Suspected Chinese Threat Actors Exploiting FortiOS Vulnerability (CVE-2022-42475) | Mandiant

By Mandiant

Published: 2023-01-19 · Archived: 2026-04-05 20:31:33 UTC

Written by: Scott Henderson, Cristiana Kittner, Sarah Hawley, Mark Lechtik

---

Mandiant is tracking a suspected China-nexus campaign believed to have exploited a recently announced vulnerability in Fortinet's FortiOS SSL-VPN, CVE-2022-42475, as a zero-day. Evidence suggests the exploitation was occurring as early as October 2022 and identified targets include a European government entity and a managed service provider located in Africa.

Mandiant identified a new malware we are tracking as “BOLDMOVE” as part of our investigation. We have uncovered a Windows variant of BOLDMOVE and a Linux variant, which is specifically designed to run on FortiGate Firewalls. We believe that this is the latest in a series of Chinese cyber espionage operations that have targeted internet-facing devices and we anticipate this tactic will continue to be the intrusion vector of choice for well-resourced Chinese groups.

On December 12, 2022, Fortinet released a PSIRT Advisory and notified customers regarding CVE-2022-42475

- Fortinet issued instructions on how to search for Indicators of Compromise
- Fortinet provided additional details including IoCs from subsequent research.

China Continues to Focus on Network Devices

This incident continues China’s pattern of exploiting internet facing devices, specifically those used for managed security purposes (e.g., firewalls, IPS\IDS appliances etc.). These devices are attractive targets for multiple reasons. First, they are accessible to the internet, and if the attacker has an exploit, they can gain access to a network without requiring any victim interaction. This allows the attacker to control the timing of the operation and can decrease the chances of detection.

The exploits required to compromise these devices can be resource intensive to develop, and thus they are most often used in operations against hardened and high priority targets; often in the government and defense sectors. With BOLDMOVE, the attackers not only developed an exploit, but malware that shows an in-depth understanding of systems, services, logging, and undocumented proprietary formats. Malware running on an internet-connected device can enable lateral movement further into a network and enable command and control (C2) by tunneling commands in and data out of a network.

It is important to note that many of these types of devices do not offer a simple mechanism to view which processes are running on the device’s operating systems. These devices are typically intended to inspect network

traffic, searching for anomalies as well as signs of malicious behavior, but are often not inherently protected themselves.

- Managed devices may provide only a limited admin interface that allows configuration and viewing/collection of logs
- Managed devices may not allow for additional security products, such as Endpoint Detection and Response (EDR) to be installed
- Access to core security features may be limited to the device manufacturer

Previous examples of public reporting by Mandiant and others on operations targeting these devices are here:

- Check Your Pulse: Suspected APT Actors Leverage Authentication Bypass Techniques and Pulse Secure Zero-Day
- NSA | APT5: Citrix ADC Threat Hunting Guidance
- Suspected Chinese Activity Exploiting Zero-Day Vulnerability, Leverages New Malware Designed for Internet-Facing Devices
- Zero-Days Exploit in SonicWall Email Security Lead to Enterprise Compromise

BOLDMOVE Backdoor

In December 2022, Mandiant identified the BOLDMOVE backdoor associated with the exploitation of CVE-2022-49475 FortiOS vulnerability. BOLDMOVE is written in C and has both Windows and Linux variants, the latter of which is intended to run (at least in part) on Fortinet devices as it reads data from a file proprietary to Fortinet.

Mandiant has not directly observed exploitation of the vulnerability; however, samples of the BOLDMOVE Linux variant have a hard coded C2 IP address that were listed by Fortinet as being involved in the exploitation, suggesting CVE-2022-49475 was exploited to deliver BOLDMOVE. In addition to the Linux variant, Mandiant also revealed a Windows version. Windows versions of BOLDMOVE appear to have been compiled as early as 2021. However, Mandiant has not seen this malware in use in the wild so it is uncertain how it was used. In-depth analysis of the malware is provided later in this post.

