THOR Util Manual

Nextron Systems GmbH

May 18, 2026

CONTENTS:

CHAPTER

ONE

WHAT IS THOR UTIL?

THOR Util provides maintenance functions such as updates, package downloads, and license retrieval. It also supports executable signature verification, custom signature encryption, report generation, and diagnostics for troubleshooting THOR scans.

CHAPTER

TWO

UPGRADE (UPGRADE) AND UPDATES (UPDATE)

You can download updates for THOR with `thor-util.exe` (Windows) or `thor-util` (Linux, macOS).

Note

THOR Util cannot update THOR binaries and signatures whose license was obtained through the ASGARD Management Center. Use the Management Center API for these updates. THOR Util only accepts the Nextron Customer Portal as a license source.

Running `thor-util --help` shows three options that seem to have a very similar meaning: "upgrade", "update" and "download".

The "download" option downloads a full package including all configuration files. The "upgrade" option fetches a full package but excludes configuration files to avoid overwriting local files such as `thor.yml` or `falsepositive_filters.cfg`.

The "update" option retrieves only the newest signature pack (not the program files).

Option	Description
upgrade	Get new program files and signatures
update	Get new signatures
download	Get new program files, signatures, and configuration files

If a full program package is already present, use the "upgrade" option.

Each option has its own help. You can view it with:

```
user@unix:~/thor$ ./thor-util *option* --help
```

The following examples show different upgrade methods:

```
C:\thor>thor-util.exe upgrade
C:\thor>thor-util.exe upgrade -a https://proxy.company.net:8080
C:\thor>thor-util.exe upgrade -a https://proxy.company.net:8080 -n dom\\user -p password
C:\thor>thor-util.exe upgrade -a https://proxy.local:8080 --ntlm -n dom\\user -p password
```

2.1 THOR TechPreview Version

To upgrade your current version to the TechPreview version, use this command:

