Vermilion Strike: Linux and Windows Re-implementation of Cobalt Strike

intezer.com/blog/malware-analysis/vermilionstrike-reimplementation-cobaltstrike/

September 13, 2021

Written by Avigayil Mechtinger, Ryan Robinson and Joakim Kennedy - 13 September 2021

Key Findings

- Discovered Linux & Windows re-implementation of Cobalt Strike Beacon written from scratch
- Linux malware is fully undetected by vendors
- Has IoC and technical overlaps with previously discovered Windows DLL files
- Highly targeted with victims including telecommunications, government and finance

Cobalt Strike is a popular red team tool for Windows which is also heavily used by threat actors. At the time of this writing, there is no official Cobalt Strike version for Linux.

In August 2021, we at Intezer discovered a fully undetected ELF implementation of Cobalt Strike's beacon, which we named **Vermilion Strike.** The stealthy sample uses Cobalt Strike's Command and Control (C2) protocol when communicating to the C2 server and has Remote Access capabilities such as uploading files, running shell commands and writing to files. The malware is fully undetected in VirusTotal at the time of this writing and was uploaded from Malaysia.

Based on telemetry with collaboration from our partners at McAfee Enterprise ATR, this Linux threat has been active in the wild since August targeting **telecom companies**, **government agencies**, **IT companies**, **financial institutions** and **advisory companies** around the world. Targeting has been limited in scope, suggesting that this malware is used in specific attacks rather than mass spreading.

After further analysis, we found Windows samples that use the same C2. The samples are re-implementations of Cobalt Strike Beacon. The Windows and ELF samples share the same functionalities.

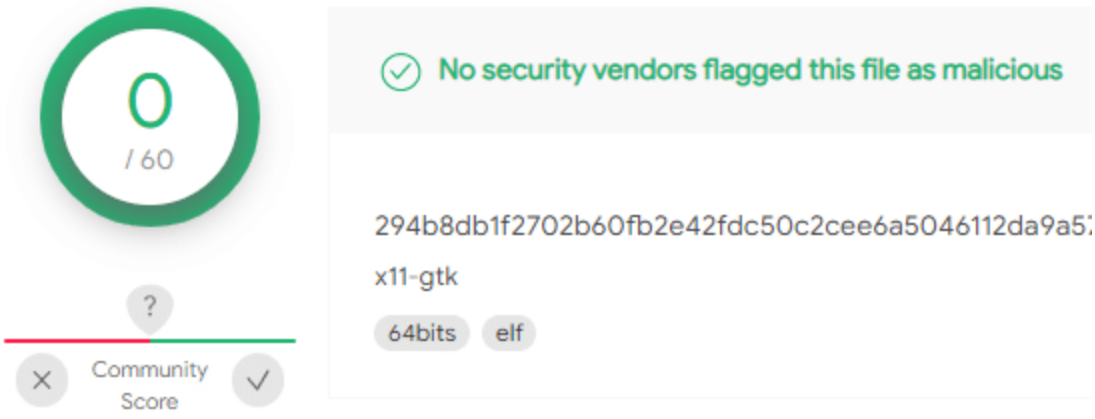
The sophistication of this threat, its intent to conduct espionage, and the fact that the code hasn't been seen before in other attacks, together with the fact that it targets specific entities in the wild, leads us to believe that this threat was developed by a skilled threat actor.

In this post we will provide a technical analysis of the samples and explain how you can detect and respond to this threat.