Attribution

We assess with low confidence that this operation has a nexus to the People’s Republic of China. China-nexus clusters have historically shown significant interest in targeting networking devices and manipulating the operating system or underlying software which supports these devices. In addition, the geographical and sector targeting is consistent with previous Chinese operations.

- Limited technical indicators point to the development of the malware as having been compiled on a machine in the UTC+8 time zone, which includes Australia, China, Russia, Singapore, and other Eastern Asian countries, and on a machine configured to display Chinese characters.
- A host survey buffer which is used by the Windows variant of BOLDMOVE in order to provide the C2 with information on the infected host starts with the string “gbk”. The comparable survey buffer of the

Linux variant starts with “utf-8”, which indicates that this field designates character encoding. If we are to consider “gbk” in this context, then this is an extension of a Chinese character set

- The exploitation of zero-day vulnerabilities in networking devices, followed by the installation of custom implants, is consistent with previous Chinese exploitation of networking devices.

Mandiant has previously reported on significant campaigns impacting networking devices, likely revealing a long-standing interest by China to embed cyber campaigns in the overarching telecommunications and networking architecture used by organizations worldwide:

- In April 2021, Mandiant reported extensively on the exploitation of Pulse Secure. Mandiant recently responded to multiple security incidents involving compromises of Pulse Secure VPN appliances.
- In March 2021, Mandiant identified three zero-day vulnerabilities in SonicWall’s Email Security (ES) product that were being exploited in the wild. Mandiant’s investigations informed us that the adversary leveraged these vulnerabilities, with intimate knowledge of the SonicWall application, to install a backdoor, access files and emails, and move laterally into the victim organization’s network.

Outlook

Mandiant has produced in depth reporting on the growing number of managed, internet-facing and connected devices targeted by Chinese threat actors. This latest campaign may be a continuation of a long-standing practice by China-nexus cyber espionage actors. This campaign and infection vector also should be strong reminders of the importance of keeping up with updates and patches, of externally facing devices or those exposed to the internet.

This campaign, and other similar campaigns, offer defenders a unique look into the vulnerabilities and gaps many organizations constantly face when services and networks are managed remotely. Given their configuration, it is very hard to measure the scope and extent of malicious activity that results from exploiting internet facing network devices, as we have little to no information that can indicate those devices are compromised.

There is no mechanism to detect malicious processes running on such devices, nor telemetry to proactively hunt for malicious images deployed on them following an exploitation of a vulnerability. This makes network devices a blind spot for security practitioners and allows attackers to hide in them and maintain stealth for long periods, while also using them to gain foothold in a targeted network.

BOLDMOVE Linux Analysis

BOLDMOVE is a fully featured backdoor written in C and compiled with GCC 11.2.1. When executed it performs a system survey and is capable of receiving commands from a C2 server that in turn allow attackers to control the file system, spawn a remote shell, or relay traffic via the infected host.

Based on indicators from the original Fortinet advisory, Mandiant was able to identify multiple Linux versions of BOLDMOVE. There are a core set of features across all observed instances of BOLDMOVE, Windows and Linux, and at least one Linux sample contained extended capabilities enabling it to alter specific behaviors and functionality of Fortinet devices, namely FortiGate Firewalls.

Core Features

Upon execution, BOLDMOVE attempts to form a session with a hard-coded C2 server. Once it is established, it performs a system survey to collect information that identifies the infected machine to the C2. Information collected is outlined in Table 1.

Index	Field Value
0	Encoding used for the strings in the survey buffer: utf-8
1	Hard-coded string that seemingly identifies the sample or campaign, e.g., “Cora/c”
2	OS version string. For Linux-based operating systems this string has the format “Linux []”, wherein the various fields are obtained from a call to the uname function. For non-Linux operating systems this string has the format []. The substring is being constructed by reading data from one of the files /etc/system-release, /etc/os-release (looking for the values of the NAME= and VERSION= keys),/migadmin/ng/vpn/map/pkginfo.json (looking for the value enclosed by the strings ver_s\":\" and \",\"chksum), /etc/debian_version.
3	Host name
4	Comma-separated list of / entries that represent network interfaces on the host
5	The effective user ID of the backdoor's process (result of geteuid())
6	The process ID of the backdoor's process
7	String of the format cwd=\r\nexecutable=\r\nevent=wv\r\nserver=139.180.128.142:443\r\n/proc/version=

