A Defender's Perspective of SSL VPN Exploitation

paraflare.com/a-defenders-perspective-of-ssl-vpn-exploitation/

Daniel Eden

Published by Daniel Eden | 15 June 2021

Daniel Eden
Director DFIR

June 15, 2021
10 min read.

The exploitation of Fortinet FortiOS vulnerabilities has been very topical of late. Over the past 12 months, ParaFlare's Incident Response team has assisted with the response, containment and remediation of several large-scale ransomware cases that involved unpatched Fortinet FortiGate firewalls.

As such, we have analysed several CVEs that have impacted Fortinet devices worldwide and have led to devastating attacks on networks from the perimeter device inwards. It is our assumption that the exploitation of CVE-2018-13379, an 'Improper Limitation of a Pathname to a Restricted Directory (Path Traversal), was to blame.

The vulnerability assigned the reference CVE-2018-13379 allows an unauthenticated, remote actor to read sensitive files on the Fortinet device, including the usernames and plaintext passwords contained in the "sslvpn_websession" file. Therefore, enabling the threat actor to access the SSL VPN with legitimate user credentials. We believe this vulnerability was most likely used by attackers to gain initial access in all recent cases investigated by ParaFlare, where an unpatched Fortinet device was involved.

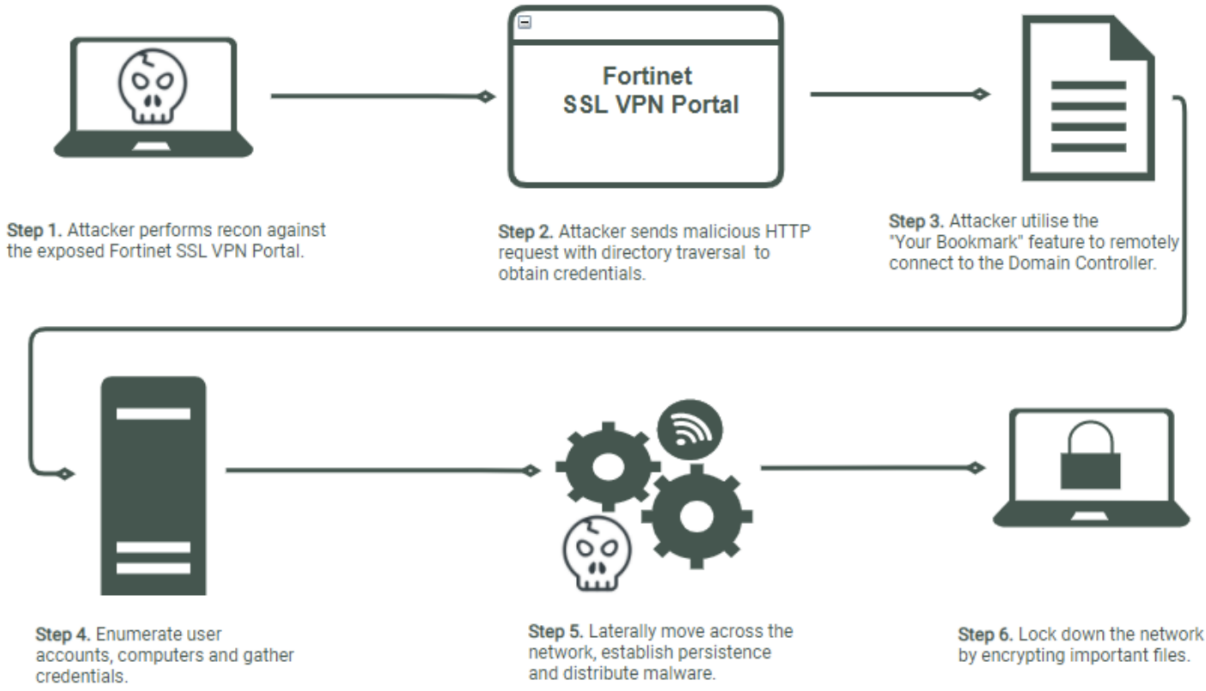
During these response engagements, while ParaFlare were able to assess the likelihood of the attacker utilising the exploits as disclosed by this blog post, the evidence has not been available to support these assumptions. This drove us to delve a little deeper into these vulnerabilities and determine whether there was any evidence that could be extracted from the devices to support the hypothesis.

Confirming or disproving assumptions regarding the root cause of an incident is important, as it allows you to make informed decisions during the response process and more effectively prioritise security requirements. Timing is critical when you are responding to security incidents, as they can substantially impact business operations. For this reason, there must be a balance between the desire to achieve certainty in your root cause analysis, minimising security risks, and enabling an organisation to recover as quickly as possible.

