

LOKI IOC SCANNER

CYBER
THREAT
INTELLIGENCE

What is LOKI ?

LOKI is a free and simple IOC scanner, a complete rewrite of main analysis modules of our full featured APT Scanner THOR. IOC stands for 'Indicators of Compromise'. These indicators can be derived from published incident reports, forensic analyses or malware sample collections in your Lab.

LOKI offers a simple way to scan your systems for known IOCs.

It supports these different types of indicators:

- MD5 / SHA1 / SHA256 hashes
- Yara Rules (applied to file data and process memory)
- Hard Indicator Filenames based on Regular Expression (e.g. `\\pwdump\\.exe`)
- Soft Indicator Filenames based on Regular Expressions (e.g. `Windows\\[\\w]\\\\.exe`)

Rule Sets

Loki Features some of the most effective rules borrowed from the rule sets of our famous THOR APT Scanner. We decided to integrate **a lot of webshell rules** as even the best Antivirus engines fail to detect most of them. We put **almost half of our hacktool rule set** into the rule base as well.

The IOC signature database is not encrypted or stored in a proprietary format. You can edit the signature database yourself and add your own IOCs. Be advised that attackers may also get access to these rules on the target systems if you use the scanner and leave the package on a compromised system.

Overview

You can easily add your own sample hashes, filename characteristics and Yara rules to the rulesets we bundled with it.

The most common use case is a so-called 'Triage' or 'APT Scan' scenario in which you scan all your machines to identify threats that haven't been detected by common Antivirus solutions. You can roll out LOKI like any other software using your preferred method or offer it on a network share. LOKI can then be started via Scheduled Task (GPO). You can simply run it using the UNC path '\\system\share\loki.exe'.

Another scenario is the use in a forensic lab. Scan mounted images with LOKI to identify known threats using the provided IOC definitions.

We quickly add IOCs derived from important threat reports to our rule sets (e.g. Regin, Skeleton Key). Use LOKI to check the integrity of your systems fast and target-oriented.

How does it do the job ?

Detection is based on four detection methods:

1. File Name IOC
Regex match on full file path/name
2. [Yara Rule](#) Check
Yara signature match on file data and process memory
3. Hash check
Compares known malicious hashes (MD5, SHA1, SHA256) with scanned files
4. C2 Back Connect Check
Compares process connection endpoints with C2 IOCs (new since version v.10)

Additional Checks:

1. Regin filesystem check (via -reginfo)
2. Process anomaly check (based on [Sysforensics](#))
3. SWF decompressed scan (new since version v0.8)
4. SAM dump check
5. DoublePulsar check - tries to detect DoublePulsar backdoor on port 445/tcp and 3389/tcp
6. [PE-Sieve](#) process check

The Windows binary is compiled with PyInstaller 2.1 and should run as an x86 application on both x86 and x64 based systems.

How-To run LOKI ?