Table 1: System Survey

Subsequently, the C2 may send commands for execution that allow attackers to control the infected device. Command codes across platforms and versions of BOLDMOVE may vary but their core capabilities do not appear to change and include:

Major Command Code	Minor Command Code	Command
0x0	0x0	Frees all resources and terminates the backdoor
0x11	0x21	Lists information on all files in the system recursively, starting from the root directory. In addition to the file's path, the information provided for each file is based on output of the stat function and includes the following fields of the stat structure (details on it can be found here): st_mode, st_size, st_mtim.tv_sec, st_uid, st_gid.
0x11	0x0	Lists information on files recursively, starting from a given directory

0x12	0x0	Creates new directory via mkdir
0x13	0x0	Removes a directory via rmdir
0x14	0x10	Given an attacker provided file path, removes an existing file (if such exists) and creates a new file instead
0x14	0x21	Closes a file descriptor that was opened for writing
0x14	0x32	Writes data to the created file
0x15	0x10	Gets a file's size before reading from it
0x15	0x21	Closes a file descriptor that was opened for reading
0x15	0x40	Reads data from a formerly opened file
0x20	0x0	Executes a shell command and sends back the output
0x20	0x33	Executes a shell command without sending back an output
0x21	0x10, 0x21, 0x43, 0x44, 0x45	Creates an interactive shell that leverages two pipes—one for processing shell input from the server and another for sending back shell outputs, thus supporting an asynchronous session between the C2 and the infected host. The various subcommands handle actions involved in forming and maintaining the shell session
0x22	0x10, 0x21, 0x32,0x33	Creates an interactive shell that leverages a single pipe for both passing server sourced inputs to the shell and retrieving command outputs from it. The formed shell works in a synchronous mode, wherein the pipe can be either probed to retrieve shell output or written with input data in each access to it. The various subcommands handle actions involved in forming and maintaining the shell session
0x30	0x15, 0x16, 0x17, 0x18	Initiates a network traffic relay session. The C2 sends a target address as an argument and further packets passed through sub-commands of this command are used to pass data back and forth to and from the target server
0x53	0x10	Deletes the backdoor's image and creates a new one with the same name as preparation for writing an updated backdoor image
0x53	0x21	Closes the file descriptor opened for writing a backdoor image update
0x53	0x32	Writes data sent from the C2 server to the formerly opened file descriptor that corresponds to the updated backdoor image

0x54	0x0	Spawns a new process of the backdoor with the argument 1, which would in turn attempt to execute an image with that name. The purpose of this action is unclear.
0x55	0x0	Same as command 0x54
0x56	0x0	Serves as an echo command; receives a command packet from the server and replies back with a packet that has the same major command code and blank body. Possibly used to check the infected host's connectivity\state.

Table 2: Supported commands

The Linux iteration of BOLDMOVE leverages several statically compiled libraries to implement its functionality:

- An undetermined and likely custom library used for event handling (reminiscent of libevent). It operates in a single-threaded mode, wherein each action is scheduled and executed as an event callback. It may allude to the fact that the developers aimed for supporting the infection of single core devices, among others.
- WolfSSL (also compiled in a single-threaded mode), which facilitates SSL encrypted communication to the C2 server.
- Musl libc

Upon failure, the malware reruns itself in a new process. In addition, if the malware is executed with a command line argument, it would not initiate the backdoor logic but rather attempt to execute the provided argument as a new process.