> There are numerous vulnerabilities exploited by attackers to breach network defences, laterally move within an environment, impersonate legitimate users, and ultimately, achieve their objective. Most often the objective has been financial gain through the extortion of victims who have had their critical data stolen and/or encrypted.

The most severe and maybe the most widely exploited Fortinet CVE, in our opinion, is CVE-2018-13379. Why? Because it is so trivial to perform and from what ParaFlare have understood from our engagements and research, it is very difficult, if not almost impossible, to confirm this as the source of compromise. Let us go through this step by step, see what this means for the network owner, attacker, and the response team.

The Attack



Step 1: External Reconnaissance

Firstly, the attackers would launch a mass scan, looking for Fortinet SSL Web VPN portals exposed to the Internet. A simple approach is to script out the IANA netblock listings, take your pick at which country you feel like targeting, and search for TCP/10443 (default Fortinet SSL Web VPN). Shodan also makes this a trivial exercise. The login screen for the Fortinet SSL VPN web portal has been demonstrated below.

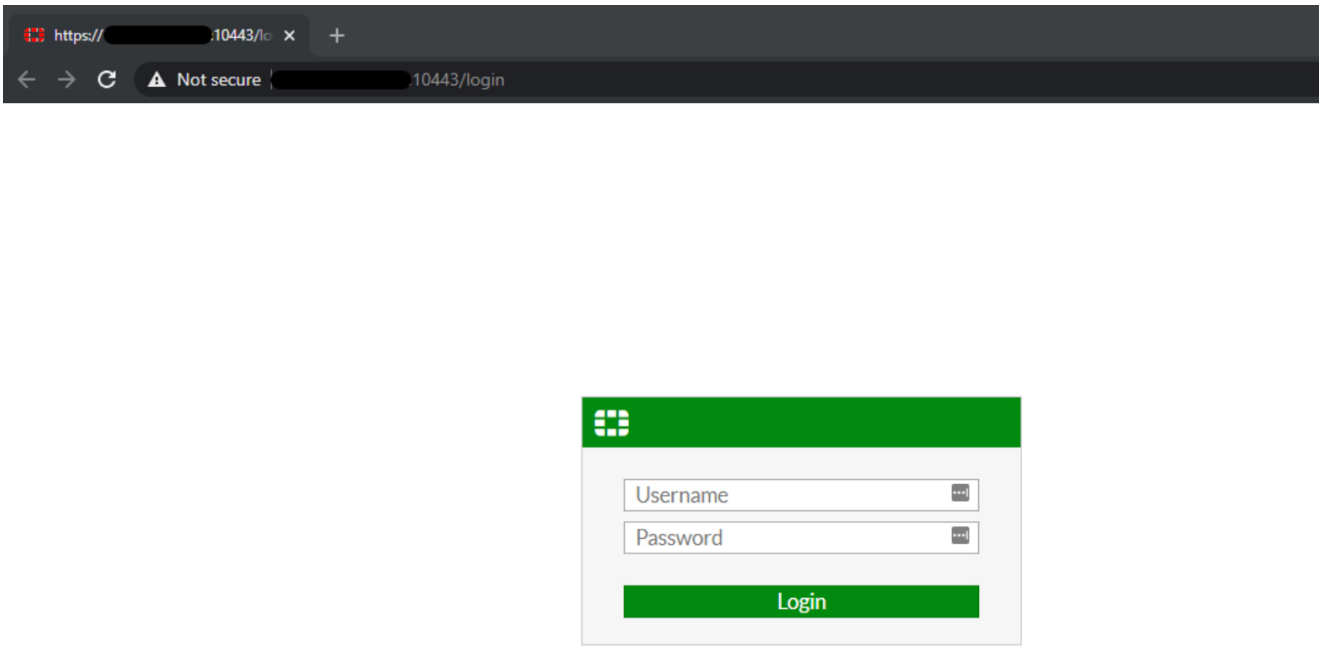


Figure 1: Example FortiGate Web VPN SSL portal

Step 2: Crafting the Malicious Request

The CVE write-up tells us that “in Fortinet FortiOS 6.0.0 to 6.0.4, 5.6.3 to 5.6.7 and 5.4.6 to 5.4.12 under SSL VPN web portal allows an unauthenticated attacker to download system files via special crafted HTTP resource requests”.