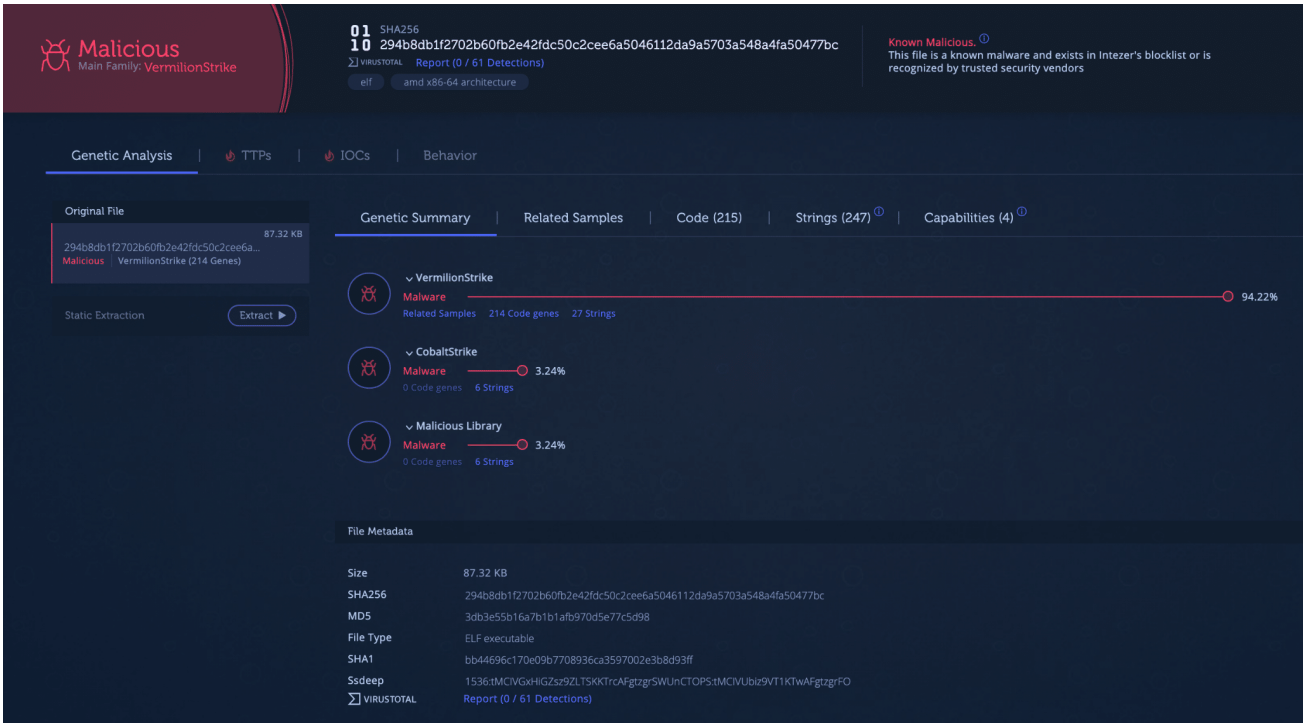
Technical Analysis

Linux File

The file was uploaded to VirusTotal from Malaysia and has no detections in VirusTotal at the time of this writing.



294b8db1f2702b60fb2e42fdc50c2cee6a5046112da9a5703a548a4fa50477bc in VirusTotal



Vermilion Strike analysis in Intezer Analyze.

The file shares strings with previously seen Cobalt Strike samples and triggers a number of YARA rules that detect encoded Cobalt Strike configurations. The ELF file is built on a Red Hat Linux distribution. It uses OpenSSL via dynamic linking. The shared object names for OpenSSL on Red Hat-based distributions are different from other Linux distributions. Because of this, it can only run on machines with Linux distribution based on Red Hat's code base.