<https://github.com/Neo23x0/Loki#run>

- Download the latest LOKI version from the [releases](#) section
- Run it once to retrieve the latest signature base repository
- Provide the folder to a target system that should be scanned: removable media, network share, folder on target system
- Right-click on loki.exe and select “Run as Administrator” or open a command line “cmd.exe” as Administrator and run it from there (you can also run LOKI without administrative privileges but some checks will be disabled and relevant objects on disk will not be accessible)

```

  _____
 /  _  _  \
/_  /_\  \
 \_  _  /
  \_\_\_

  _____
 /  _  _  \
/_  /_\  \
 \_  _  /
  \_\_\_

Copyright by Florian Roth, Released under the GNU General Public License
July 2017, Version 0.23.3

DISCLAIMER - USE AT YOUR OWN RISK
Please report false positives via https://github.com/Neo23x0/Loki/issues

[NOTICE] Starting Loki Scan SYSTEM: PROMETHEUS TIME: 20170906T06:57:29Z PLATFORM:
OWS
[INFO] File Name Characteristics initialized with 2518 regex patterns
[INFO] C2 server indicators initialized with 32804 elements
[INFO] Malicious MD5 Hashes initialized with 16214 hashes
[INFO] Malicious SHA1 Hashes initialized with 6552 hashes
[INFO] Malicious SHA256 Hashes initialized with 20691 hashes
[INFO] False Positive Hashes initialized with 30 hashes
[INFO] Processing YARA rules folder X:\Workspace\Loki\dist\loki\./signature-base
[INFO] Initializing Yara rule apt_agent_btz.yar
[INFO] Initializing Yara rule apt_alien spy_rat.yar
[INFO] Initializing Yara rule apt_apt10.yar
[INFO] Initializing Yara rule apt_apt17_malware.yar
[INFO] Initializing Yara rule apt_apt19.yar
[INFO] Initializing Yara rule apt_apt28.yar
[INFO] Initializing Yara rule apt_apt29_grizzly_steppe.yar
[INFO] Initializing Yara rule apt_apt30_backspace.yar
[INFO] Initializing Yara rule apt_apt6_malware.yar
[INFO] Initializing Yara rule apt_backdoor_ssh_python.yar
[INFO] Initializing Yara rule apt_backspace.yar
[INFO] Initializing Yara rule apt_beepservice.yar
[INFO] Initializing Yara rule apt_between-hk-and-burma.yar
[INFO] Initializing Yara rule apt_blackenergy.yar
[INFO] Initializing Yara rule apt_blackenergy_installer.yar
[INFO] Initializing Yara rule apt_blutermite_endivi.yar
[INFO] Initializing Yara rule apt_buckeye.yar
[INFO] Initializing Yara rule apt_carbon_paper_turla.yar
[INFO] Initializing Yara rule apt_casper.yar
```

How-To Analyse Reports ?

- The resulting report will show a GREEN, YELLOW or RED result line.
- Please analyze the findings yourself by:
 1. uploading non-confidential samples to [Virustotal.com](https://www.virustotal.com)
 2. Search the web for the filename
 3. Search the web for keywords from the rule name (e.g. EQUATIONGroupMalware_1 > search for “Equation Group”)
 4. Search the web for the MD5 hash of the sample
 5. Search in my [customer APT search engine](#) for file names or identifiers
- Please report back false positives via the [Issues](#) section (mention the false positive indicator like a hash and/or filename and the rule name that triggered)

```
[ALERT]
FILE: M:\websHELL\138shell\F\FatalShell.php.txt SCORE: 140 TYPE: PHP SIZE: 16375
FIRST_BYTES: 3c3f7068700a73657373696f6e5f737461727428 / <?phpsession_start(
MD5: b15583f4eaad10a25ef53ab451a4a2d6d
SHA1: 9da47055cc121171c1d820bc9185e69f0d9de1f3
SHA256: 6939898da48e7b6607e02143c78d3201ae21230a8e49225d1aac646a42d6df0 CREATED: Tue
Jul 19 18:18:38 2016 MODIFIED: Fri Oct 03 09:42:03 2014 ACCESSED: Thu Sep 07 11:18:1
0 2017
REASON_1: Yara Rule MATCH: _antichat_php_php_FatalShell_php_php_a_gedit_php_php SUBSC
ORE: 70
DESCRIPTION: Semi-Auto-generated - from files antichat.php.php.txt, FatalShell.php.p
hp.txt, a_gedit.php.php.txt
MATCHES: Str1: if(@$_POST['save'])writeF($file,$_POST['data']); Str2: if($action=="ph
peval"){ Str3: $uploadFile = $_dirupload."/".$_POST['filename']; ... (truncated)
REASON_2: Yara Rule MATCH: WebShell_Generic_PHP_3 SUBSCORE: 70
DESCRIPTION: PHP Webservers Github Archive
MATCHES: Str1: header('Content-Length:'.$filesize($file).'); Str2: <textarea name=""
command\\' rows="" cols=""$_.POST['comma Str3: i ... (truncated)
[WARNING]
FILE: M:\websHELL\138shell\F\Fuckphpshell.txt SCORE: 70 TYPE: PHP SIZE: 9478
FIRST_BYTES: 3c3f706870200a0a6572726f725f7265706f72 / <?php error_repor
MD5: 554e50c1265bb0934fcc8247ec3b9052
SHA1: b86cd7c940f98f92f4f54b5a32b2b7c445d2e91c
SHA256: cd3e1a5c6a57eb63d88f3c54e6b47b28dcc1e1057d31722d044c72da81cade5b CREATED: Tue
Jul 19 18:18:38 2016 MODIFIED: Fri Oct 03 09:42:03 2014 ACCESSED: Thu Sep 07 11:18:1
0 2017
REASON_1: Yara Rule MATCH: fuckphpshell_php SUBSCORE: 70
DESCRIPTION: Semi-Auto-generated - file fuckphpshell.php.txt
MATCHES: Str1: $succ = "Warning! Str2: Don't be stupid .. this is a priv3 server, so
take extra care! Str3: \\*--- MEMBERS AREA ---*/ Str4: preg_ma ... (truncated)
[ALERT]
FILE: M:\websHELL\138shell\G\GFS web-shell ver 3.1.7 - PRIU8.txt SCORE: 280 TYPE: UNK
NOWN SIZE: 24829
FIRST_BYTES: 3c3f0a2f2a0a2a2a2a2a2a2a2a2a2a2a2a / </?*****
MD5: be0f67f3e995517d18859ed57b4b4389
SHA1: 3b9eb1ae129ec61fda2ebb85f73c17a8d55d916a
SHA256: 907a3892d825218fb7bc253f1e42c77e859c545f5ba026160e62d81a9ab39c37 CREATED: Tue
Jul 19 18:18:38 2016 MODIFIED: Fri Oct 03 09:42:03 2014 ACCESSED: Thu Sep 07 11:18:1
0 2017
REASON_1: Yara Rule MATCH: websHELL_gfs_sh_r57shell_r57shell127_SnIpEr_SA_xxx SUBSCOR
E: 70
DESCRIPTION: Web Shell
```

