graduate School of Natural Science and Technology
Okayama University, Okayama, Japan

Katsunari Yoshioka
Graduate School of Environment and Information Sciences / Institute of Advanced Sciences
Yokohama National University, Kanagawa, Japan

***Abstract*—Mirai is an active malware that targets and poses constant threats to IoT devices. IoT malware penetrates IoT devices illegally, makes them download other malware such as bots, and infects them. Therefore, to improve the security of IoT devices, it is important to analyze the behaviors of IoT malware and take countermeasures. In this study, to analyze the behaviors of IoT malware after entering IoT devices and propose new security functions for operating systems to prevent activities such as IoT malware infection, we analyze Telnet logs collected by a honeypot of IoT devices. Thereafter, we report the analysis results regarding IoT malware input commands. The results show that many commands related to shell execution, file download, changing file permissions, and file transfer, are often executed by IoT malware.**

***Index Terms*—IoT, malware, Telnet log**

I. Introduction

Malware that target IoT devices is active, and IoT devices are under constant threat [1]. Mirai [2] is a malware that turns networked devices running on Linux into remotely controlled bots that can be used as part of a botnet in large-scale network attacks. It penetrates IoT devices using application protocols such as Telnet and targets easily guessable passwords. When malware such as Mirai succeed in logging into such IoT devices through Telnet, it downloads other malware and executes commands that achieve its purpose, turning IoT devices into bots. IoT devices converted into bots sometimes try to penetrate other IoT devices using the same method to spread the infection. In addition, attackers sometimes perform distributed denial-of-service attacks using botnets created using such processes. Therefore, to make IoT devices more secure, it is important to analyze the behaviors of IoT malware and take countermeasures. In this paper, we report the analysis results of commands in execution logs when IoT devices are logged in by means of Telnet.

II. Analysis of Telnet logs

A. Purpose

The purpose of this study is to propose security functions at the level of the operating system (OS) to prevent the infecting activities of IoT malware and achievement of the attack objectives after infection. To accomplish this purpose, it is necessary to analyze the behaviors of IoT malware after they enter IoT devices. Therefore, we analyze these behaviors of IoT malware via Telnet.

Part of this research was supported by JST PRESTO, JPMJPR1938, and JSPS Grants-in-Aid for Scientific Research JP19H04111.

Several previous studies have reported analyses of IoT malware. IoTPOT [3] analyzed the five malware families that target Telnet-enabled IoT devices and reported the infection behaviors, behaviors during DDoS attacks, and their Botnet architecture. Conversely, our study focuses on the command sequence of the Telnet log to clarify the behaviors of IoT malware during infection and malware download as well as to identify the functions required for security mechanisms to prevent such behaviors. In another work [4], malware classification was achieved by focusing on its binary and command sequences; however, the present study is different because we analyze the command sequences for OS-level security functions, such as access control.

B. Summary of Telnet logs

In this study, we used Telnet logs collected using a honeypot of IoT devices (IoTPOT [3]) . This honeypot was operated with an open Telnet port. It records Telnet commands when connected from the Internet by means of Telnet. Telnet logs from September 1 to 7, 2017 were analyzed. Approximately 3.8 million logs were collected in 7 days.

C. Analysis of executed commands

To analyze frequently executed commands, we investigated the number of times they appeared in logs. Table I lists the top 30 commands with the highest number of occurrences.

Table I shows the following.

1) Several commands were executed by busybox.
2) There are many commands to execute the shell.
3) The internal command "`enable`" was executed several times.
4) There are many commands that can download, change permissions, or delete files.
5) Shell scripts have been executed many times.

TABLE I
COMMANDS WITH THE HIGHEST NUMBER OF OCCURRENCES (TOP 30)

Command line	**Appearances**
/bin/busybox BAT	473,792
sh	173,762
shell	172,186
/bin/busybox ECCHI	169,204
system	111,822
enable	68,004
/bin/busybox kill -9 2232	44,150
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /	42,179
bin/busybox rm /dev/.nippon	38,848
/bin/busybox cat /dev/.nippon	38,840
/bin/busybox echo -e '\\x6b\\x61\\x6d\\x69/dev' > /dev/.nippon	38,303
sh ftp1.sh	36,575
chmod 777 tftp1.sh	34,445
sh tftp1.sh	34,445
sh tftp2.sh	34,445
chmod 777 tftp2.sh	34,411
rm -rf *	33,032
chmod 777 cunty.sh	32,059
sh cunty.sh	32,059
/bin/busybox wget	30,403
/bin/busybox tftp	30,403
rm -rf cunty.sh tftp1.sh tftp2.sh ftp1.sh	29,929
/bin/busybox cat /.nippon	29,455
/bin/busybox rm /.nippon	29,455
/bin/busybox MIRAI	29,104
/bin/busybox echo -e '\\x6b\\x61\\x6d\\x69' > /.nippon	29,048
>elfLoad	27,777
/bin/busybox echo -ne '\\x0181c46036\\x01' || /bin/busybox echo -ne '\\x0281c46036\\x01'	19,742
/bin/busybox ps	19,453
/bin/busybox cat /proc/mounts	18,717

From these observations, it can be conjectured that they download some type of files, execute them, and delete them after logging in.