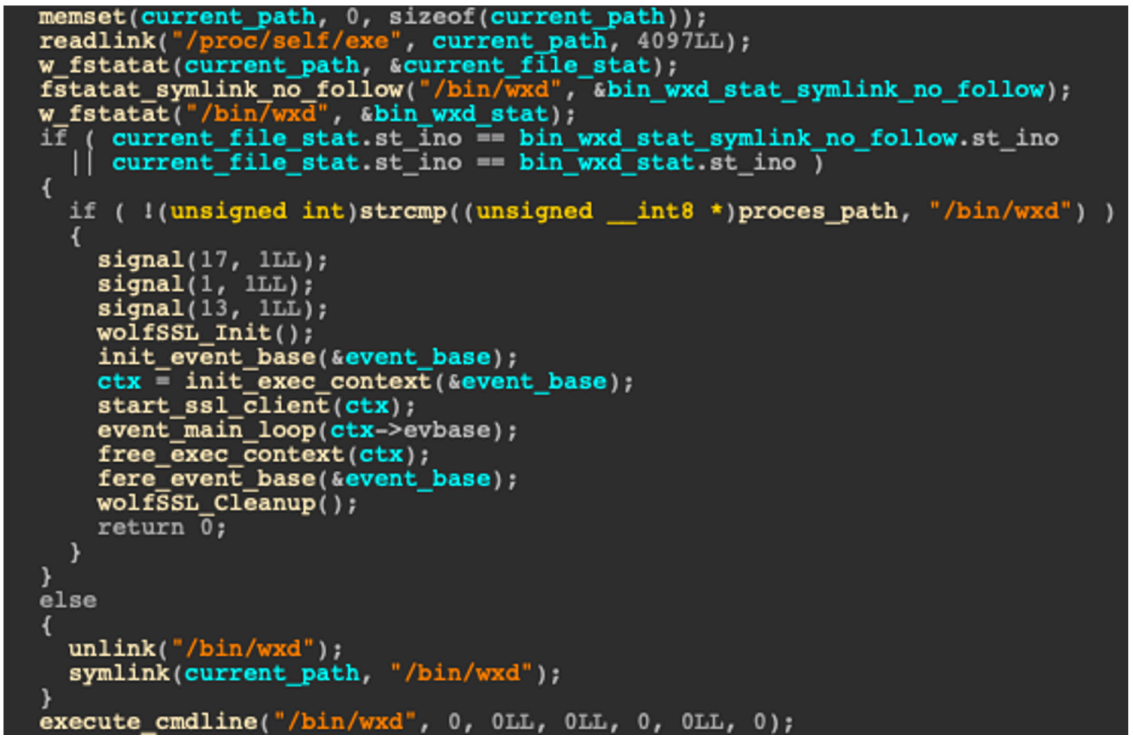
Prior to starting the backdoor's logic, the malware calls the signal function in order to ignore the signals SIGCHLD, SIGHUP, SIGPIPE.

Extended Features

The extended version of BOLDMOVE (MD5: 3191cb2e06e9a30792309813793f78b6) contains all the aforementioned functionality but with additional features.

The extended version contains Execution Guardrails (T1480) by verifying that it is executing from a specific path. It accomplishes this in the following manner:

1. Retrieving its own path from `/proc/self/exe`
2. Obtaining an inode from this resultant path via fstatat
3. Obtain a secondary inode from the statically defined path `/bin/wxd`
4. Comparing these two inode records



```
memset(current_path, 0, sizeof(current_path));
readlink("/proc/self/exe", current_path, 4097LL);
w_fstatat(current_path, &current_file_stat);
fstatat_symlink_no_follow("/bin/wxd", &bin_wxd_stat_symlink_no_follow);
w_fstatat("/bin/wxd", &bin_wxd_stat);
if ( current_file_stat.st_ino == bin_wxd_stat_symlink_no_follow.st_ino
  || current_file_stat.st_ino == bin_wxd_stat.st_ino )
{
  if ( !(unsigned int)strcmp((unsigned __int8 *)proces_path, "/bin/wxd") )
  {
    signal(17, 1LL);
    signal(1, 1LL);
    signal(13, 1LL);
    wolfSSL_Init();
    init_event_base(&event_base);
    ctx = init_exec_context(&event_base);
    start_ssl_client(ctx);
    event_main_loop(ctx->evbase);
    free_exec_context(ctx);
    fere_event_base(&event_base);
    wolfSSL_Cleanup();
    return 0;
  }
}
else
{
  unlink("/bin/wxd");
  symlink(current_path, "/bin/wxd");
}
execute_cmdline("/bin/wxd", 0, 0LL, 0LL, 0, 0LL, 0);
```