At the end of the scan LOKI generates a scan result. This result can be:

- System seems to be clean :

```
LOKI
Simple IOC Scanner
(C) Florian Roth - BSK Consulting GmbH
Jan 2015
Version 0.3.0
DISCLAIMER - USE AT YOUR OWN RISK

[INFO] LOKI - Starting Loki Scan on PROMETHEUS
[INFO] File Name Characteristics initialized with 34 regex patterns
[INFO] File Name Suspicious Characteristics initialized with 51 regex patterns
[INFO] Malware Hashes initialized with 43 hashes
[INFO] False Positive Hashes initialized with 8 hashes
[INFO] Successfully compiled Yara rules from file thor-hacktools.yar
[INFO] Successfully compiled Yara rules from file thor-webshells.yar
[INFO] Successfully compiled Yara rules from file yara_rules.yar
[INFO] Scanning C:\Program Files ...
[RESULT] SYSTEM SEEMS TO BE CLEAN.
```

- Suspicious Objects detected :

```
LOKI
Simple IOC Scanner
(C) Florian Roth - BSK Consulting GmbH
Jan 2015
Version 0.3.0
DISCLAIMER - USE AT YOUR OWN RISK

[INFO] LOKI - Starting Loki Scan on PROMETHEUS
[INFO] File Name Characteristics initialized with 34 regex patterns
[INFO] File Name Suspicious Characteristics initialized with 51 regex patterns
[INFO] Malware Hashes initialized with 43 hashes
[INFO] False Positive Hashes initialized with 8 hashes
[INFO] Successfully compiled Yara rules from file thor-hacktools.yar
[INFO] Successfully compiled Yara rules from file thor-webshells.yar
[INFO] Successfully compiled Yara rules from file yara_rules.yar
[INFO] Scanning C:\ibm ...
[WARNING] File Name Suspicious IOC matched PATTERN: \\s\\.exe DESC: Suspicious File Name MATCH: C:\ibm\s.exe
[WARNING] File Name Suspicious IOC matched PATTERN: \\[a-zA-Z]\\.exe$ DESC: Suspicious File Name MATCH: C:\ibm\s.exe
[RESULT] SUSPICIOUS OBJECTS DETECTED!
[RESULT] Loki recommends a deeper analysis of the suspicious objects.
```