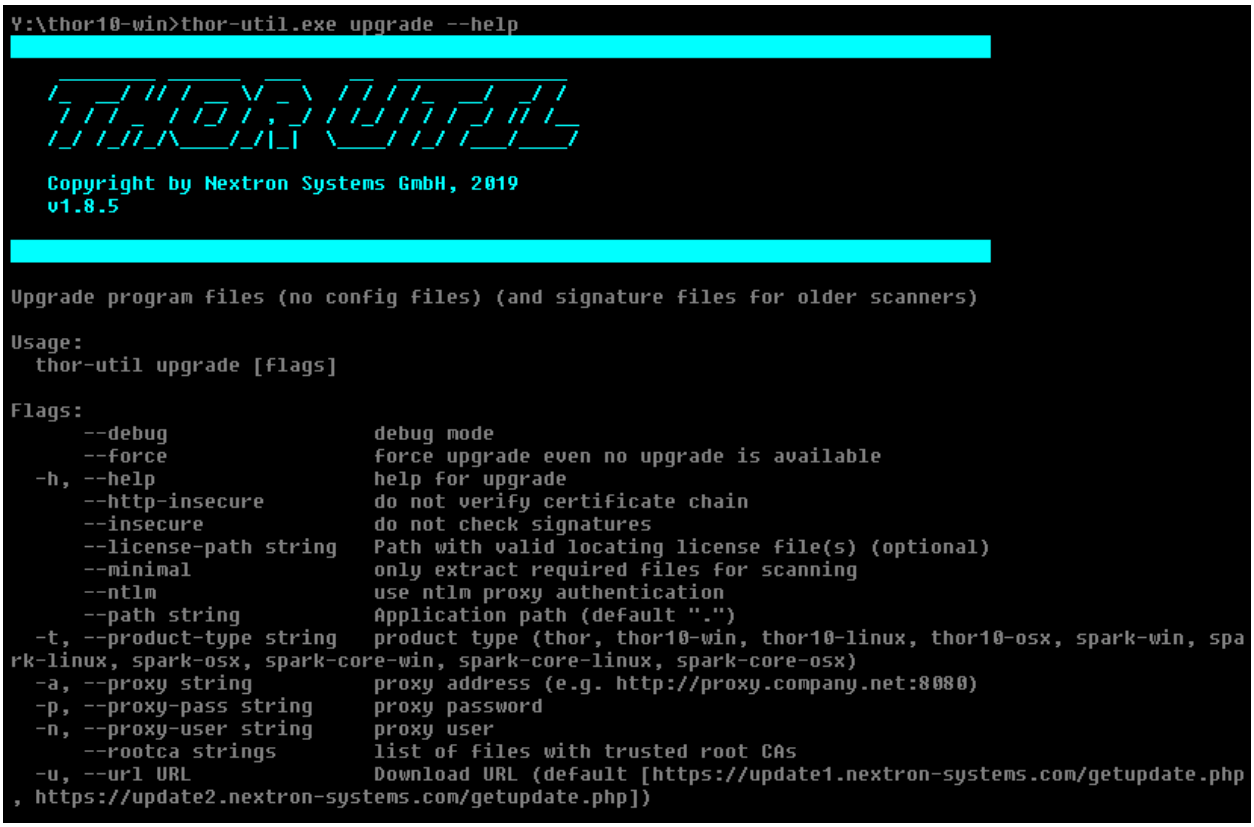


Fig. 1: THOR-util Upgrade Help

```
C:\thor>thor-util.exe upgrade --techpreview
```

You can find more information on the TechPreview version here.

> **Hint**
>
> To make the TechPreview version persistent, consider adding it to your THOR Util configuration file (`thor-util.yml`). Please see *TechPreview configuration*.

2.2 Update Locations

When using the full version of THOR, the following update mirrors must be accessible via HTTPS:

```
update1.nextron-systems.com
update2.nextron-systems.com
```

When using THOR Lite, the following server must be accessible:

```
update-lite.nextron-systems.com
```

> **Hint**
>
> For a detailed and up-to-date list of our update and licensing servers, please visit https://www.nextron-systems.com/hosts/.

2.3 SigDev Signatures

New signatures are usually validated for 1-2 days before they are published as stable. In rare cases, such as a newly discovered severe threat or public proof-of-concept code, you may want to use the latest SigDev signatures while they are still in validation.

To retrieve the latest SigDev signatures, use the `thor-util.exe update --sigdev` flag.

To reset the signature set to the latest stable version, use `thor-util.exe update --force`. This retrieves the stable set and enforces the download even if the current set is newer.

2.4 Update Server Information

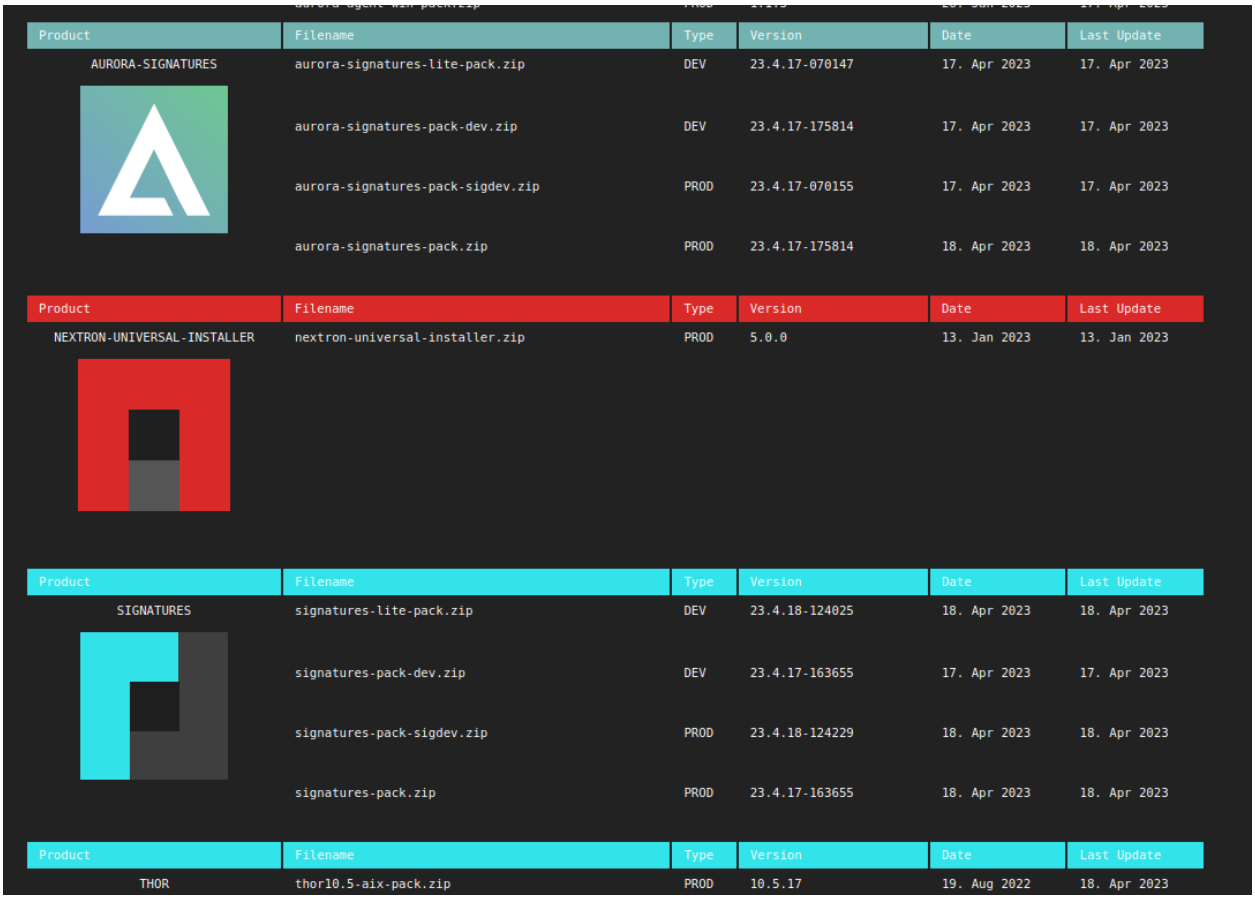
You can view information about the available update packages on this site:

