National Cyber Security Centre
a part of GCHQ

Malware Tipper

Line Runner

Persistent webshell targeting Cisco Adaptive Security Appliance (ASA) devices.

Version 1

24 April 2024

Line Runner

Persistent webshell targeting Cisco Adaptive Security Appliance (ASA) devices.

Executive summary

- Line Runner is a Lua webshell which is persisted using novel abuse of Cisco Adaptive Security Appliance (ASA) WebVPN customisation functionality, assigned CVE-2024-20359.
- Line Runner has the capability to run arbitrary Lua code sent via tasking.
- Line Runner implements multiple defence evasion techniques to avoid detection and prevent recovery.
- Line Runner is distinct from Line Dancer, an in-memory shellcode loader covered in another NCSC Malware Tipper.
- Line Dancer and Line Runner have been observed being deployed and in use together to achieve actor objectives on ASA devices.
- Checking for the presence of Line Runner as an administrator via traditional means is difficult. However, a solution is provided in this report to identify a compromised device.
- This activity is tracked by Cisco Talos as the ArcaneDoor campaign.
- It is advised to check for the presence of Line Dancer prior to checking for the presence of Line Runner, as a device reboot will remove traces of Line Dancer.
- NCSC would like to acknowledge the support from the Canadian Centre for Cybersecurity (CCCS) and the Australian Signal Directorate's Australian Cyber Security Centre (ACSC) on the malware investigation.

NCSC would like to thank Cisco for enabling the analysis in this report. If you discover the presence of Line Runner on a Cisco ASA device, please contact Cisco via the following link and report it to the respective cyber security centre or agency in your jurisdiction.

Malware details

Metadata

Filename	`client_bundle_install.zip`
Description	`ZIP file which installs the Line Runner webshell that includes anti-forensic measures when run.`
Size	`4459 bytes`

Introduction

This report covers analysis of Line Runner, a Lua webshell which is installed persistently by abusing Cisco ASA WebVPN plug-in customisation functionality which has been assigned CVE-2024-20359.

Line Runner was observed in use by the ArcaneDoor campaign discussed by Cisco Talos in this blog post.

The recovered file this report is based on, `client_bundle_install.zip`, was recovered through forensic analysis of a Cisco ASA device.

Once the ZIP file has been placed into the directory `disk0:/` a reboot of the Cisco ASA device will install Line Runner.

It is currently unknown how the Line Runner malware is placed onto the device, be that through credential use, direct access, exploiting a vulnerability, or another mechanism.

Line Runner is believed to be an additional means of access onto the ASA devices which survives reboots, and a vehicle for accessing data staged into the Lua file system for exfiltration. Despite Line Runner being persistent, it should be noted that it is non-trivial to detect.

Functionality

Overview

Line Runner is a persistent Lua webshell which abuses an auto-install feature for WebVPN plug-ins to launch itself. The webshell is very succinct in functionality, simply interpreting and executing data it is sent via HTTP(S) GET parameters as a local Lua script.

Line Runner implements multiple defence evasion techniques, some bespoke to Cisco ASA functionality. For example, when Line Runner is installed, no files or directories discussed in this report are visible to an administrator.

The majority of malicious functionality implemented in Line Runner stems from the file `csco_config.lua` which is in the root of `client_bundle_install.zip`.

As part of its infection process, Line Runner adds malicious code to multiple system files for defence evasion purposes. The malicious code to be added to system files is found in text files packaged in the ZIP.

Figure 1 below shows a tree listing of the malicious zip file.

```
client_bundle_install.zip\csco_config.lua
client_bundle_install.zip\csco_config2.lua
client_bundle_install.zip\client_bundle_install\test\hash.txt
client_bundle_install.zip\client_bundle_install\test\stgvdr.txt
client_bundle_install.zip\client_bundle_install\test\umtfc.txt
client_bundle_install.zip\client_bundle_install\test\laecsnw.tx
t
client_bundle_install.zip\client_bundle_install\test\index.txt
client_bundle_install.zip\client_bundle_install\plugin\rdp.jar
```

Figure 1: Tree listing of `client_bundle_install.zip`

Execution

Cisco ASA devices contain functionality which allows for a WebVPN plug-in to be automatically installed on device reboot. WebVPN plug-ins are Java applets which can be installed manually via the command-line or the Adaptive Security Device Manager (ASDM) too.

