Enhancing CTI Processes with Code Search Technology

Carlos Rubio & Jonas Wagner

www.threatray.com
● Search in cyber security
● Searchable binary code
● Using code search for malware identification
● Making OSINT searchable
● Key Takeaways
Search is used in many areas of cyber security...

https://*/api???17.php

Match?

https://evil.ch/apiput17.php
https://evil.ch/apiget17.php
Searching for binary code...

New malware

Match?

abc.exe 91%
contract.exe 73%
...
Searchable binary code
Binary code similarity

Broad representation of a file, includes (meta-)data, strings and code.

Brittle to compiler differences, code mutations, new variants.
Binary code search engine

Diagram:
- Binary A
  - Function A1
  - Function A2
  - Function A3
  - Function A4
  - Indexing
  - Function X1
  - Function X2
  - Function X3
  - Function Y1
  - ...

Table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Hash</th>
<th>Similarity</th>
<th>Function Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary B</td>
<td>2C204908...</td>
<td>82%</td>
<td>290 / 350</td>
</tr>
<tr>
<td>Binary X</td>
<td>9340c8fae...</td>
<td>61%</td>
<td>215 / 350</td>
</tr>
<tr>
<td>Binary Y</td>
<td>73c59aa0...</td>
<td>55%</td>
<td>192 / 350</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Using code search for malware identification
Malware identification: Determine malware family for a piece of unknown malware.

Unknown binary is 80% similar to Cobalt Strike
## Malware Identification

<table>
<thead>
<tr>
<th>File Hash</th>
<th>Threats</th>
<th>Environment</th>
<th>Analysis ID</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>06422a403ee38c1d299f9f609f2d071655a4f21e6b8bc5d7e17bbcd0eb1726855</td>
<td>Malicious</td>
<td>DLL (PE, x86-32)</td>
<td>IR-2022-17</td>
<td>VSingle</td>
</tr>
</tbody>
</table>

### Static Analysis

<table>
<thead>
<tr>
<th>Threats</th>
<th>Score (%)</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSingle</td>
<td>80%</td>
<td>56/70</td>
</tr>
</tbody>
</table>

The sample was identified as malicious with a high score of 80% for VSingle, indicating 56/70 functions were identified as malicious.
● Almost all “new” malware is a variation of previous versions.

● Variations happen because:
  ○ bypass detections (explicit)
  ○ malware evolution (implicit)

● There is of course completely new malware, but that is rare.

● What we need is a solid signature that is resilient to variations.
Code search for malware identification

70% match
Mutated MimiKatz
Mimikatz project

Source: https://github.com/gentilkiwi/mimikatz
Bypassing Windows Defender & Yara rules

- INDICATOR_TOOL_PWS_Mimikatz (DitekSHen)
- Mimikatz_Gen_Strings (Author: Florian Roth)
- Mimikatz_Strings (Author: Florian Roth)
- win_mimikatz_w0 (Author: Benjamin DELPY (gentilkiwi))
- mimikatz (Author: Benjamin DELPY (gentilkiwi))

Source: https://github.com/a0rtega/metame

Source: https://github.com/matterpreter/DefenderCheck
Resilience through code search technology

5/7 are similar or equal
~70%
Making OSINT searchable
How to relate an unknown binary file to OSINT?

- Hash
- IPs
- Sandbox
- Mutex
- URLs
- YARA signatures
- Code search signatures
- Malware family

Is it related to OSINT?
How to relate an unknown binary file to OSINT?

Unknown binary

Is it related to OSINT?

Hash

IPs

Mutex

URLs

YARA signatures

Code search signatures

Malware family

Code search

Similar to …

OSINT

Cert.pl NASK

Vidar stealer campaign targeting Baltic region and NATO entities

While working on our automatic configuration extraction, we came across a rather unusual-looking Vidar sample. The decrypted strings included domain names of such organizations as the NATO Strategic Communications Centre of Excellence.

MalwarePotato @MalwarePotato - 13h
@AgentTests #malware

“payment.jpg” 84320b06b0bbf035923c23402f9b2625

dsecond-stage DLL:

http://195.215.71.200/kytoolendants.bmp

Previously delivering #malware as reported by @jwignjmunkey

Cert.pl NASK

Vidar stealer campaign targeting Baltic region and NATO entities

While working on our automatic configuration extraction, we came across a rather unusual-looking Vidar sample. The decrypted strings included domain names of such organizations as the NATO Strategic Communications Centre of Excellence.