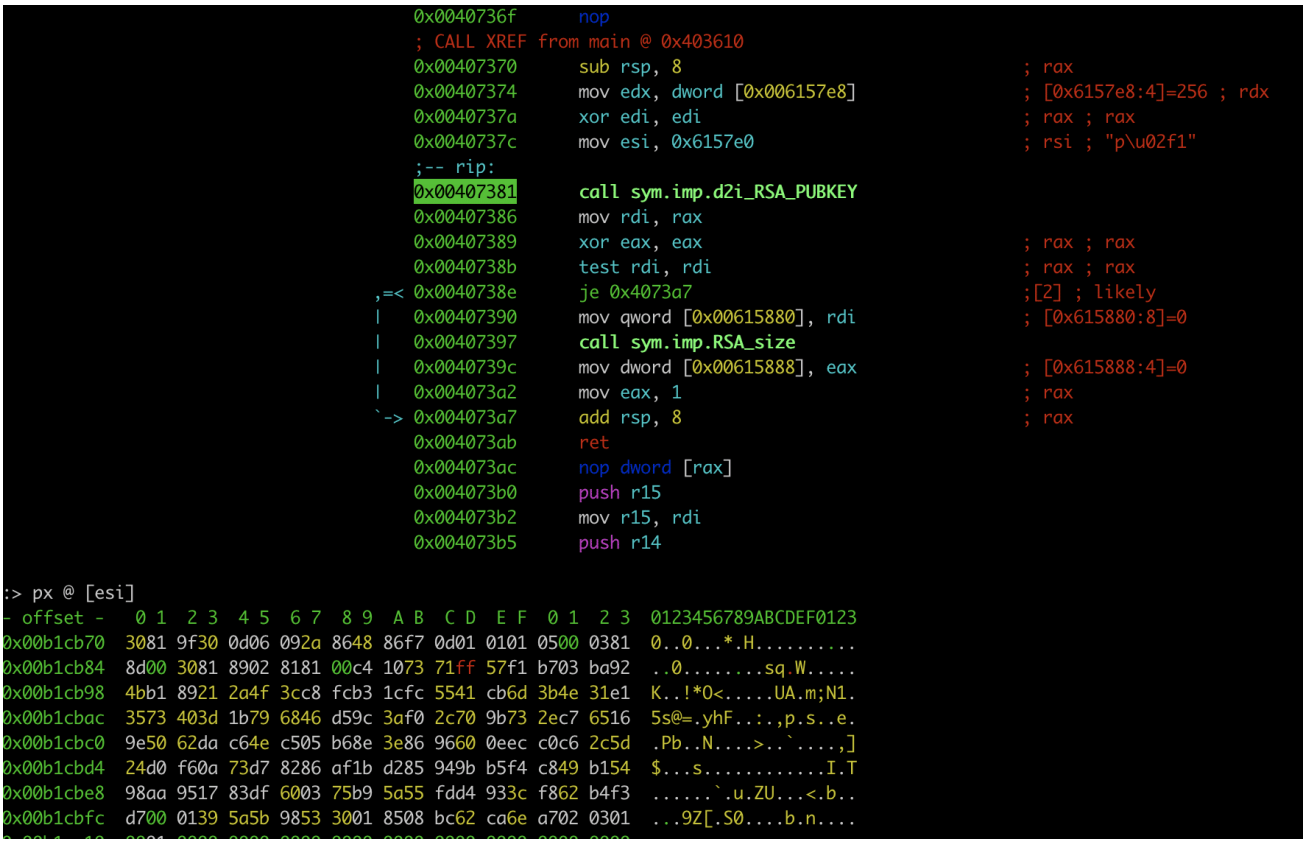
Initialization

The sample starts by forcing itself to run in the background using daemon. It will decrypt the configuration, using the XOR key **0x69**, shown in the screenshot below. The key **0x69** is a common value used by Cobalt Strike's encrypted configuration too. Vermilion Strike's configuration format is the same as Cobalt Strike. Tools used for extracting Cobalt Strike configurations can also be used to extract Vermilion Strike configuration. The Windows components of the configuration are ignored for this Linux version.

```
 A 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789A  comment
30 ......................'.............................0
e1 ..0...*.H............0........sq.W.....K..!*O<.....UA.m;N1.
b1 5s@=.yhF..:.,p.s..e..Pb..N....>..`....,]$...s............I.
00 T......`.u.ZU...<.b.....9Z[.S0....b.n.....................
00 ..........................................................
69 ........................update.microsoftkernel.com,/dot.gi
00 f,update.microsofthk.com,/ca..............................
00 ..........................................................
00 ..........................................................
2f ..................................................Mozilla/
42 4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; GTB
00 7.4; InfoPath.2)..........................................
00 .......@/template/template.jsp.............................
00 ..........................................................
00 ..........................................................
00 ..........................................................
00 ..........................................................
00 ..........................................................
00 .....Cookie........param1=format........param2=output.....
00 ..!Referer: http://www.microsoft.com.......................
00 ..........................................................