https://update1.nextron-systems.com/info.php

2.5 Upgrading to a specific version

To upgrade to a specific THOR version, use the `--version` flag:

```
C:\thor>thor-util.exe upgrade --version 10.7.27
```



Product	Filename	Type	Version	Date	Last Update
AURORA-SIGNATURES	aurora-signatures-lite-pack.zip	DEV	23.4.17-070147	17. Apr 2023	17. Apr 2023
	aurora-signatures-pack-dev.zip	DEV	23.4.17-175814	17. Apr 2023	17. Apr 2023
	aurora-signatures-pack-sigdev.zip	PROD	23.4.17-070155	17. Apr 2023	17. Apr 2023
	aurora-signatures-pack.zip	PROD	23.4.17-175814	18. Apr 2023	18. Apr 2023

Product	Filename	Type	Version	Date	Last Update
NEXTRON-UNIVERSAL-INSTALLER	nextron-universal-installer.zip	PROD	5.0.0	13. Jan 2023	13. Jan 2023

Product	Filename	Type	Version	Date	Last Update
SIGNATURES	signatures-lite-pack.zip	DEV	23.4.18-124025	18. Apr 2023	18. Apr 2023
	signatures-pack-dev.zip	DEV	23.4.17-163655	17. Apr 2023	17. Apr 2023
	signatures-pack-sigdev.zip	PROD	23.4.18-124229	18. Apr 2023	18. Apr 2023
	signatures-pack.zip	PROD	23.4.17-163655	18. Apr 2023	18. Apr 2023

Product	Filename	Type	Version	Date	Last Update
THOR	thor10.5-aix-pack.zip	PROD	10.5.17	19. Aug 2022	18. Apr 2023

Fig. 2: Update server information

> **Hint**
>
> It is also possible to specify more complex expressions instead of a single version. The full syntax is described here. To e.g. upgrade to the latest THOR 10, you can use `upgrade --version ^10.0.0`.

When `--version` is used, `--techpreview` is ignored.

CHAPTER

THREE

DOWNLOAD PACKAGES (DOWNLOAD)

Use the "download" command to download scanner packages for Windows, Linux, and macOS.

> **Note**
>
> THOR Util cannot download THOR binaries and signatures whose license was obtained through the ASGARD Management Center. Use the Management Center API for these downloads. THOR Util only accepts the Nextron Customer Portal as a license source.

This option is useful when you need to download updates on an Internet-connected machine and transfer them to a system without Internet access.

```
C:\thor>thor-util.exe download -t thor10-win
```

3.1 THOR TechPreview Version

To download the TechPreview version, use the following command-line flag:

```
C:\thor>thor-util.exe download -t thor10-win --techpreview
```

You can find more information on the TechPreview version here.