On device boot if a ZIP file, named `client_bundle_install.zip` is present in `disk0:/` then it is unzipped and the `.jar` file in the `/plugin` directory is installed as a WebVPN plug-in, by executing the dynamically generated

command shown in Figure 2. The plug-in's name is also used to set the command-line options. The file bundled in our malicious ZIP is named `rdp.jar`, so the plug-in protocol is set to `rdp`. The use of RDP in this instance is not significant.

```
import webvpn plug-in rdp disk0:/client_bundle_install/client_bundle_install/plugin/rdp.jar
```

Figure 2: Generated CLI command to install the plug-in

Analyst Comment: The ZIP file does not have to be explicitly be called client_bundle_install.zip but the file has to match the Lua pattern `^client_bundle[%w_-]%.zip$`.

The above command being run will first execute a Lua callback function which runs the Lua script `csco_config.lua` which is in the root of the unzipped `client_bundle_install.zip`.

This Lua script contains all the malicious functionality of Line Runner and is responsible for carrying out the rest of the activity covered in this report.

Under normal operation, after the Cisco code to install the WebVPN plug-in has completed, the system deletes the ZIP file. However, it is resident in Lua's in-memory filesystem as `/client_bundle.zip`.

Persistence

To maintain persistence on the device, and circumvent the system deletion of the ZIP described at the end of the 'Functionality (Execution)' section, Line Runner copies the malicious ZIP file to `/run/lock/subsys/krbkdc6`. To write to this location it uses a directory traversal technique shown below.

```
ifs.dump(cbizipcontent, "disk0:/csco_config/../../../../../run/lock/subsys/krbkdc6")
```

Figure 3: Directory traversal example

On system shutdown this file is then renamed back to `client_bundle_install.zip` and placed in the `disk0:/` root directory, so it is then loaded again on device boot. This is described in more detail in the 'Defence Evasion (/etc/init.d/umountfs)' section.

Webshell

Line Runner implements a Lua webshell that is tasked via HTTP(S) GET requests. Line Runner requires two HTTP parameters to be present for tasking, one which acts as a token and the other contains the payload to be run. Due to the way the webshell has been implemented in an underlying WebVPN dependency, it is reachable by sending GET requests to multiple unauthenticated WebVPN endpoints.

The ZIP file contains a text file, `client_bundle_install/test/hash.txt` which contains the code for the legitimate Cisco ASA WebVPN file `browser_inc.lua` with some added Line Runner code which implements a webshell allowing the actor to run arbitrary Lua code. The Lua script, `browser_inc.lua` appears to be an include file as part of WebVPN server-side code. Multiple server-side files include (and thus, execute) `browser_inc.lua`, when being served. The actor has been observed directing tasking to `portal.css`. An example of this can be found in the Canadian Centre for Cybersecurity (CCCS) published advisory here.

Analyst Comment: The legitimate functionality of `browser_include.lua` appears to be used to populate a `browser_info` structure with the browser, browser version, and operating system type by parsing the User-Agent field of incoming requests.

To interact with the webshell there must be two HTTP GET parameters present, the specific strings of which are known to vary between victims. The first parameter acts as a unique token preventing bulk Cisco ASA probing by defensive entities.

The first HTTP GET parameter is a 32-character alphanumeric string and the value must match a 32-character, hardcoded alphanumeric authentication token in the webshell. The tokens are victim specific and are distinct from a Line Dancer token used for a similar purpose.

The second parameter is also a 32-character alphanumeric string, and the value of this parameter is written to the file `/cdmgcDE9FFA6FF5A3CB9B51F5BC024A1F57CF` and it is executed as a Lua script. The file is deleted after it has been run.

To ensure the actor's copy of `browser_inc.lua` is executed and not the original one, the actor writes the modified version of the file `browser_inc.lua` (`hash.txt`) into the following directory `disk0:/csco_config/97/webcontent/` with the filename `1515480F4B538B669648B17C02337098`. This filename is the MD5 hash of `+CSCOE+/include/browser_inc.lua`.