D. Analysis of names of executed commands

The number of appearances of only names of commands excepting arguments from the command line were investigated. The number of appearances that include "busybox" in the first argument and its second argument were investigated. Table II lists the top 30 command names with the highest number of occurrences.

Along with Table I, Table II also shows several commands regarding the use of busybox, changing file permissions, deleting files, and execution of the shell. Based on these results, we display the commands that call file names under "Commands" in Table III. Table III shows many commands that execute the shell, download files, change file permissions, and transfer files.

TABLE II
NAMES OF COMMANDS WITH THE HIGHEST NUMBER OF OCCURRENCES (TOP 30)

Command name	**Appearances**
/bin/busybox BAT	473,792
echo	399,583
sh	327,915
/bin/busybox cat	311,704
/bin/busybox rm	308,515
/bin/busybox echo	288,706
shell	172,186
/bin/busybox ECCHI	169,204
rm	164,416
chmod	124,659
system	111,822
tftp	80,916
enable	68,004
/bin/busybox kill	66,760
/bin/busybox chmod	61,423
cd	53,623
/bin/busybox cp	47,922
/bin/busybox wget	45,076
wget	42,745
ftpget	38,314
/bin/busybox tftp	30,403
/bin/busybox MIRAI	29,104
>elfLoad	27,777
/bin/busybox ps	19,453
Do	18,779
done	18,779
>elfDrp	18,341
cat	12,679
9/dev/mqueue'	11,945
>/tmp/.ptmx	10,852

TABLE III
THE NUMBERS OF APPEARANCES OF COMMANDS EXIST IN LINUX

Command	**Appearances**
sh	327,915
rm	164,416
chmod	124,659
tftp	80,916
wget	42,745
ftpget	38,314
cat	12,679
/bin/echo	5,805
/usr/bin/printf	5,805
ping	400
dd	209
service	158
cp	60
telnet	37
fgrep	30
curl	4

E. Analysis of order of executing commands

To analyze the order of executing commands, we concatenated names of commands and investigated those that appear often. Based on our analysis of the research results, commands were often executed in the following four orders.

- enable → system → shell → sh
- enable → shell → sh
- system → shell → sh
- shell → sh

`enable` is a command used to allow access to privileged-mode commands, and `system` is a command used to navigate to a menu of system-management options. `shell` and `sh` are commands that execute a Bourne shell. These commands are executed in a series to access the Linux shell [5].

III. Conclusion

We studied commands used by malware and the orders in which they are executed by analyzing Telnet logs of an IoT honeypot. After IoT malware logs in to an IoT device, it often operates files, downloads malware, and executes it. Thus, to prevent malicious activities in IoT devices, we should focus on preventing the executing of such malicious commands.

In the future, we will consider security mechanisms to protect IoT devices from being exploited by malware that logs in to IoT devices illegally.

References

[1] B. Vignau, R. Khoury and S. Hallé, “10 years of iot malware: A feature-based taxonomy,” *In Proceedings of the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp.458–465, 2019.

[2] Antonakakis, M. et al., “Understanding the mirai botnet,” *In Proceedings of 26th USENIX Conference on Security Symposium*, pp.1093–1110, 2017.

[3] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama and Christian Rossow, “IoTPOT: A Novel Honeypot for Revealing Current IoT Threats,” *Journal of Information Processing*, Vol.24, Issue 3, pp.522–533, 2016.

[4] C.-J. Wu, S.-Y. Huang, K. Yoshioka and T. Matsumoto, “IoT Malware Analysis and New Pattern Discovery Through Sequence Analysis Using Meta-Feature Information,” *IEICE Transactions on Communications*, Vol.E103.B, Issue 1, pp.32–42, 2020.

[5] Sam Edwards, Ioannis Profetis, “Hajime: Analysis of a decentralized internet worm for IoT devices,” Tech. Rep. 1, Rapidity Networks, 2016, URL https://security.rapiditynetworks.com/publications/2016-10-16/hajime.pdf.