CHAPTER

FOUR

INSTALL PACKAGES (INSTALL)

Use the "install" command to install packages that have already been downloaded.

You can obtain packages by:

- using the "download" command in THOR Util
- using the URL shown during update or upgrade procedures

This command is often used to update THOR program folders on systems without Internet access.

CHAPTER

FIVE

CUSTOM SIGNATURE ENCRYPTION (ENCRYPT)

Use THOR Util's "encrypt" command to encrypt YARA signatures and IOC files.

> **Note**
>
> THOR Util and THOR Util Lite use different signature encryption methods. Custom signatures must therefore be encrypted using the corresponding version of THOR Util.

```
C:\thor>thor-util.exe encrypt --help
```

```
C:\Users\neo\Downloads\build 8>thor-util encrypt --help
--------------------------------------------------------------------------------
   ________ ______  ___    __  ____  _ __
  /_  __/ // / __ \/ _ \  / / / / /_(_) /
   / / / _  / /_/ / , _/ / /_/ / __/ / /
  /_/ /_//_/\____/_/|_|  \____/\__/_/_/

   THOR / SPARK Update and License Utility
   Copyright by Nextron Systems GmbH, May 2018
   v1.2.3

--------------------------------------------------------------------------------

Encrypt signature files

Usage:
  thor-util encrypt <file|files> [flags]

Examples:
  thor-util encrypt foo.txt
  thor-util encrypt foo.txt bar.txt baz.txt
  thor-util encrypt *.txt

Flags:
      --debug            debug mode
      --destdir string   output directory (otherwise encrypted file will be written to sour
ce directory)
  -h, --help             help for encrypt
```

Fig. 1: THOR Util's Encrypt Feature Help

The "encrypt" command accepts a single file, a list of files, or wildcards as input.

```
C:\thor>thor-util.exe encrypt ~/sigs/case14.yar
C:\thor>thor-util.exe encrypt ~/sigs/case14.yar ~/sigs/case14-hashes.txt
C:\thor>thor-util.exe encrypt ~/sigs/case14.\*
```

THOR Util automatically detects the signature type based on the file extension.

File Type	Clear Text Extension	Extension of Encrypted File
IOC File	.txt	.dat
YARA Rule	.yar, .yara, .yac (compiled YARA)	.yas
Sigma	.yml, .yaml	.yms
STIXv2	.json	.jsos

Place encrypted IOC files in the `./custom-signatures` subfolder in the program directory and encrypted YARA rules in the `./custom-signatures/yara` subfolder.

CHAPTER

SIX

REPORT GENERATION (REPORT)

Use the `report` command to generate an HTML report from plain text log files.