Figure 1: Path Execution Guardrails

The extended version contains a command that can perform Indicator Blocking (T1562.006) by disabling Fortinet daemons `miglogd` and `syslogd`. It also contains a command enabling it to patch memory address spaces of the same logging daemons. Due to Mandiant being unable to obtain those executables from Fortinet devices, we are unable to accurately determine the nature of those patches. However, Mandiant assesses it is likely that they are intended to disable a logging capability during the backdoor's run-time. Each patch data is kept in the following struct:

```
 struct st_log_patch_struct
{
    char fortigate_version_name[24];
    __int64 target_addr1;
    __int64 patch_bytes1;
    __int64 target_addr2;
    __int64 patch_bytes2;
}log_patch_struct;
```

Table 4 in Appendix A summarizes the targeted FortiGate Devices, their corresponding patched addresses, and bytes.

Additionally, the extended version of BOLDMOVE contains a command capable of modifying proprietary Fortinet logs on the system. It checks the following paths:

- `/tmp/log`
- `/var/log/log`
- `/var/log`

For filenames matching the format:

- `elog`
- `offset/elog.ofs`
- `offset/elog..cidx`

One of BOLDMOVE's extended variant commands is capable of decompressing, parsing, and overwriting the undocumented structure pertaining to those proprietary log files allowing the attacker to modify chosen parts of the logs.

The extended version contains a Watchdog like feature that may enable the malware to persist across upgrades. To accomplish this, BOLDMOVE monitors two files via the fstatat function:

```
 /data/lib/libgif.so
/data/lib/libips.so
```

If the size of these files differs, BOLDMOVE performs the following actions:

- Creates a backup of the legitimate file `/data/lib/libips.so` stored at `/data/lib/libiptcp.so`
- Overwrites the legitimate library `/data/lib/libips.so` with a trojanized version of it located at `/data/lib/libgif.so`

Thus, if there were to be a system patch that replaced /data/lib/libips.so and the malware was still executing, it would be able to undo the patch and maintain execution.

In addition, the extended version contains a command that allows the attackers to send requests to an internal Fortinet service, possibly to modify device settings or expose internal parts of the associated network to the internet. BOLDMOVE reads the contents of `/dev/cmdb/vdom` and parses its information to retrieve a numeric value, which may be associated with a virtual domain on the device. Then it creates a connection to "127.0.0.1", localhost, over an attacker provided port. This suggests that a server is expected to run on that port locally. The command handler facilitates sending attacker-chosen data over the established connection and sending back any retrieved response back to the C2.

Table 3 outlines some of the differences between the Windows and Linux variants of BOLDMOVE that were identified by Mandiant:

	Windows	**Linux**
Compiler	C and compiled with MinGW (GCC: (GNU) 10.2.1 20210227) Compile Time: 2021-08-26 07:13:04	C and compiled with GCC 11.2.1 20211120 Compile Time: Unknown
SSL/TLS	No	Yes

UserAgent	**curl/6.12.34** (this is a non-public version of libcurl, last v6 build was 6.5; also, the malware itself does not make actual use of libcurl)	**curl/6.12.34**
C2	Private class C IP Address	Globally routable IP Address
Supports light weight systems	**No** • Uses an event driven model wherein event callbacks are used instead of threads. This is facilitated by a library like the one leveraged by the Linux variant of BOLDMOVE, however the reason for using it in Windows is unclear.	**Yes** • Uses an event driven model, wherein event callbacks are used instead of threads • Musl is compiled statically into the malware's binary image.Musl has been associated for its lighter utilization of resources in compraison to other libc variants. • WolfSSL that is used by the malware for encrypting traffic to the C2 is also designed in part with embedded devices in mind.
Encryption	Established connection packets are encrypted with Salsa20: Key: <8_byte_pseudorandom_nonce> || "**e**8d**m_$Gb**"	Established sessions are encrypted with AES128: Key: <8_byte_pseudorandom_nonce> || "rg8P@TD(" IV: <8_byte_pseudorandom_nonce> || "**e**5s**m_$Gb**"
Campaign	0.1c#2021-08-26 15:13:01	Charlotte/c(other campaign names were observed in different samples of the Linux variant)