- Indicators detected :

```

LOKI

Simple IOC Scanner

<C> Florian Roth - BSK Consulting GmbH
Jan 2015
Version 0.3.0

DISCLAIMER - USE AT YOUR OWN RISK

[INFO] LOKI - Starting Loki Scan on PROMETHEUS
[INFO] File Name Characteristics initialized with 34 regex patterns
[INFO] File Name Suspicious Characteristics initialized with 51 regex patterns

[INFO] Malware Hashes initialized with 43 hashes
[INFO] False Positive Hashes initialized with 8 hashes
[INFO] Successfully compiled Yara rules from file thor-hacktools.yar
[INFO] Successfully compiled Yara rules from file thor-webshells.yar
[INFO] Successfully compiled Yara rules from file yara_rules.yar
[INFO] Scanning M:\sonstige3 ...
[ALERT] Yara Rule MATCH: Amplia_Security_Tool FILE: M:\sonstige3\getlsasrvaddr.e
xe
[ALERT] Yara Rule MATCH: Amplia_Security_Tool FILE: M:\sonstige3\md5.csv
[ALERT] Yara Rule MATCH: WindowsCredentialEditor FILE: M:\sonstige3\wce.exe
[ALERT] Yara Rule MATCH: Amplia_Security_Tool FILE: M:\sonstige3\wce.exe
[ALERT] Yara Rule MATCH: WCE_Modified_1_1014 FILE: M:\sonstige3\wce.exe
[ALERT] Yara Rule MATCH: WindowsCredentialEditor FILE: M:\sonstige3\wce64.exe
[ALERT] Yara Rule MATCH: Amplia_Security_Tool FILE: M:\sonstige3\wce64.exe
[ALERT] Yara Rule MATCH: WCE_Modified_1_1014 FILE: M:\sonstige3\wce64.exe

[RESULT] INDICATORS DETECTED!

[RESULT] Loki recommends a forensic analysis and triage with a professional tria
ge tool like THOR APT Scanner.

```

How-To Update LOKI ?

LOKI includes a separate updater tool named *loki-upgrader.exe* or *loki-upgrader.py* :

```

usage: loki-upgrader.py [-h] [-l log-file] [--sigsonly] [--progonly] [--nolog]
                        [--debug]

Loki - Upgrader

optional arguments:
  -h, --help            show this help message and exit
  -l log-file            Log file
  --sigsonly            Update the signatures only
  --progonly            Update the program files only
  --nolog               Don't write a local log file
  --debug               Debug output

```

It allows updating the compiled `loki.exe` for Windows and the signature-based sources.

When running `loki.exe -update` it will create a new upgrade process and exits LOKI in order to replace the `loki.exe` with the newer one, which would be locked otherwise.

How-To Use LOKI ?

```
usage: loki.exe [-h] [-p path] [-s kilobyte] [-l log-file] [-r remote-loghost]
               [-a alert-level] [-w warning-level] [-n notice-level]
               [--printAll] [--allreasons] [--noprocsan] [--nofilescan]
               [--scriptanalysis] [--rootkit] [--noindicator] [--reginfs]
               [--dontwait] [--intense] [--csv] [--onlyrelevant] [--nolog]
               [--update] [--debug]
```

Loki - Simple IOC Scanner

optional arguments:

-h, --help	show this help message and exit
-p path	Path to scan
-s kilobyte	Maximum file size to check in KB (default 5000 KB)
-l log-file	Log file
-r remote-loghost	Remote syslog system
-a alert-level	Alert score
-w warning-level	Warning score
-n notice-level	Notice score
--printAll	Print all files that are scanned
--allreasons	Print all reasons that caused the score
--noprocsan	Skip the process scan
--nofilescan	Skip the file scan
--scriptanalysis	Activate script analysis (beta)
--rootkit	Skip the rootkit check
--noindicator	Do not show a progress indicator
--reginfs	Do check for Regin virtual file system
--dontwait	Do not wait on exit
--intense	Intense scan mode (also scan unknown file types and all extensions)
--csv	Write CSV log format to STDOUT (machine proceessing)
--onlyrelevant	Only print warnings or alerts
--nolog	Don't write a local log file
--update	Update the signatures from the "signature-base" sub repository
--debug	Debug output

