New RapperBot Campaign – We Know What You Bruting for this Time

fortinet.com/blog/threat-research/new-rapperbot-campaign-ddos-attacks

November 15, 2022

After FortiGuard Labs reported on RapperBot in our previous article titled So RapperBot, What Ya Bruting For? in August 2022, there was a significant drop in the number of samples collected in the wild. But in early October 2022, new samples with the same distinctive C2 protocol used by RapperBot were detected.

Unlike the murky objectives of the previous campaign, it is quickly evident that these samples are part of a separate campaign to launch Distributed Denial of Service (DDoS) attacks against game servers, which we believe to be a re-emergence of a similar campaign from earlier this year.

Affected Platforms: Linux
Impacted Users: Any organization
Impact: Remote attackers gain control of the vulnerable systems
Severity Level: Critical

This article discusses the differences observed in this campaign and its relation to the previous RapperBot and similar campaigns in the past.

RapperBot Rebooted

FortiGuard Labs encountered this campaign by hunting for samples using the unique bot ID used by RapperBot to communicate with its Command-and-Control (C2) server, as reported in the previous article.

But once we analyzed these new samples, we observed a significant difference between them and the earlier campaign. In fact, it turns out that this campaign is less like RapperBot than an older campaign that appeared in February and then mysteriously disappeared in the middle of April. Other related campaigns uncovered during this investigation are detailed later in this article.

Network Protocol and Denial-of-Service (DoS) Attacks

The C2 network protocol used in previous campaigns remains essentially unchanged, with additional commands added to support the Telnet brute force. The list of commands and IDs are shown below:

- **0x00**: Register (used by the client)
- **0x01**: Keep-Alive/Do nothing
- **0x02**: Stop all DoS attacks and terminate the client
- **0x03**: Perform a DoS attack
- **0x04**: Stop all DoS attacks
- **0x06**: Restart Telnet brute forcing
- **0x07**: Stop Telnet brute forcing

The previously reported RapperBot campaign was limited to a few generic DoS methods against TCP and UDP services. This campaign adds DoS attacks against the GRE protocol (likely reusing the Mirai source code) and the UDP protocol used by the Grand Theft Auto: San Andreas Multi Player (SA:MP) mod.

Here are the DoS attack commands supported by this botnet:

- **0x00**: Generic UDP flood
- **0x01**: TCP SYN flood
- **0x02**: TCP ACK flood
- **0x03**: TCP STOMP flood
- **0x04**: UDP SA:MP flood targeting game servers running GTA San Andreas: Multi Player (SA:MP)
- **0x05**: GRE Ethernet flood
- **0x06**: GRE IP flood

- **0x07**: Generic TCP flood

These specific commands, coupled with the absence of HTTP-related DDoS attacks, suggests that this campaign is primarily geared toward game server DDoS.

Telnet Self-propagation

The most significant difference in the new campaign was the complete replacement of the SSH brute forcing code with the more usual Telnet equivalent. FortiGuard Labs has observed similar drastic modifications within RapperBot samples, as detailed in our previous report, adding and removing even DoS attack code on an apparent whim.

The Telnet brute forcing code is designed primarily for self-propagation and resembles the old Mirai Satori botnet. Unlike the earlier SSH brute-forcing campaign, the plaintext credentials are embedded into the malware instead of being downloaded from the C2.

Figure 1. Function initializing the credential list

These credentials used appear to be default credentials for IoT devices. To optimize brute forcing efforts, the malware compares the server prompt upon connection to a hardcoded list of strings to identify the possible device and then only tries the known credentials for that device. Unlike less sophisticated IoT malware, this allows the malware to avoid trying to test a full list of credentials. While not exactly a novel technique, it is still uncommon compared to other IoT botnets.

Based on the prompt messages hardcoded into the malware, most of the targeted devices are IoT devices such as routers and DVRs. This campaign seems especially interested in older devices with the Qualcomm MDM9625 chipset, such as LTE modems. It attempts to specifically gain root access to these devices via a default password, despite having the same credentials in the list embedded in the binary.

Figure 2. Gaining root access on devices with a default password

Like the earlier SSH brute-forcing campaign, once it has successfully gained access, it sends the credentials used, the compromised device's IP address, and its architecture to the C2 server on a separate port, 5123. After reporting, the malware attempts to install its main payload binary on the compromised device.

It first parses the Executable and Linkable Format (ELF) header of the */bin/busybox* file for the *e_machine* field, which provides the architecture of the compromised device. This allows it to download and deploy a RapperBot payload of the correct architecture to ensure proper execution. This selective behavior is more efficient than the shotgun approach in most IoT malware families, whereby all the binaries for the supported architectures are downloaded and executed in the victim's system.

Based on the payload binaries we collected, this botnet currently seems to only target devices running on ARM, MIPS, PowerPC, SH4, and SPARC architectures. Moreover, it specifically checks and stops its self-propagation if the device is detected to be running on Intel processors.

The bot then downloads its payload via software installed on the compromised device, such as *ftpget*, *wget*, *curl,* or *tftp,* before executing the payload.

Figure 3. Downloading the payload binary using the wget tool

If none of the software mentioned above is installed, it will extract and send an embedded binary downloader to the compromised device that executes and downloads the primary payload.

Unlike in Satori, these embedded downloaders are stored as escaped byte strings, probably to simplify parsing and processing within the code.