Table 3: Differences between the Windows and Linux variants of BOLDMOVE

The survey and commands are functionally equivalent amongst both Linux and Windows.

Windows and Linux Variant Comparison

Table 3 shows the distinction between the Windows and Linux variants of BOLDMOVE. Most importantly, the Windows variant appears to have been compiled a year before the Linux variants. This discrepancy in time could indicate that the attackers have been developing BOLDMOVE and possibly using it in the wild since that time. The differences may offer insight into the functionality and intended use of the malware.

- There are a few differences in choices of libraries that were statically compiled into each of the variants. While the WolfSSL library was used in Linux in order to encrypt traffic, the Windows variant does not make use of it. In addition, the Linux version leverages a statically compiled Musl libc library as opposed to standard libc functions imported as a result of compiling the Windows variant with MinGW. The usage of the Musl libc in the Linux variant along with a library that facilitates an event driven communication with the C2 server, could indicate that the Linux version is generally intended to be used on embedded devices, and devices with low processing power.
- Mandiant assesses that the BOLDMOVE Linux variant was deployed on Fortinet devices after a successful exploitation of CVE-2022-42475. However, the method for initial infection from the Windows variant is currently unclear. With that in mind, a private class C IP address (192.168.120[.]206) that was used in the Windows variant could indicate that it was used to communicate with an infected device inside the network following lateral movement or was merely used for testing.

Acknowledgment

Mandiant would like to acknowledge Fortinet’s assistance in sharing information, coordinating, and analyzing Mandiant’s findings to verify its veracity.

Appendix A: Patches

Table of patches made in memory addresses of `miglogd` and `syslogd` logging daemons on various FortiGate versions by the extended Linux version of the BOLDMOVE backdoor. Those patches are made seemingly in order to weaken logging mechanisms during the malware’s run-time.

FortiGate Version	Address 1	Bytes Written to Address 1	Address 2	Bytes Written to Address 2
FG100F v7.0.5	0x1E4BFA8	E0 03 02 AA 7F 0A 00 B9	0x25A6A50	E0 03 02 AA 1F 00 00 71
FG100F v7.0.7	0x1E88B68	E0 03 02 AA 7F 0A 00 B9	0x2604C90	E0 03 02 AA 1F 00 00 71

FG101F v6.4.10	0x1A5DD80	E0 03 02 AA 7F 0A 00 B9	0x213C154	E0 03 02 AA 1F 00 00 71
FG101F v6.4.8	0x1A2FA90	E0 03 02 AA 7F 0A 00 B9	0x20F0C00	E0 03 02 AA 1F 00 00 71
FG200D v6.0.11	0x1E4F9CC	48 89 D0 90 90 83 F8 00	0x0EC73DF	48 89 D0 90 90 49 89 C7
FG200E v6.0.12	0x1DB524D	48 89 D0 90 90 83 F8 00	0x0F03262	48 89 D0 90 90 49 89 C5
FG200E v6.4.4	0x19409FD	48 89 D0 90 90 83 F8 00	0x1FABDDA	48 89 D0 90 90 85 C0 7F
FG200E v7.0.4	0x1E65991	48 89 D0 90 90 C7 43 08	0x25D5F31	48 89 D0 90 90 85 C0 7F
FG200E v7.0.8	0x1ECAE81	48 89 D0 90 90 C7 43 08	0x2665951	48 89 D0 90 90 85 C0 7F
FG200E v7.2.0	0x1F3AFD1	48 89 D0 90 90 C7 43 08	0x26EB5C1	48 89 D0 90 90 85 C0 7F
FG201F v6.4.7	0x1AB581D	48 89 D0 90 90 83 F8 00	0x217156A	48 89 D0 90 90 85 C0 7F
FG201F v6.4.9	0x1ABF90D	48 89 D0 90 90 83 F8 00	0x218388B	48 89 D0 90 90 85 C0 7F
FG240D v6.0.12	0x1E5558C	48 89 D0 90 90 83 F8 00	0x0EC753F	48 89 D0 90 90 49 89 C7
FG3H0E v6.2.10	0x2019ABD	48 89 D0 90 90 83 F8 00	0x1FB826B	48 89 D0 90 90 85 C0 7F
FG5H0E v6.0.5	0x1CF537D	48 89 D0 90 90 83 F8 00	0x0EBD7B0	48 89 D0 90 90 49 89 C5
FG6H1E v6.4.8	0x1A1E21D	48 89 D0 90 90 83 F8 00	0x20CE65A	48 89 D0 90 90 85 C0 7F