Scan: `loki.[exe|py] -p path_to_scan`

[illegible]

Scan output:

```
[WARNING]
FILE: C:\testing\AppData\mozilla\dre.bin SCORE: 80 TYPE: EXE SIZE: 66252
FIRST_BYTES: 4d5a90000300000004000000ffff0000b8000000 / MZ
MD5: d6278a53daea5f16e7d2fbec40d5438e
SHA1: 02df5a727b4b9299037600c17d844424ab0d249
SHA256: 4d54bec2da63ad3535e51c508db075025539349d7950607356f279ce127b163f CREATED: Mon Feb 16 19:20:09 2015 M
ODIFIED: Mon Nov 10 23:56:57 2014 ACCESSED: Mon Feb 16 19:20:09 2015
REASON 1: File Name IOC matched PATTERN: (application data|AppData|Anwendungsdaten)\\mozilla\\[^\\]+\\.bin SU
BSCORE: 80 DESC: Kaspersky Carbanak APT Malware Hash http://goo.gl/0Nhx2
[NOTICE]
FILE: C:\testing\excludes\temp.edb SCORE: 50 TYPE: EXE SIZE: 657392
FIRST_BYTES: 4d5a90000300000004000000ffff0000b8000000 / MZ
MD5: d8b7b276710127d233abedb7313aac36
SHA1: 27011d2fc22e894bd8a48de03a82b64f0bdbbabc
SHA256: 55a1e12963fed3b94e0c6817112dbdde5b2d24c2bc0d76e8435d0a5b108b9e57 CREATED: Sat Apr 18 12:51:19 2015 M
ODIFIED: Fri Jul 06 09:59:22 2012 ACCESSED: Sat Apr 18 12:51:19 2015
REASON 1: Vara Rule MATCH: HackTool_Producers SUBSCORE: 50
DESCRIPTION: Hacktool Producers String
MATCHES: Str1: gentilkiwi.com
[ALERT]
FILE: C:\testing\excludes\Exchange Server\ClientAccess\OAB\its_mimi.exe SCORE: 160 TYPE: EXE SIZE: 418304
FIRST_BYTES: 4d5a90000300000004000000ffff0000b8000000 / MZ
MD5: Tcb84e9f78557389202e25bd4b2b400bd
SHA1: cf89deb5fcb558930d73cdab18651ceabb8285bf6
SHA256: 3a04c554f8a5458a86bfd5e84ca5e4495e109dfaf857333677d899b15d722a70 CREATED: Thu Mar 24 10:57:55 2016 M
ODIFIED: Sun Jan 31 16:02:26 2016 ACCESSED: Thu Mar 24 10:57:55 2016
REASON 1: Vara Rule MATCH: Powerkatx_DLL_Generic SUBSCORE: 80
DESCRIPTION: Detects Powerkatx - a Mimikatz version prepared to run in memory via Powershell (overlap with o
ther Mimikatz versions is possible)
MATCHES: Str1: kuhl_m_lsadump_getUsersAndSamKey ; kull_m_registry_RegOpenKeyEx SAM Accounts (0x%08x) Str2: k
uhl_m_lsadump_getComputerAndSyskey ; kuhl ... (truncated)
REASON 2: Vara Rule MATCH: mimikatz SUBSCORE: 80
DESCRIPTION: mimikatz
MATCHES: Str1: L\x03\u00ffDI\u00ffDx03H\u00ffD Str2: L\u00ffD\u00ffDI\u00ffD\u00ffD\u00ffD\u00ffD\u00ffD\u00ffD\u00ffD
```