Analyst Comment: The directory `disk0:/csco_config/` is not visible via a `show` or `dir` Cisco command due to having the hidden attribute set on the filesystem. It is unclear if this overwrites the original copy of the file `browser_inc.lua`.

The following Lua table data in Figure 4 is appended to the file `disk0:/csco_config/97/webcontent/index.ini` which informs WebVPN that the file created above is the file `browser_inc.lua` which has a WebVPN filepath of the following URI `/+CSCOE+/include/browser_inc.lua`.

```
webContent{
  ['hash'] = "1515480F4B538B669648B17C02337098",
  ['path'] = "../+CSCOE+/include/browser_inc.lua",
  ['fn'] = "browser_inc.lua",
  ['dir'] = "../+CSCOE+/include",
  ['ext'] = "lua",
}
```

Figure 4: Lua table appended to `index.ini`.

Defence Evasion

WebVPN Plug-in install prevention

Line Runner gains execution via the auto-install of plug-ins functionality implemented in Cisco ASA devices. However, Line Runner prevents the full installation of the `rdp.jar` plugin by temporarily patching out the Lua function on the Cisco ASA which imports plugins. Therefore, execution of Line Runner does not result in the installation of a new WebVPN plug-in appearing in the installed plug-ins list.

Line Runner achieves this by temporarily overwriting the function pointer `csco_config.importPlugin`, which is responsible for carrying out the installation of the provided plugin. Line Runner restores the function pointer of `csco_config.importPlugin` to its original location after it has gained execution so that any subsequent legitimate attempts to install plug-ins succeed.

The result of this is that an unexpected plug-in would not appear to administrators if they were to check, but also Line Runner does not require a legitimate Java applet plugin (the `rdp.jar` file it masquerades to the system as a plugin is a file containing simply "`111`").

WebVPN customisation import/export hooking

Line Runner replaces the file `csco_config.lua` after its initial run with another which deletes the webshell if an attempt is made to import or export WebVPN customisations.

After it has run once, Line Runner deletes the malicious file `csco_config.lua` and replaces it with another from the ZIP, initially called `csco_config2.lua`, but renamed to be `csco_config.lua`. The impact of this is that prior to the import or export of WebVPN customisation or a plug-in, `csco_config.lua` is called, which in this instance will delete the webshell from the system before any customisation is exported or imported.

Webshell Removal

Line Runner prepends code to several system files (via the directory traversal technique) which removes the webshell during certain events.

/etc/init.d/umountfs

This Bash script is run by the Cisco ASA prior to system reboot and is used to unmount local file systems. Line Runner prepends some code to this file using the directory traversal technique which will remove the webshell and restore the original `index.ini` file which was modified previously – as discussed in the 'Functionality (Webshell)' section.

This script is also responsible for restoring the Line Runner installer ZIP (`client_bundle_install.zip`) back into the root of `disk0`, causing Line Runner to install again on system boot. The code for this line is obfuscated with base64 encoding.

Analyst Comment: Due to this being the only place where Line Runner restores itself, if this script does not run then Line Runner is not persistent. ***A hard reboot by pulling the power plug from the Cisco ASA has been confirmed to prevent Line Runner from re-installing itself.***

/asa/scripts/lina_exe_cs.sh

It is believed that the modifications to files in this section remove the presence of the webshell files and any traces to modifications of system files on device boot. Sleeping for 30 seconds is likely to allow for some component to be fully loaded during boot before proceeding.

This script, by its own description is used to "`launch the lina_cs process outside vASA`". Not much is known about this script and what executes it, but it is believed that it is executed on system start, after the Client Bundle Install functionality has installed Line Runner.

Line Runner creates a file, `/asa/scripts/lina_cs` and populates it with Bash code found in the following file from the ZIP directory `client_bundle_install/test/stgvdr.txt`.

The Bash script, when executed will:

- Sleep for 30 seconds.
- Change the file ownership and file metadata timestamps for the file `/etc/init.d/umountfs` (modified, described in the previous section) to match that of the file `/etc/init.d/rcS`.
- Delete the file `/mnt/disk0/csco_config/97/webcontent/1515480F4B538B669648B17C02337098` (the webshell).
- Restore the backed up `index.ini` file.
- Delete the file `/asa/scripts/lina_cs` (itself) from the system.