00 ..........................................................
00 ...................&Content-Type: application/octet-stream.
00 ......!Referer: http://www.microsoft.com................id.
00 ..........................................................
00 ..........................................................
6e ....................................@%windir%\syswow64\run
73 dll32.exe.....................................@%windir%\s
00 ysnative\rundll32.exe.....................................
00 .\\%s\pipe\msagent_%x......................................
00 ..........................................................
00 ..........................................GET.............
00 ........POST.......................#.....................
28 ........$........(((((((((((((((((((((((((((((((((((((((
28 ((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

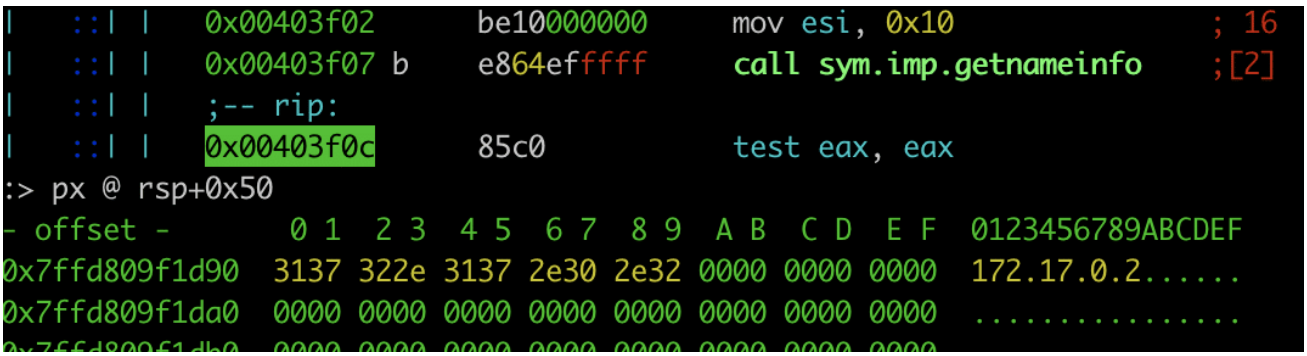
Decoded configuration of the beacon.

Further decryption is performed in a heap with decoded strings, keys, and values required by the beacon for its operation. The beacon will then generate a SHA256 hash sourced from a random number seeded from the thread ID. This value will be used later in DNS beaconing. Next, a public RSA key will be imported for later use.



Importing of public RSA key to encrypt machine fingerprint.

The beacon will begin fingerprinting the machine. A random number will be generated and the process ID will be fetched. It will grab the kernel version of the machine using **uname**. Next, the beacon will fingerprint network information through the **getifaddrs** function. It will loop through the interfaces looking for IPv4 addresses. It will gather the interface with an address not equal to “127.0.0.1” and stage the IPv4 address.



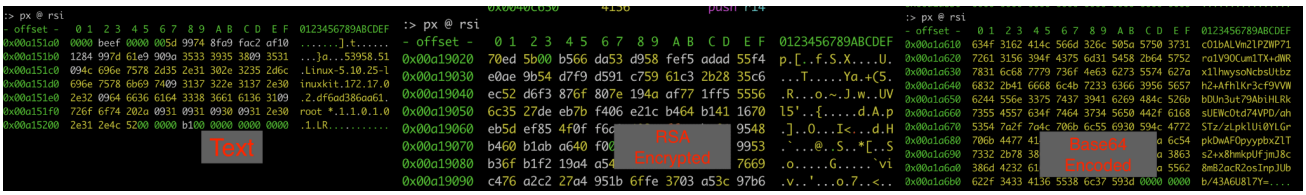
Network interface fingerprinting.

Next, the beacon will fingerprint the entry in the local password database for information about the current effective user ID of the process.