MalwarePotato @MalwarePotato - 13h
@AgentTests #malware

“payment.jpg” 84320b06b0bbf035923c23402f9b2625

dsecond-stage DLL:

http://195.215.71.200/kytoolendants.bmp

Previously delivering #malware as reported by @jwignjmunkey
Unknown binary OSINT search
Unknown binary OSINT search

Submit unknown binary

Show OSINT

OSINT reports, blogs, Twitter

OSINT analysis

Found 7 similar samples from 9 OSINT sources.

<table>
<thead>
<tr>
<th>URL</th>
<th>SHA-256</th>
<th>SHA-1</th>
<th>MDS</th>
<th>VERDICT</th>
<th>SIMILARITY</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://blogs.jpcert.or.jp/kn/2022/07/vsingle.html">https://blogs.jpcert.or.jp/kn/2022/07/vsingle.html</a></td>
<td>416ed95df4964477bebf...</td>
<td>ea2df903b370d8e409e...</td>
<td>6ebc777ab8d3500a574d...</td>
<td>51%</td>
<td></td>
</tr>
<tr>
<td><a href="https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-">https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-</a>...</td>
<td>416ed95df4964477bebf...</td>
<td>ea2df903b370d8e409e...</td>
<td>6ebc777ab8d3500a574d...</td>
<td>51%</td>
<td></td>
</tr>
<tr>
<td><a href="https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-">https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-</a>...</td>
<td>416ed95df4964477bebf...</td>
<td>ea2df903b370d8e409e...</td>
<td>6ebc777ab8d3500a574d...</td>
<td>51%</td>
<td></td>
</tr>
<tr>
<td><a href="https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-">https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-</a>...</td>
<td>416ed95df4964477bebf...</td>
<td>ea2df903b370d8e409e...</td>
<td>6ebc777ab8d3500a574d...</td>
<td>51%</td>
<td></td>
</tr>
<tr>
<td><a href="https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-">https://blog.siemens.com/blogs/enterprise-threat-intelligence/tonefly-north-</a>...</td>
<td>416ed95df4964477bebf...</td>
<td>ea2df903b370d8e409e...</td>
<td>6ebc777ab8d3500a574d...</td>
<td>51%</td>
<td></td>
</tr>
</tbody>
</table>
VSingle JPCERT report

Some types of malware use DGA, obfuscate destination information, or contain fake C2 server information in order to hide the original C2 server. Others obtain C2 server information from legitimate servers. Recently, the malware used by Lazesos [1] has been updated to retrieve C2 server information from GitHub. This article focuses on the updates of VSingle. VSingle has two versions, one targeting Windows OS and the other targeting Linux OS, and this article is based on the latter, which has more updates.

Communication Method

The current version of VSingle uses wget command to communicate with the C2 server while the previous versions used system call. Figure 3 shows a part of the code that executes the wget command. (VSingle on Windows OS does not include this update and uses Windows API, not wget command.)

Access Patterns to GitHub

The GitHub repository from which the communication is obtained is not fixed but dynamically generated. The following is the pattern of URLs to be accessed.

https://raw.githubusercontent.com/##########/master/README.de

The user name and repository name are the string randomly selected from the following list + a random string added.

<table>
<thead>
<tr>
<th>Username</th>
<th>Repository name</th>
</tr>
</thead>
<tbody>
<tr>
<td>garzia</td>
<td>Arcano3</td>
</tr>
<tr>
<td>woody</td>
<td>Write</td>
</tr>
<tr>
<td>t3se</td>
<td>after</td>
</tr>
<tr>
<td>lucky</td>
<td>luxuryboy</td>
</tr>
<tr>
<td>llove</td>
<td>pnpgather</td>
</tr>
<tr>
<td>v0sleij</td>
<td>happy1m</td>
</tr>
<tr>
<td>edwvsje</td>
<td>la2rplik</td>
</tr>
<tr>
<td>polaris</td>
<td>d01a</td>
</tr>
<tr>
<td>gravity</td>
<td>Dr0nek</td>
</tr>
<tr>
<td>wint3r</td>
<td>Panda3</td>
</tr>
<tr>
<td>summer</td>
<td>cpssponso</td>
</tr>
<tr>
<td>ggoddflck</td>
<td></td>
</tr>
</tbody>
</table>
Appendix A: GitHub repository used by the attacker

- https://github.com/bgrav1ty13j/bPanda3
- https://github.com/fwo0d17n/fWr0te
- https://github.com/glucky18p/gluxuryboy
- https://github.com/gfi0t18p/gpick/
- https://github.com/jv0siej21g/jia3rpik

Appendix B: C2 Server

- https://mantis.westlinks.net/api/soap/mc_enum.php
- https://www.shipshorejob.com/ckeditor/samples/samples.php
- https://ougreen.com/zone
- https://tecnojournals.com/general
- https://semiconductboard.com/xcor
- https://bluedragon.com/login
- https://tecnojournals.com/prest

Appendix C: Malware hash value

- 19bba61e6fc6af9280c5abd86c09c6f2d475c09c6c7fffc399a35c3d70277aed1
- 2eb16dcb1097a59f07787ab285a013f5fe235428cb46b948d4f9cc8e5efa5b5c
- 414ed95d149ef4aa77beb86edcede0306714c497cde14edee67b0c1425ce451d3d7
Updated Preet backdoor

The attackers used an updated version of Stonefly's custom Preet backdoor. Analysis of the backdoor revealed that it is a multistage tool:

Stage 1 is the main binary. A python script is used to unpack the binary and shellcode.