Figure 4. List of embedded binary downloaders

The binary downloaders are written by echoing the bytes and piping the content to a file in the victim system. As labeled in Figure 4, each binary has a hardcoded URL for downloading the payload binary of the proper architecture.

Figure 5. Writing downloader binary and executing it

No attempts to persist on infected or brute-forced devices were observed for this campaign.

Related Campaigns

FortiGuard Labs compared samples for this and related campaigns from the past to find any links with the previously reported RapperBot campaign.

We observed that the earliest samples for this campaign were from December 2021 and that the SA:MP attack was only added in February 2022. This campaign mysteriously disappeared in mid-April 2022, resurfacing in Oct 2022 with the addition of the self-propagation feature.

We also found older samples from another campaign that was active in August-September 2021 with an almost identical list of credentials. These samples contain slightly fewer credentials and a simpler self-propagation code that only supports downloading the payload via wget or the binary downloader embedded directly into the sample. This campaign did not support stopping or restarting the Telnet propagation, and while the samples support the same commands, their associated IDs did not match.

Figure 6. Timeline of related campaigns

The similar lists of credentials suggest that the threat actor behind this current campaign has access to the source code for the earlier campaign, as this code was not found in other IoT malware samples.

Connections to RapperBot

The fact that samples from both campaigns use the same C2 protocol, coupled with the absence of this campaign during the RapperBot campaign active between June and Aug 2022 and its recent reappearance, seems to be more than a coincidence.

With the several similarities between the two campaigns outlined below, we believe that either the same threat actor might be behind both campaigns or each campaign might have branched from the same privately-shared source code.

1. The C2 commands and corresponding IDs are identical in both campaigns (excluding the Telnet-related commands, as those do not apply to RapperBot)
2. Both campaigns show a certain degree of effort in optimizing the brute forcing implementation. Code for the brute forcing implementation is significantly more structured than typical IoT malware that copies and pastes code with minimal modifications.
3. RapperBot also supported the TCP STOMP attack popularized by Mirai. This attack was not observed in the earlier campaigns mentioned above. However, as both Mirai and Satori source code are publicly available, this is considered a very weak link between the campaigns.

If both campaigns were related, the reason for restarting an older campaign remains a mystery.

Conclusion

Based on the undeniable similarities between this new campaign and the previously reported RapperBot campaign, it is highly likely that they are being operated by a single threat actor or by different threat actors with access to a privately-shared base source code.

Unlike the previous RapperBot campaign, this new campaign has a clear motivation to compromise as many IoT devices as possible to build a DDoS botnet.

Although this new campaign has evolved significantly from previous campaigns, mitigating it remains the same—setting strong passwords for all devices connected to the internet.

FortiGuard Labs will continue to monitor RapperBot's development.

Fortinet Protections

The FortiGuard Antivirus service detects and blocks this threat as **ELF/Mirai**, **Linux/Mirai**, and **ELF/Gafgyt**.

The FortiGuard AntiVirus service is supported by FortiGate, FortiMail, FortiClient, and FortiEDR, and the Fortinet AntiVirus engine is a part of each of those solutions. Customers running current AntiVirus updates are protected.

FortiGuard Labs provides the Rapper.Botnet IPS signature against RapperBot C2 activity.

The FortiGuard Web Filtering Service blocks the C2 servers and download URLs.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

IOCs

Files

3d5c5d9e792e0a5f3648438b7510b284f924ab433f08d558b6e082e1d5414a03
7afcac5f71e9205879e0e476d3388898a62e7aa4a3e4a059884f40ea36cfd57f
8ec79a35700f6691f0d88d53647e9f2b75648710ecd119e55815331fc3bdd0b5
a12ad4bc394d60bc037271e1c2df1bd2b87bdaaba85f6c1b7d046341f027cc2d
f000bf482040b48595badee1fc56afb95449ac48b5dc35fe3a05542cbf18f658
4aa9175c1846557107ec197ea73d4cc8dbe6d575a8fd86ae214ff9b3a00e438b
f98261eb7dc122449c158118cc9c660683206983a9e90ff73eb88c4705e0c48e

Download URLs

hxxp://185[.]216[.]71[.]149/armv4l

hxxp://185[.]216[.]71[.]149/armv5l

hxxp://185[.]216[.]71[.]149/armv6l

hxxp://185[.]216[.]71[.]149/armv7l

hxxp://185[.]216[.]71[.]149/mips

hxxp://185[.]216[.]71[.]149/mipsel

hxxp://185[.]216[.]71[.]149/powerpc

hxxp://185[.]216[.]71[.]149/sparc

hxxp://185[.]216[.]71[.]149/sh4

hxxp://185[.]216[.]71[.]149/bot_arm4_el

hxxp://185[.]216[.]71[.]149/bot_arm5_el

hxxp://185[.]216[.]71[.]149/bot_arm6_el

hxxp://185[.]216[.]71[.]149/bot_arm7_el

hxxp://185[.]216[.]71[.]149/bot_mips_eb

hxxp://185[.]216[.]71[.]149/bot_mips_el

hxxp://185[.]216[.]71[.]149/bot_sh_el

C2

185[.]216[.]71[.]149

Learn more about Fortinet's FortiGuard Labs threat research and global intelligence organization and Fortinet's FortiGuard AI-powered Security Services portfolio. Sign up to receive our threat research blogs.