Hash-based IOCs

```
#
# LOKI CUSTOM EVIL HASHES
# This file contains MD5, SHA1 and SHA256 hashes and a short info like file name
# or hash origin
#
# FORMAT -----
#
# MD5;COMMENT
# SHA1;COMMENT
# SHA256;COMMENT
#
# EXAMPLES -----
#
# 0c2674c3a97c53082187d930efb645c2;DEEP PANDA Sakura Malware - http://goo.gl/R3e6eG
# 000c907d39924de62b5891f8d0e03116;The Darkhotel APT http://goo.gl/DuS7WS
# c0318cb12b827c03d556c8747b1e323225df97bdc4258c2756b0d6a4fd52b47;Operation SMN Hashes http://goo.gl/

# 563d1512178cec1f6a73c98d565c98fa;Cygwin nc.exe example

5d853a8de18d844a9ab269f3d51e5072;Five Eyes QUERTY Malware20120.dll.bin
cc8b737edb3f11c9c5dba57035c63103;Five Eyes QUERTY Malware20120.xml
67ac8dc6589a07d950bd12f534dc9789;Five Eyes QUERTY Malware20120_cmdDef.xml
40451f20371329b992fb1b85c754d062;Five Eyes QUERTY Malware20121.dll.bin
ff0afae5c68c5177ed0a3d6339810cae;Five Eyes QUERTY Malware20121.xml
1bc8f4d4f4551c6efbbb1fe9f965dca49;Five Eyes QUERTY Malware20121_cmdDef.xml
0ed11a73694999bcb45d18b4189f41ac2;Five Eyes QUERTY Malware20123.sys.bin
066b6253afc3ad0efe9a15cead4ef7d8;Five Eyes QUERTY Malware20123.xml
790d1b48e97985deb710a94eb927c27;Five Eyes QUERTY Malware20123_cmdDef.xml

ad61e8daeeba43e442514b177a1b41ad4b7c6727;Skeleton Key Malware
5083b17cccc0dd0557dfc544f84e2ab55d6acd92;Skeleton Key Malware

20831a820a5f5f41353b5afab659f2ad42ec6df5d9692448872f3ed8bbb40ab92;Regin Malware Sample
225e9596de85ca7b1025d6e444f6a01aa6507feef213f4d2e20da9e7d5d8e430;Regin Malware Sample
392f32241cd3448c7a435935f2ff0d2cdc609dda81dd4946b1c977d25134e96e;Regin Malware Sample
40c46bcab9acc0d6d235491c01a66d4c6f35d884c19c6f410901af6d1e33513b;Regin Malware Sample
4139149552b0322f2c5c993abccc0f0d1b38db4476189a9f9901ac0d57a656be;Regin Malware Sample
4e39bc95c5323ab586d740725a1c8cbcd0e1fe453f7c4cac7cced9a26e42cc9;Regin Malware Sample
5001793790939009355ba841610412e0f8d60ef5461f2ea272ccf4fd4c83b823;Regin Malware Sample
5c81cf8262f9a8b0e100d2a220f7119e54edfc10c4fb906ab7848a015cd12d90;Regin Malware Sample
753d4a5914af58b23a9e0ce6a262cd230ed8bb2c30da3d42d26b295f9144ab7;Regin Malware Sample
7d38eb24cf5644e090e45d5efa923aff0e69a60fb0ab627e8929bb485243926;Regin Malware Sample
8098938987e2f29e3ee416b71b932651f6430d15d885f2e1056d41163ae57c13;Regin Malware Sample
```