Stage 2 is shellcode. It performs the following actions:

- Sleeps for 19,999 seconds, probably in an attempt to evade sandbox detection
- Opens a mutex, with the name specified in the Stage 3 shellcode
- Instead of loading an executable file, it starts Internet Explorer (iexplore.exe) or explorer.exe and injects the Stage 3 shellcode into either. It sets up a named pipe ("\pipe\pipe") for communication. The file name of the main binary is sent over the pipe.

Stage 3 is more shellcode.

Stage 4 is the payload. It is an HTTP remote access tool (RAT) that supports various commands, including:

1. Download (Download a file and save locally)
2. Upload (Upload a file to a C&C server)
3. Set Interval (Change C&C server query interval - in minutes)
4. Shell Execute (Execute a command in the shell)
5. Download Plugin
6. Update (Download a new version and replace)
7. Info (Return debug information about the current infection)
8. Uninstall
9. Download Executable

The malware can support four different kinds of plugins: executable files, VBS, BAT, and shellcode. It supports three different persistence modes: Startup, LNK, Service, Registry, and Task Scheduler.
# Indicators of Compromise

If an IOC is malicious and the file is available to us, Symantec Endpoint products will detect and block that file.

<table>
<thead>
<tr>
<th>SHA256</th>
<th>Description</th>
<th>File name(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3b7798a4c17a3a2b58824f767ac372c543b592473da9e5d1c4d0e1e333522f94</td>
<td>Sproxy tiny proxy server</td>
<td>svh.host.exe</td>
</tr>
<tr>
<td>7ab32079e7035f00dea198ff3ed9a6b8a68005c02b1f2bbcd3b20e480754452</td>
<td>Backdoor/Prevt</td>
<td>mfs.exe, mgs_updnt.exe</td>
</tr>
<tr>
<td>537e6e222ebc8a066745e5e9d829c26c60b3a8c60c6006f5b593422e0a3dfb0d2f48a</td>
<td>Backdoor/svhost.exe</td>
<td>svhost.exe</td>
</tr>
<tr>
<td>589e707c91c8c95d34d5c62d82e3e4d6b8c83f59965e8a9e42b074a8720</td>
<td>Backdoor/Prevt</td>
<td>svhost.exe</td>
</tr>
<tr>
<td>4530144ad9a43257f1f5315d550a44a4a67f743cc0f297e86884659563aed777a771</td>
<td>Backdoor/Prevt</td>
<td>svhost.exe</td>
</tr>
<tr>
<td>dbd2d4e4b5247f99be020606c59452b08041f456267f67c7e4552e3deb007e0a9</td>
<td>Backdoor/Prevt</td>
<td>svhost.exe</td>
</tr>
<tr>
<td>41ed9d5e964946477be6f92e0e306574ca4790fe1f4ed4ed670c12452c45351d7</td>
<td>Backdoor/Prevt</td>
<td>svhost.exe</td>
</tr>
<tr>
<td>3cc6f15f5e4d563a4e5b5e3a3503c466578ac9e93f4c73586c3eaa4db05101</td>
<td>Info stealer</td>
<td>mfs.exe</td>
</tr>
<tr>
<td>863e32d58677546a61e2446437509802be3a3eb433969e80e2ff7a4af8d2f2</td>
<td>Info stealer</td>
<td>mfs.exe</td>
</tr>
<tr>
<td>551763de88b1e7b2ac2739c2ca992ea403ce2035c843384c9b649ac411726c</td>
<td>Info stealer</td>
<td>mfs.exe</td>
</tr>
<tr>
<td>2a34583b03b00299ac03f4f8a83187bce6507675549088b43e8568622</td>
<td>Info stealer</td>
<td>mfs.exe</td>
</tr>
<tr>
<td>8c3d914b089c930a1beb0d3401659884898660e3b02e6d2819d3ab9a4533c3</td>
<td>Invoke-ThreatHash</td>
<td>rpng/usr</td>
</tr>
<tr>
<td>07bf1946e2d803489c093e4e3287e28d3c2d0d50d58d4102d8853e6496ef9e</td>
<td>Mimikatz</td>
<td>pl.exe</td>
</tr>
<tr>
<td>6b6b89f51550ab32f06d852f01d5d3e571a5e06f0ce3e4e215940a401b66b086c9</td>
<td>Mimikatz</td>
<td>pl.exe</td>
</tr>
<tr>
<td>5a7128d04c00a4e80af832150b3d4779778e2c29b88b89f466357f378</td>
<td>PuTTY PSCP</td>
<td>pvhost.exe</td>
</tr>
<tr>
<td>32bf69516f86dd7b7ba321992b9f9316c9f498c0315e5a805df44e</td>
<td>Real VNC Bypass Authentication Scanner</td>
<td>vnc.exe, vncw.exe</td>
</tr>
<tr>
<td>35e33e6e3434e2e21e7b5d407e04b8e70017037c9cde0d4770f0c3195e4a</td>
<td>Runasuser privilege escalation tool</td>
<td>sempr.exe</td>
</tr>
</tbody>
</table>