FG6H1E v6.4.9	0x1A2862D	48 89 D0 90 90 83 F8 00	0x20DF7FB	48 89 D0 90 90 85 C0 7F
FG6H1E v7.2.1	0x20AFCE1	48 89 D0 90 90 C7 43 08	0x28BF201	48 89 D0 90 90 85 C0 7F
FG800D v6.2.10	0x20E18ED	48 89 D0 90 90 83 F8 00	0x2080AEB	48 89 D0 90 90 85 C0 7F
FG800D v6.2.11	0x20E1B2D	48 89 D0 90 90 83 F8 00	0x2080D2B	48 89 D0 90 90 85 C0 0F
FG800D v7.0.8	0x1F61271	48 89 D0 90 90 C7 43 08	0x272DCF1	48 89 D0 90 90 85 C0 7F
FGT5HD v6.4.10	0x1A317CD	48 89 D0 90 90 83 F8 00	0x210250B	48 89 D0 90 90 85 C0 7F
FGT60F v6.4.10	0x1953248	E0 03 02 AA 7F 0A 00 B9	0x1FFD6A4	E0 03 02 AA 1F 00 00 71
FGT60F v6.4.4	0x1904898	E0 03 02 AA 7F 0A 00 B9	0x1F7BF88	E0 03 02 AA 1F 00 00 71
FGT60F v6.4.8	0x192D018	E0 03 02 AA 7F 0A 00 B9	0x1FB7450	E0 03 02 AA 1F 00 00 71
FGT60F v6.4.9	0x193B0B0	E0 03 02 AA 7F 0A 00 B9	0x1FFC304	E0 03 02 AA 1F 00 00 71
FGT80F v6.4.10	0x19F6360	E0 03 02 AA 7F 0A 00 B9	0x20ADA54	E0 03 02 AA 1F 00 00 71
VM64 v6.2.3	0x1A64193	48 89 D0 90 90 83 F8 00	0x0F2F646	48 89 D0 90 90 85 C0 48

Table 4: Patches made in memory addresses of miglogd and syslogd logging daemons on various FortiGate versions

Appendix B: IOCs

- **Basic BOLDMOVE**
  - MD5: 12e28c14bb7f7b9513a02e5857592ad7
  - SHA256: 3da407c1a30d810aaff9a04dfc1ef5861062ebdf0e6d0f6823ca682ca08c37da
- **Extended BOLDMOVE**
  - MD5: 3191cb2e06e9a30792309813793f78b6
  - SHA256: 0184e3d3dd8f4778d192d07e2caf44211141a570d45bb47a87894c68ebebeabb
- **Windows version of BOLDMOVE**
  - MD5: 54bbea35b095ddfe9740df97b693627b
  - SHA256: 61aae0e18c41ec4f610676680d26f6c6e1d4d5aa4e5092e40915fe806b679cd4

Posted in

- Threat Intelligence