Filename-based IOCs

```
#
# LOKI File Name Characteristics
# This file contains regex definitions and a description
#
# APPLICATION -----
#
# Every line is treated as REGEX case sensitive.
# Every line includes a description that gives information about the file name
# based IOC
#
# FORMAT -----
#
# # COMMENT
# REGEX;DESCRIPTION
#
# EXAMPLES -----
#
# # Various examples from APT case X
# \\svcsstat.exe;Case 1 - infector
# \\(server|services|smrr|srrm|svchost|svhost|svshost|taskmgr)\.exe$;Common IOC
# ProgramData\\Mail\\MailAg\\.Case 2
# (Anwendungsdaten|Application Data|APPPDATA)\\sydmain.dll;Malware X Case 3
# (TEMP|Temp)\\(F\\|F\\).*(cmd|yls)$;Case 2
# (LOCAL SETTINGS\\Temp|Local Settings\\Temp|Local\\Temp)\\(word\\.exe|winword\\.exe);Case 2
#
# Ncat Example
# bin\\nc.exe;Ncat Demo
#
# Regin
# \\usbclass\\.sys;File name known from REGIN malware
# \\adpu160\\.sys;File name known from REGIN malware
# \\msrdc64\\.dat;File name known from REGIN malware
# \\msdcsvc\\.dat;File name known from REGIN malware
# \\config\\SystemAudit\\.Evt;File name known from REGIN malware
# \\config\\SecurityAudit\\.Evt;File name known from REGIN malware
# \\config\\SystemLog\\.evt;File name known from REGIN malware
# \\config\\ApplicationLog\\.evt;File name known from REGIN malware
# \\ime\\imesc5\\dicts\\pintlgbp\\.imd;File name known from REGIN malware
# \\ime\\imesc5\\dicts\\pintlgbp\\.imd;File name known from REGIN malware
# ystem32\\winhttp\\.dll;File name known from REGIN malware
# ystem32\\wshnetc\\.dll;File name known from REGIN malware
# \\SysWow64\\wshnetc\\.dll;File name known from REGIN malware
# ystem32\\svcsstat\\.exe;File name known from REGIN malware
# ystem32\\svcsstat\\.exe;File name known from REGIN malware
# IME\\IMESC5\\DICTS\\PINTLGBP\\.IMD;File name known from REGIN malware
# ystem32\\wsharp\\.dll;File name known from REGIN malware
# ystem32\\wshnetc\\.dll;File name known from REGIN malware
# pchealth\\helpectr\\Database\\cdata\\.dat;File name known from REGIN malware
# pchealth\\helpectr\\Database\\cdata\\.edb;File name known from REGIN malware
# Windows\\Panther\\setup\\.etl\\.000;File name known from REGIN malware
# ystem32\\wbem\\repository\\INDEX2\\.DATA;File name known from REGIN malware
# ystem32\\wbem\\repository\\OBJECTS2\\.DATA;File name known from REGIN malware
# ystem32\\dnscache\\.dat;File name known from REGIN malware
# ystem32\\mregnx\\.dat;File name known from REGIN malware
# ystem32\\dispn32\\.dat;File name known from REGIN malware
# ystem32\\dmidskwk\\.dat;File name known from REGIN malware
# ystem32\\nvvrnsnu\\.dat;File name known from REGIN malware
# ystem32\\tapiscfg\\.dat;File name known from REGIN malware
# ystem32\\pciclass\\.sys;File name known from REGIN malware
```

Package LOKI with a private Rule Set

LOKI can be packaged with a custom encrypted rule set, which is embedded in the pyinstaller package. In order to include your own rules, place them in a directory named private-signatures in the LOKI directory and execute *build.bat*

```
loki/
├─ private-signatures/ <-- YARA rules places in here will by added to loki.exe
├─ signature-base/    <-- clear text and still required (retrieved by loki-upgrader.exe)
│   └─ iocs/
│   └─ yara/
```

In order to successfully run the build script, you need to install PyInstaller. We use PyInstaller 2.1 due the problem that Packages built with PyInstaller 3 don't run on Windows 2003 and XP based systems. (yes, we need that in incident response - there are even productive systems out there running Windows 2000 or Windows NT).

The easiest way to do install PyInstaller is:

pip install pyinstaller==2.1

After that, you can just run the build script :

build.bat

You can verify whether the signature set is valid by calling loki-package-builder.py manually :

C:\Python27[-x64]\python.exe loki-package-builder.py --ruledir signatures --target rules

The usage of this tool is:

```
usage: loki-package-builder.py [-h] --ruledir RULEDIR --target TARGET

Package builder for Loki

optional arguments:
  -h, --help            show this help message and exit
  --ruledir RULEDIR     directory containing the rules to build into Loki
  --target TARGET       target where to store the compiled ruleset
```

Add Signature and IOCs

The IOC files for hashes and filenames are stored in the './signature-base/iocs' folder. All '.yar' files placed in the './signature-base/yara' folder will be initialized together with the rule set that is already included. Use the 'score' value to define the level of the message upon a signature match.