```
|           0x00404097      4883ec40        sub rsp, 0x40
|           0x0040409b      e8a0edffff      call sym.imp.geteuid        ;[4] ; uid_t geteuid(void)
|           0x004040a0      89c7            mov edi, eax
|           0x004040a2      89c5            mov ebp, eax
|           0x004040a4 b    e877effffff     call sym.imp.getpwuid       ;[5]
|           ;-- rip:
:> px @ [rax]
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x0166f950  726f 6f74 0078 0030 3a30 3a72 6f6f 7400  root.x.0:0:root.
0x0166f960  2f72 6f6f 7400 2f62 696e 2f62 6173 6800  /root./bin/bash.
```

Fingerprinting of local password database.

The beacon will then fingerprint the hostname of the machine. The collected information will be formatted into a string, encrypted with the public RSA key, and base64 encoded, as is standard for communication with a Cobalt Strike server. The stages are shown below.



Stages of formatting the machine fingerprint.

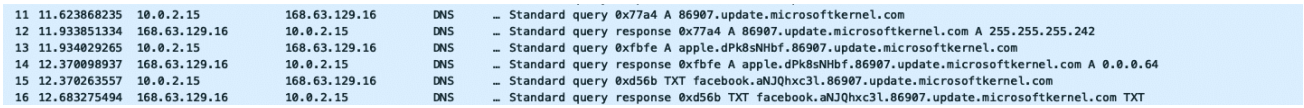
Prepended to the fingerprint string is the value “1.0.1.LR”. This appears to be an internal version string. A similar string, “W1.0.1,” was found in a newly discovered Windows sample of Vermilion Strike that shares the same C2 and malware functionality.

The encrypted data is sent to the C2 server in a similar way that the metadata is sent from a Cobalt Strike beacon to the C2 server. The payload that is encrypted starts with the marker **0xbeef**. The same marker is used by the legitimate Cobalt Strike beacon.

Command and Control

Command and Control is primarily performed over DNS but also available over HTTP. This DNS-based approach for communications can help avoid traditional defenses that monitor HTTP traffic. Commands are received via DNS Address (A) and Text (TXT) records. The beacon first makes DNS requests out to hardcoded subdomains and gets an IP address returned. Normally, DNS requests on hostnames are intended to be translated into an IP address for which to visit. In this case, the IP address returned is not used as an IP address but for triggers to change the beacon behavior.

Once the beacon gets the signal to download a task, it will perform a DNS TXT query to the domain's nameservers, as shown below.



Packet capture of C2 communication.

The result of the TXT query is a base64 encoded and AES encrypted struct containing task information. An example of a returned task is shown below.

```
:> ps @ [rdi]
wQ7mYmyLh2gijCitUJCe5hI2hjrY5HmdSwc0d0ARn+TQKYLPuMioT59HiMW//f8aaRop4UADOtum/j2pBhxqGg==
:>
```

A DNS TXT query result for a task.

A decrypted task is shown below.