The aim of the game is to grab a specific “KEYFILE” from disk. This KEYFILE is a symlink2 to a file called “sslvpn_websession” that lives in /dev/cmdb. What is this file and why does it exist? According to the limited internet research ParaFlare conducted on the purpose of this file, it is not clear why the file is present in the first place, and as the fix to this vulnerability is to remove the file, we then assume it serves no critical purposes. However, what the file contains is extremely sensitive information, you guessed it… cleartext Web SSL VPN authenticated username and passwords.

To exploit this vulnerability, it is as easy as appending the FortiGate URL with the path shown in Figure 2 with a simple GET request. A path traversal attack (also known as directory traversal) aims to access files and directories that are stored outside the web root folder. By manipulating variables that reference files with “dot-dot-slash (../)” sequences and its variations, or by using absolute file paths, it may be possible to access arbitrary files and directories[3].

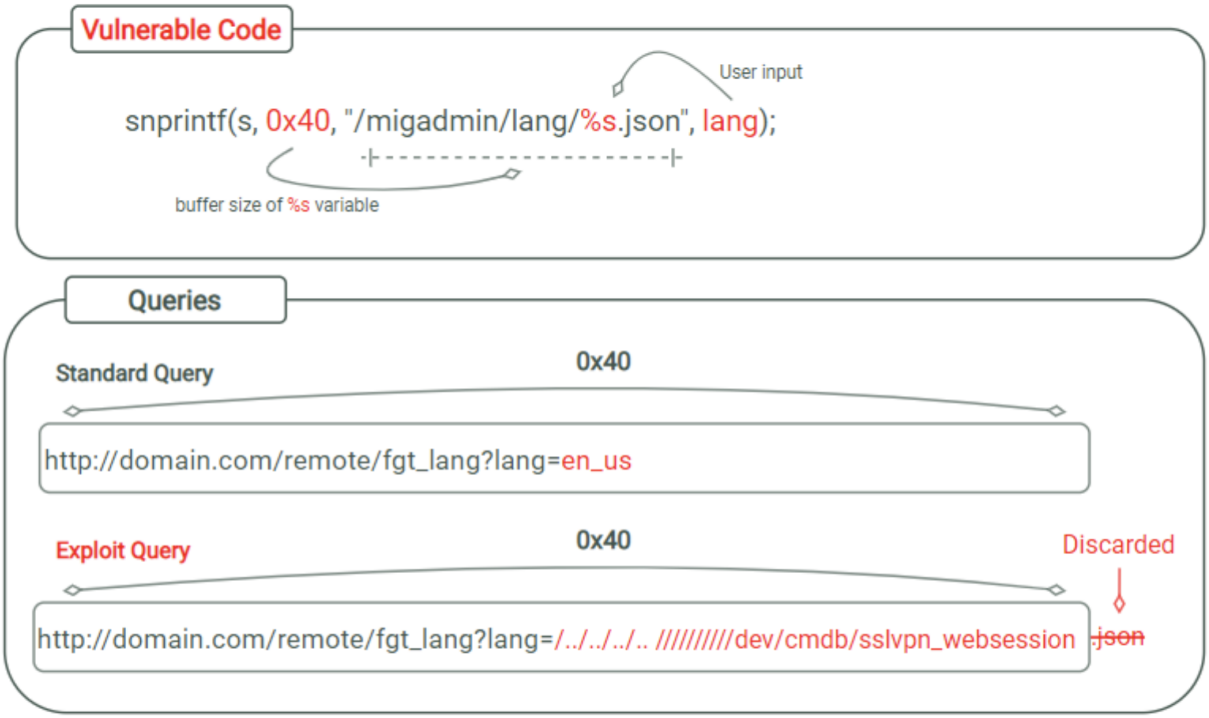


Figure 2: Exploiting the snprintf function

The attack essentially leverages a flaw in the snprintf function where an attacker can pass their own variables into the GET request. This allows for the valid URI to be overridden and file contents to be shown. The extra "/"'s in the path denote the buffer size of the overwrite up to the file attribute:

For instance, to traverse to "/dev/cmdb" you will need an extra eight slashes ('/'):

```
dev = 3, / = 1, cmdb = 4 (total of 8 bytes)
```

If you want to grab any file, simply calculate the path offset length and append the required slashes.

Step 3: Execution

Metasploit is an open-source framework that can be used to test and exploit vulnerabilities. Within Metasploit there is a module that can be used to exploit CVE-2018-13379 extremely easily.