You can add hash, c2 and filename IOCs by adding files to the './signature-base/iocs' subfolder. All hash IOCs and filename IOC files must be in the format used by LOKI (see the default files). The files must have the strings "hash", "filename" or "c2" in their name to get pulled during initialization.

For Hash IOCs (divided by newline; hash type is detected automatically) :

Hash;Description [Reference]

For Filename IOCs (divided by newline) :

Description [Reference]

Regex;Score;False Positive Regex

You can use the following external variables in the YARA rules that your provide LOKI :

- filename - e.g. condition: \$s1 and not filename == 'nmap.exe'
- filepath - e.g. condition: filepath == 'C:\Windows\cmd.exe'
- extension - e.g. condition: uint32(0) == 0x5a4d and extension == ".txt"
- filetype - eg. condition: extension == ".txt" and filetype == "EXE"
- (see file-type-signatures.cfg in signature-base repo for all detected file types)
- md5 - legacy value

Threat Intel Receivers

Since version v0.10 LOKI includes various threat intel receivers using the public APIs of these services to retrieve and store the IOCs in a format that LOKI understands. It is no problem if these indicators overlap with the ones already included. Loki uses a filename regex or hash only once. (no performance impact).

The threat intel receivers have also been moved to the [signature-base](#) sub repository with version 0.15 and can be found in `"/signature-base/threatintel"`.

Provide your API key via `-k APIKEY` or set it in the script header.

Open Threat Exchange (OTX) Receiver

It's a simple script that downloads your subscribed events/iocs from [Alienvault OTX](#) and stores them in the correct format in the `'./iocs'` subfolder. The script is located in the `"/threatintel"` folder and is named `"get-otx-iocs.py"` :

```
usage: get-otx-iocs.py [-h] [-k APIKEY] [-o dir] [--verifycert] [--debug]

OTX IOC Receiver

optional arguments:
  -h, --help            show this help message and exit
  -k APIKEY             OTX API key
  -o dir                Output directory
  --verifycert          Verify the server certificate
  --debug              Debug output
```

MISP Receiver

A simple script that downloads your subscribed events/iocs from a custom [MISP](#) instance and stores them in the correct format in the `'./iocs'` subfolder. YARA rules stored in MISP will be written to the `'./iocs/yara'` subfolder and automatically initialized during startup. The script is located in the `"/threatintel"` folder and is named `"get-misp-iocs.py"` :

```
usage: get-misp-iocs.py [-h] [-u URL] [-k APIKEY] [-l tframe] [-o dir]
                        [-y yara-dir] [--verifycert] [--debug]

MISP IOC Receiver

optional arguments:
  -h, --help            show this help message and exit
  -u URL                MISP URL
  -k APIKEY             MISP API key
  -l tframe             Time frame (e.g. 2d, 12h - default=30d)
  -o dir                Output directory
  -y yara-dir           YARA rule output directory
  --verifycert          Verify the server certificate
  --debug              Debug output
```

LOKI vs THOR

	LOKI	THOR
Type	Free / Open Source	Enterprise Product
Main Use Case	Preventive Scanning / Triage	Incident Response / Live Forensics
Platform	Windows (precompiled), Linux / macOS (source with dependencies)	Windows
Size (Binaries)	8 MB	16 MB
Language	Python	Python
Modules	3	26
Bundled Signatures	Open Source (~3000 YARA rules)	THOR's Signature Set (~9000 YARA rules)
Support and Testing	Github README & Issues, Travis-CI	Manual & Support Portal, Internal CI
Special Extras	Levenshtein check PESieve check Double Pulsar check	... a lot, see comparison



THOR

1871	Filename Characteristics
14515	Malicious Hashes
310	Hacktools (Yara)
633	Web Shells (Yara)
466	Generic APT Rules (Yara)
149	Customer APT Rules (Yara)
732	Malware, Exploits, Trojans, Negative Rules, Java, SCADA
...	



LOKI

101	Filename Characteristics
74	Malicious Hashes
185	Hacktools (Yara)
612	Web Shells (Yara)
52	Generic APT Rules (Yara)