```
:> px @ rsi
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x00635f80  6123 8bfb 0000 0010 0000 0004 0000 0008  a#..............
0x00635f90  006d dd00 0000 0000 4141 4141 4141 4141  .m......AAAAAAAA
0x00635fa0  1010 1010 1010 1010 1010 1010 1010 1010  ................
```

Decrypted command.

Tasks that the beacon can perform are:

- Change working directory
- Get current working directory
- Append/write to file
- Upload file to C2
- Execute command via popen
- Get disk partitions
- List files

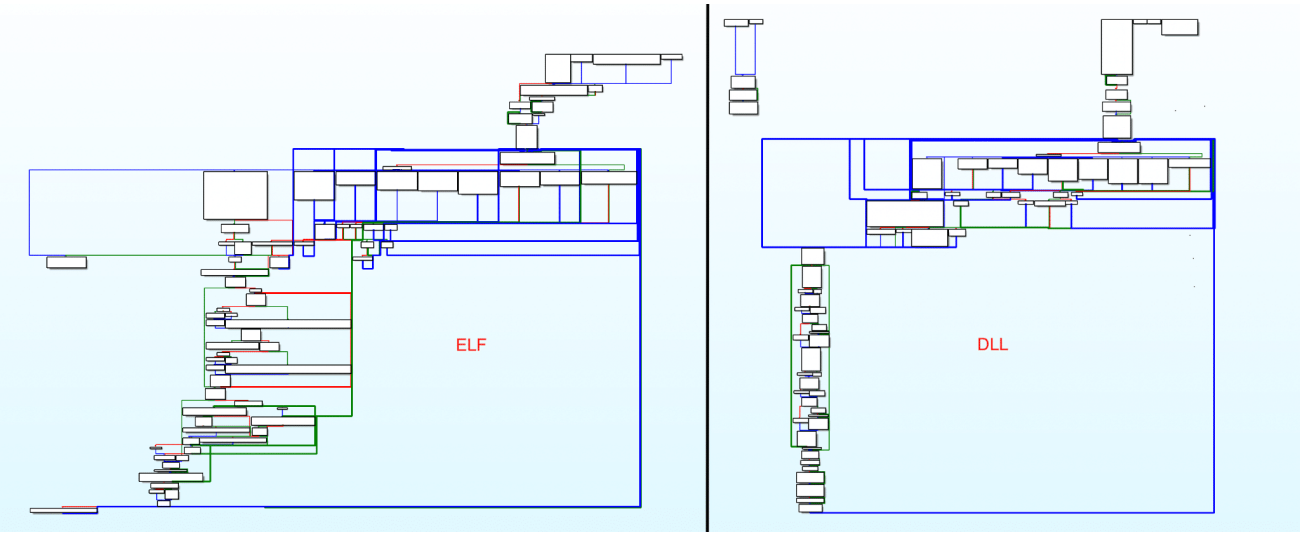
The malware uses a separate thread to execute the tasks. The tasks are scheduled as jobs via a semaphore to ensure not too many jobs are executed at once. Vermilion Strike has a third way of communicating with the C2 server via ICMP ping messages. The malware adds the current pid to the offset **0x4** in the header and the encrypted payload is sent as data in the ICMP packet. The data size for an ICMP packet is limited to 65,507 bytes but the malware uses a size limit of 64,000 bytes for the payload. The code for sending and

processing ICMP messages exists in the malware but the code for enabling it via the configuration is not present. This means it has the capability but can't be configured to use it. This suggests it may be a new feature that hasn't been fully developed yet.

Links to Windows Files

When investigating this Linux file, we discovered related Windows samples. The first sample we noticed was: **3ad119d4f2f1d8ce3851181120a292f41189e4417ad20a6c86b6f45f6a9fbcfc**. This is a 32-bit EXE sample that shares a C2 IP address (160.202.163[.]100). This is a stager that will fetch a DLL from the C2 over HTTP and execute it in-memory.

An example of the next stage DLL is **7129434afc1fec276525acfeee5bb08923ccd9b32269638a54c7b452f5493492**. This sample, first noticed in 2019 by Silas Cutler, is the Windows DLL equivalent of the ELF file. The functionality is almost exactly the same, except for the Windows environment. A side-by-side comparison of the configuration decoding function for the ELF and DLL beacons is shown below.



Configuration decryption function comparison.

The DLL has the same domains as the ELF for C2, as well as an additional configured domain "amazon.hksupd[.]com".

Using the stager we managed to get a new payload from the server (e40370f463b4a4feb2d515a3fb64af1573523f03917b2fd9e7a9d0a741ef89a5). It has a lot of shared code with the sample from 2019. This sample and another Windows version of Vermilion Strike (**c49631db0b2e41125ccade68a0fe7fb70939315f1c580510e40e5b30ead868f5**) includes a similar version string as the ELF version. The version string in these samples is "W1.0.1".