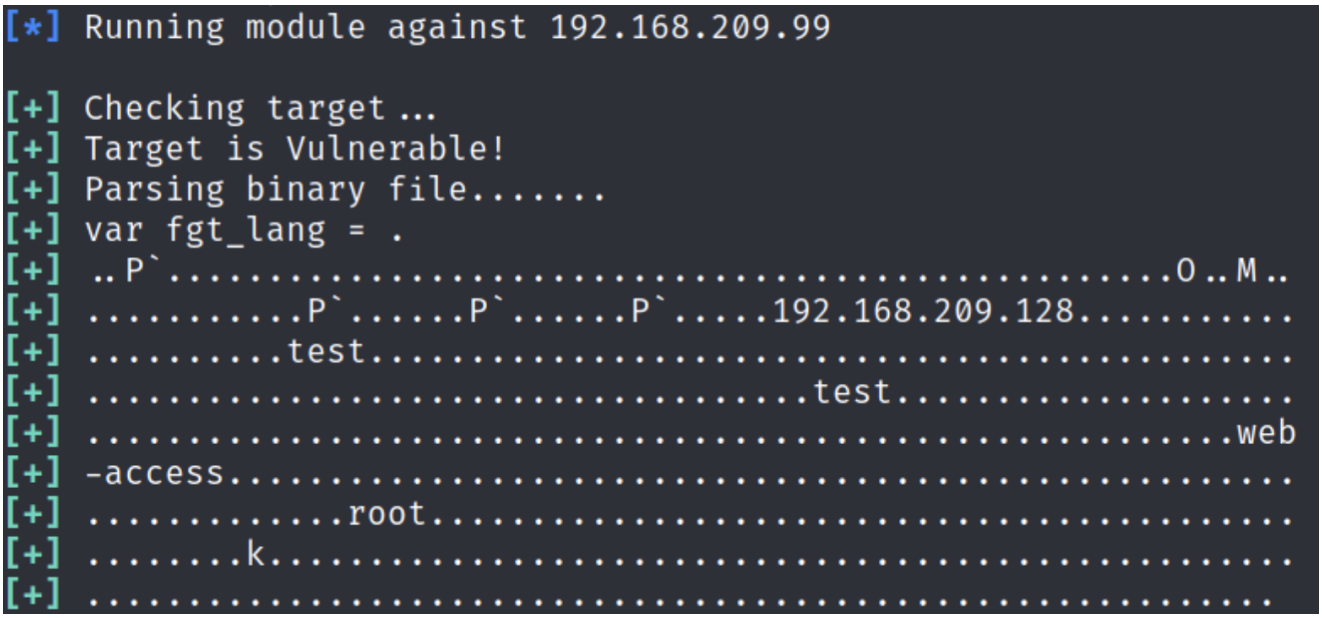


Figure 3: Running Metasploit against CVE-2018-13379

Let me break down what has been captured in Figure 3:

- The target victim is 192.168.209.128 (a Fortinet 6.0.1 virtual appliance)
- This Metasploit module confirms that the target is vulnerable to CVE-2018-13379
- The credentials being used is user:test and password:test, which have been successfully retrieved using this exploit module.

Step 4: Success

Now using these credentials, assuming that multifactor authentication is not enabled (which invariably, it is not), you can use those credentials to login into the SSL Web VPN portal. This is where things get interesting. Do you see anything here that looks enticing?

Figure 4: Successfully logged in

Now that you have successfully authenticated to the portal, you can decide to download FortiClient for continual ease of access, but also to blend in with other users. What should be noted here, is that if Active Directory SSO is used to access the VPN server, then essentially Active Directory credentials are exposed, and which inevitably may include domain administrators.

Additionally, the “Your Bookmark” feature is a feature-rich ‘in browser’ RDP wrapper based on the Apache Guacamole protocol. Effectively, all you need to do to make your life super simple (and the attacker’s) is save your favourite server credentials (e.g., Domain

Admin) inside a bookmark. Then use the Fortinet Web SSL VPN via your browser to RDP directly into the machine of your choice. This is used quite widely amongst organisations to provide third party access to systems, such as contractors or vendors.

Step 5: The Wrap Up

As an attacker, you have made it from the internet to the Domain Controller in a matter of four clicks; scan -> exploit -> login -> move laterally.

The Response

Now that the attack was proven successful, how do we as responders work out what has happened and if CVE-2018-13379 was indeed the point of compromise for the attack? This is where things get difficult.

As FortiGate's are a security device, you can log events via syslog, FortiAnalyzer or FortiCloud, the latter of which sends the logs to a specified storage location. ParaFlare have assisted multiple clients in the analysis of the Fortinet logs, and unfortunately there are approximately zero logs in relation to the "migadmin" web application (Web SSL VPN portal). What we wanted to know was whether we were missing a configuration or potentially identify logs hidden somewhere on the device to confirm our theory.