| b4e3d85e0f7b8d8d55ec7f94f36c96c9d88c48b20995e8d1f25ac3f0c7b99a7737890 | Suspected proxy tool | tapi.exe |
| 0e2c87f9558a431902d242782c20771734d7e2377106e9ebd20d4d435c9f7c4f43a | Keylogger | avgkey.exe, avgekey.exe |
| 14718f7d4d4e4e2391b772242057b7b5d809409164756023533d79261b9503c960e2f | Suspected SOCKS proxy tool | svhost.exe |
| 59e6d245f919eda8b63925f5a4437c553eb5c66b6d12920a67804a347378020ccdf | Unknown file | protect.exe |
| 73996060f47ba3586d021c198966b10246645d59d83f9a9802c7f736b421c865f4 | Unknown file | wax4315.tmp |
| cc6b769d89d02e52433b7bb3a9075875e4e7e4e7e269a13172f78f8ce2d0b8756c1 | Unknown file | N/A |
| dadd5e5ee0249f6eb2eb1b7cbe9ac984a3e87bb5d36e6a0b0c8bad7b87 | Unknown file | smssv.cvc |
| b07a5ed54597570c7f4db922032d29f78e07726b0ac56e6b5c7091b8313d36d0a0a | WinRAR (old version) | na.exe |
| b7e6791797f028c271e34916969270899699d0d72b6b481842d08904dfe3e7e7 | WinSCP | winscp.com |
| de0001c7f1f868d2fd2842556f6b7c2e01e2e3927f67d06103337a314b34b43 | WinSCP | winscp.com |
| 1f460e3c3925a5d0f5e5e0196260067d56e53336e33e8e53a137e230a462c2 | wmservices.py | notepad.exe |

**tecnomuerto[.]com** | Domain | N/A |
---|---|---|
| semiconductor[.]com | Domain | N/A |
| cyancore[.]com | Domain | N/A |
| bluedragon[.]com | Domain | N/A |
| hugops[.]tecnomuerto[.]com | review | N/A |
| hugops[.]tecnomuerto[.]com | general | N/A |
| hugops[.]tecnomuerto[.]com | xmi | N/A |
| hugops[.]tecnomuerto[.]com | xcors | N/A |
| hugops[.]cyancore[.]com | Find | N/A |
| hugops[.]bluedragon[.]com | Login | N/A |
OSINT indexing workflow

01 OSINT reports, blogs, Twitter
02 Extracted file hashes
03 Malware binaries
04 Show OSINT
05 Index binaries

OSINT search engine

Extracted file hashes:
- d64a530b27af5515d72be4545784fe8b0f
- e37f03784178f0c8c89727518031024f
- c38b818e6e84eb7c088d20ba68f935a5f
- bd55b3701232f45f37e4550458824dbf17
- 3b3bf373c61e5f268e6ba2a7ae76
- 54515603c4566f6f88a81a7e048418f
- 51b9d472f771b44c5fa36d82183e8a541f
- 2b03b66c97e56d962e2a50f8731f

Show OSINT

Index binaries

OSINT reports, blogs, Twitter

Extracted file hashes

Malware binaries

OSINT search engine
Some real-world challenges

- Many malicious binaries are packed and the relevant code can only be found by statically or dynamically unpacking the binary files.

- Each binary contains library code. Need to avoid comparing library code with library code.

- Get the actual binary referenced in the sources.
The amount of variants and mutations it necessary to move towards resilient malware identification.

- Code search technology can provide the next step in this direction.

OSINT reports hold a lot of value – but it’s locked behind hashes.

- By transforming binary code into a searchable IOC, we can unlock their potential.
Thank you for your attention

carlos@threatray.com  
jonas@threatray.com