```
0x100025ec      8d442444        lea eax, [var_44h_18]
0x100025f0      e8ebf1ffff      call fcn.100017e0           ;[1]
0x100025f5      6a06            push 6                      ; 6
0x100025f7      68b4f50210      push 0x1002f5b4             ; hit0_2 ; "W1.0.1"
0x100025fc      8d442444        lea eax, [var_44h_19]
0x10002600      e8dbf1ffff      call fcn.100017e0           ;[1]
```

Internal version string in recent Windows versions.

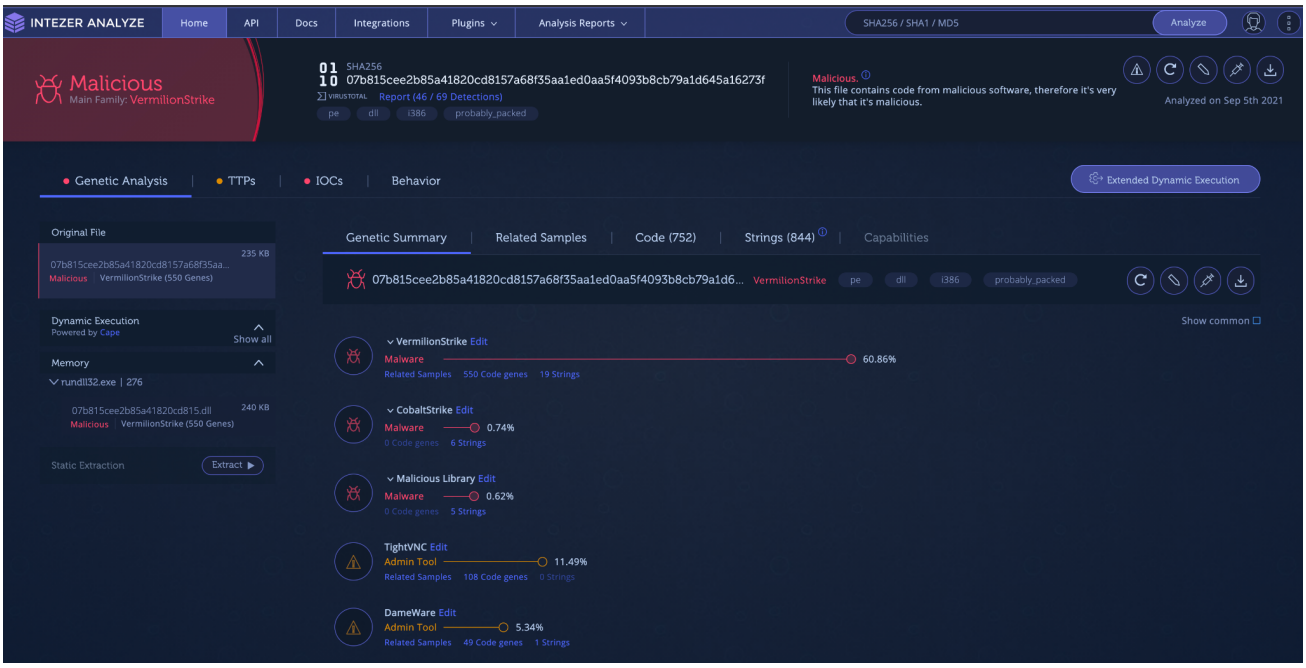
Conclusion

Vermilion Strike and other Linux threats remain a constant threat. The predominance of Linux servers in the cloud and its continued rise invites APTs to modify their toolsets in order to navigate the existing environment. Linux threats often have low detection rates compared to their Windows counterparts due to reasons discussed in Why we Should be Paying More Attention to Linux Threats.

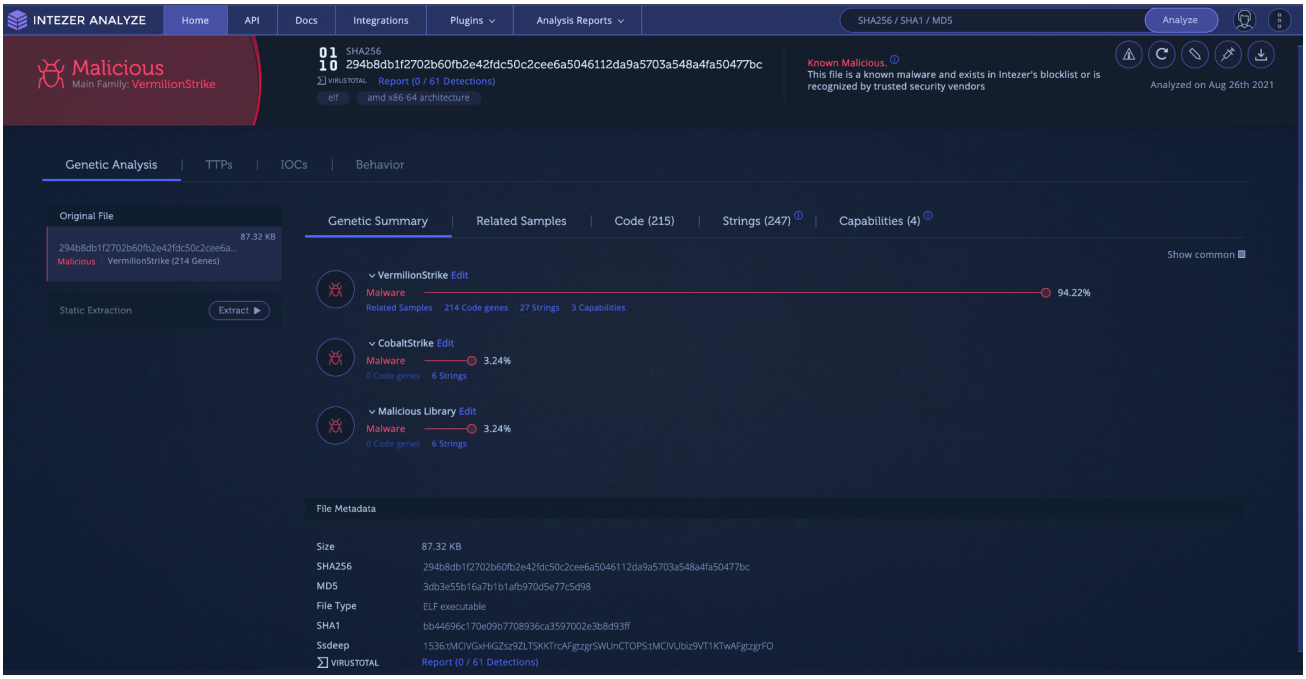
Vermilion Strike is not the only Linux port of Cobalt Strike's Beacon. Another example is the open-source project geacon, a Go-based implementation. Vermilion Strike may not be the last Linux implementation of Beacon.

Detection and Response

Intezer Analyze can detect both Linux and Windows variants of Vermilion Strike, based on code reuse, TTPs, and strings. Shown below are the verdicts for both versions.



Intezer Analyze verdict of Windows version of Vermilion Strike.



Intezer Analyze verdict of Linux version of Vermilion Strike.

Detect if a Machine in Your Network Has Been Compromised

Get full runtime visibility over your code

For Linux-based systems, use Intezer Protect to get alerted on any malicious or unauthorized code executed in runtime. Protect 10 hosts, nodes or machines for free

For Windows-based systems, use the Intezer Analyze Endpoint Scanner to scan the entire memory of your machines to find any traces of malicious code running on them.

We also recommend using the IoCs section below to ensure that the Vermilion Strike process does not exist anywhere on your system.