Line Runner prepends some Bash code which it takes from the file (`laecsnw.txt`), found in the ZIP, to the file `/asa/scripts/lina_exe_cs.sh`.

This code will:

- Execute the previously described script, `/asa/scripts/lina_cs`.
- Restore the currently running script, `/asa/scripts/lina_exe_cs.sh`, to a previously backed up clean copy.
- Change the file ownership and file metadata timestamps to match that of the file `/asa/scripts/rcS.common`.

Detection

Multiple detection opportunities have been identified for Line Runner, both of which require at least a device reboot.

Remote forensics

With the understanding discussed in this report about Line Runner and its method of execution, it is known that the Cisco ASA on device boot will iterate `disk0:/` looking for file matching the Lua pattern `^client_bundle[%w_-]%.zip$`.

If an administrator creates a file which matches the expected Lua pattern in the `disk0:/` directory before device reboot, it will take precedence over the maliciously restored copy of the zip discussed in the 'Functionality (Persistence)' section.

The result of the above is that on device reboot the Cisco ASA will read, error on, and delete the administrator created file and the maliciously restored actor zip file will be present if a `show disk0:/` command is carried out post-reboot.

This will simultaneously disinfect the device, confirm whether it is infected and provide you with the malicious zip present on the Cisco ASA device.

```
ciscoasa> enable
ciscoasa# show version | redirect disk0:/client_bundle.zip
ciscoasa# show disk0:
ciscoasa# reload

# ... device reboots

ciscoasa> enable
ciscoasa# show disk0:

# If device was compromised
# ... client_bundle.zip would be missing
# ... client_bundle_install.zip [Line Runner] would be present
```

Figure 5: Commands to confirm compromise

The extra ZIP file can then be copied off the Cisco ASA device via either USB or TFTP and confirmed to be malicious via analysis of the contents e.g. by checking if the structure is the same as shown in Figure 1.

Disk forensics

NCSC have also confirmed that traditional disk forensics is a viable option to confirm compromise. However, this will require physical access to the device and specialist forensic equipment.

Warning: physical modification of the device is not recommended by Cisco and renders the device damaged and un-supportable in the field.

The Cisco ASA device tested by NCSC was a Cisco ASA 5525-X which has an 8GB eUSB card plugged into the motherboard. When powered off, the device can be disassembled to expose the motherboard, and the eUSB chip can be unplugged (no desoldering required) and forensically imaged using standard forensic software.

Analysis of the forensic image was able to confirm the presence of deleted files such as the webshell and malicious zip discussed in this report.

It would also be possible to perform a clean reboot of the device, interrupting it with an ESC key press before boot. The eUSB card could then be removed and read in the same fashion described above. However, the malicious ZIP file would be present directly via an eUSB reader, bypassing the requirement for a forensic image of the device.

Conclusion

Line Runner represents a significant threat to Cisco ASA devices. The actors have understood the opportunities for execution, persistence, and defence evasion on the Cisco ASA device very well.

Understanding the opportunities for persistence and execution will have required significant and advanced understanding of Cisco ASA devices, as there would appear to be no documentation surrounding Client Bundle Install functionality.

The simple measures that have been put in place to prevent detection are incredibly effective on host due to the environmental constraints in which administrators operate in on Cisco ASA devices. Lack of access to the underlying Bash shell, limited access to the local filesystem and no access to the in-memory filesystem combine to make detection incredibly difficult.

Detecting Line Runner therefore required an equally inventive solution discussed above, which also involved understanding how the Cisco ASA software and the malware operated.

Not placing the webshell in a file directly accessible via the web, combined with seemingly victim specific HTTP parameters (i.e. an authentication token) also make it challenging to scan for.

Disclaimer

This report draws on information derived from NCSC and industry sources. Any NCSC findings and recommendations made have not been provided with the intention of avoiding all risks and following the recommendations will not remove all such risk. Ownership of information risks remains with the relevant system owner at all times.

This information is exempt under the Freedom of Information Act 2000 (FOIA) and may be exempt under other UK information legislation.

Refer any FOIA queries to ncscinfoleg@ncsc.gov.uk.