Testing via a mock syslog server (python socket UDP/514 listener) whilst firing off the exploit resulted in no findings, even with all "Logging" configuration options enabled in the Fortinet Administrator console. So clearly, it was not a configuration issue that we could easily change to obtain the evidence we needed.

As some folks may remember, the Citrix NetScaler vulnerability CVE-2019-19781 was exploited on-masse around 2019-2020. Performing forensics on virtual appliances brings up memories of ephemeral RAM disks and insanely locked down shells in the form of wrapped command line interfaces. Given this, some creative thinking was needed in order to try and find these logs, if they even existed.

Here are some other methods that were attempted to get some form of "patient zero" evidence in logs:

- Made sure all logging facilities were enabled in the Fortinet console
- Captured memory of the Fortinet virtual appliance and keyword searched for "GET /remote", "sslvpn_websession". Some remnants were discovered in memory within Apache web logs, however this did not identify a disk-based file or location. Further to this, the remnants were sporadic and did not actually detail the exploit. My suspicion is that the embedded Apache web server is writing access logs to the migadmin web application in memory only.
- Image and ramdisk tarball analysis for signs of logging also produced no results.

- Explored the “fnsysctl” super admin command on the Fortinet console, attempting to read certain log files in /var/log. This looked promising until I quickly figured out that fnsysctl and a very limited console shell barely gives you any room to move. I attempted to profile the system layout, looking for directories or files of interest, however this fell short.
- Explored the migadmin web application for any directories that may have resembled log file locations.
- Explored the idea of using the path traversal CVE to my advantage by arbitrarily grabbing files from disk, specifically in /var/log. Whilst this works, there is no actual log files for me to grab that is of interest (apart from what is being sent to syslog already)

Prevention of Successful CVE-2018-13379 Exploitation

To protect against this type of vulnerability, the theory of using the inbuilt FortiGate IPS/SSL-decryption modules and custom signatures to block the delivery of the crafted payload body was investigated. This is a cumbersome exercise and requires heavy customization of interfaces, VIPs, and signatures. Ultimately it would be much faster to patch the appliance. However, if you want to take this option, here are the steps to achieve it:

- Traffic flow remains on the single exposed FortiGate device, but with some crafty NAT’ing and VIP’s, you can effectively re-route the interface via your new IPS policy.
- To caveat this, turning on IPS and SSL Inspection will *probably* incur CPU penalties, so please keep this in mind before implementing these measures in production.

A more detailed breakdown of this process has been demonstrated below.

1. For SSL-VPN settings, configure connection settings to listen on one of the internal interfaces.
2. Configure Static-NAT using Virtual IP to map the SSL-VPN gateway (IP/FQDN) to the IP address of internal interface selected in step 1.
3. Create IPv4 Policy using this Virtual IP to allow traffic from external interface (configured with SSL-VPN gateway IP) to the new internal interface listening on port TCP/10443.
4. Create custom IPS signature to match the desired pattern. Directory traversal pattern for instance.
   e.g. F-SBID(--name “SSLVPN.IPS.Event”; --protocol tcp; --service HTTP; --flow from_server; --pcre “..\x3b”;)
5. Create new IPS Sensor using the IPS signature created in step 4 and configure the action as either Block or Monitor and enable logging.
6. Add IPS sensor to new policy created in step 3.
7. Any security events triggering this new IPS sensor will be logged and display the action taken as either dropped or detected.

Given proper infrastructure architecture and design, placing the SSL Web VPN behind a Web Application Firewall would be a critical measure to ensure protection against web-based attacks. The steps that were just provided are limited in their ability to truly defend against web-based attacks; IPS is not designed to be that stop gap.

Enabling multifactor authentication is also an important security control. Although it cannot prevent the exploit from happening, it would almost always prevent the attacker from utilising the credentials that had been exposed.

Concluding Remarks

To wrap up, as defenders, what can we do to identify if a network was compromised with this CVE? We can only assume that CVE-2018-13379 was the culprit. Defenders can perform behavioural analysis on the existing logs, such as inbound and outbound firewall analysis, geolocation analysis and login time analysis, to name a few. This investigation will definitely highlight abuse, misuse, or attacker movements, but it will not definitively present evidence to answer the age-old question of “how did they get in?”. This exploitation provides a classic demonstration of it being easier for the attacker to gain access to a network rather than the defender being able to protect it from such an attack.

[1] https://docs.fortinet.com/document/fortigate/6.2.0/cookbook/579694/ssl-vpn-web-mode-for-remote-user
[2] A symlink is a type of file that links to another file or a folder
[3] https://owasp.org/www-community/attacks/Path_Traversal

Have a comment? Join the conversation on LinkedIn

- Blog
- Technical research