Response

If you are a victim of this operation, take the following steps:

1. Kill the process and delete all files related to the malware.
2. Make sure that your machine is clean and running only trusted code using a runtime security platform like Intezer Protect, or use Intezer Analyze Endpoint Scanner for Windows systems.
3. Make sure that your software is up-to-date with the latest versions and security patches and configured to security best practices.

IoCs

ELF

294b8db1f2702b60fb2e42fdc50c2cee6a5046112da9a5703a548a4fa50477bc

PE

Stager

3ad119d4f2f1d8ce3851181120a292f41189e4417ad20a6c86b6f45f6a9fbcfc

Beacon

7129434afc1fec276525acfeee5bb08923ccd9b32269638a54c7b452f5493492

c49631db0b2e41125ccade68a0fe7fb70939315f1c580510e40e5b30ead868f5

07b815cee2b85a41820cd8157a68f35aa1ed0aa5f4093b8cb79a1d645a16273f

e40370f463b4a4feb2d515a3fb64af1573523f03917b2fd9e7a9d0a741ef89a5

C2

160.202.163.100

update.microsofthk[.]com

update.microsoftkernel[.]com

amazon.hksupd[.]com

Intezer would like to thank McAfee ATR for their help during the research process.

Avigayil Mechtinger
Avigayil is a product manager at Intezer, leading Intezer Analyze product lifecycle. Prior to this role, Avigayil was part of Intezer's research team and specialized in malware analysis and threat hunting. During her time at Intezer, she has uncovered and documented different malware targeting both Linux and Windows platforms.

Ryan Robinson
Ryan is a security researcher analyzing malware and scripts. Formerly, he was a researcher on Anomali's Threat Research Team.

Joakim Kennedy
Dr. Joakim Kennedy is a Security Researcher analyzing malware and tracking threat actors on a daily basis. For the last few years, Joakim has been researching malware written in Go. To make the analysis easier he has written the Go Reverse Engineering Toolkit (github.com/goretk), an open-source toolkit for analysis of Go binaries.