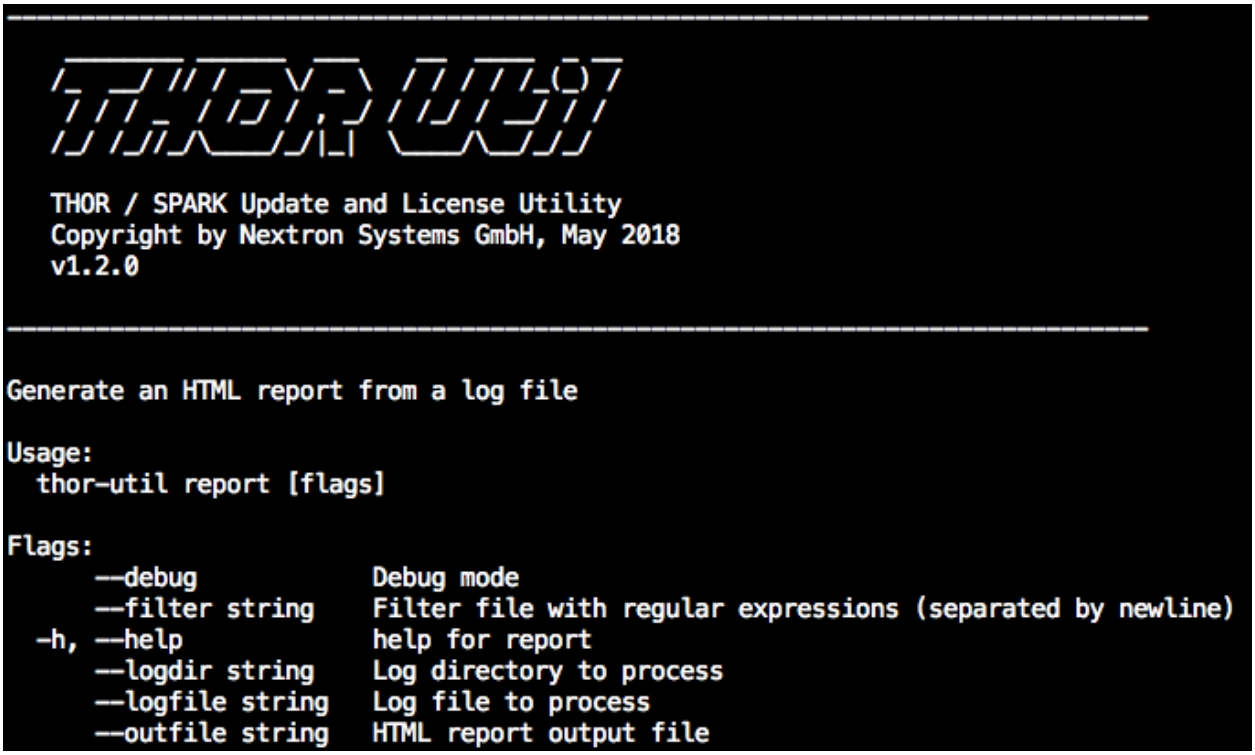
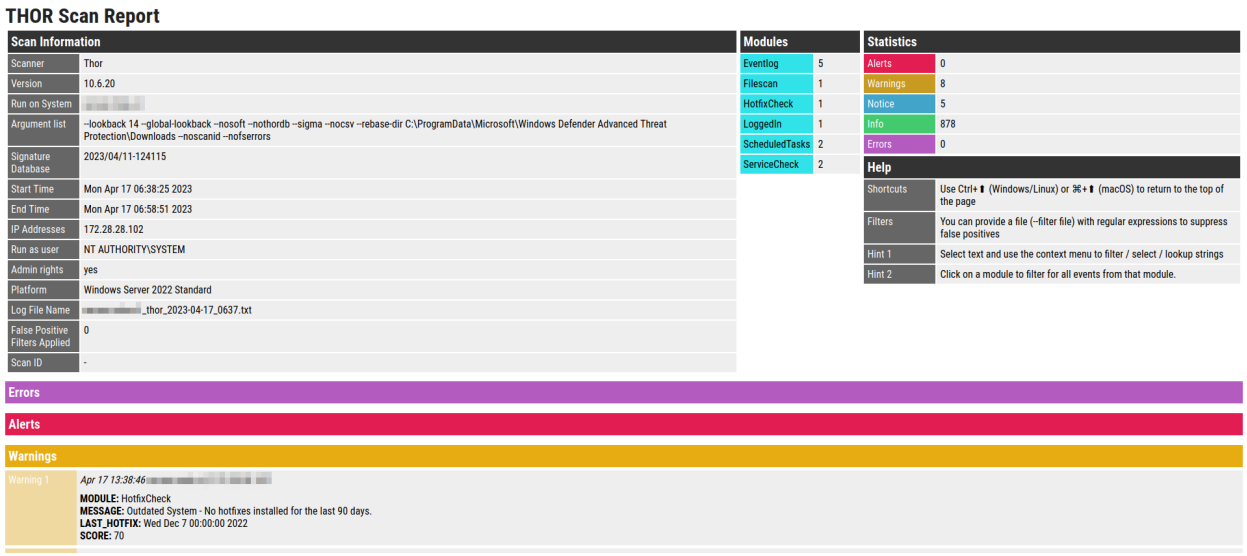


Fig. 1: THOR Util's report generation functions

```
user@unix:~/thor$ ./thor-util report --logfile system-xyz_thor.log
user@unix:~/thor$ ./thor-util report --logdir ./logs
```

See this blog post for details:

https://www.nextron-systems.com/2018/06/20/thor-util-with-html-report-generation/



THOR Scan Report

Scan Information	
Scanner	Thor
Version	10.6.20
Run on System	[redacted]
Argument list	--lookback 14 --global-lookback --nosoft --nothordb --sigma --nocsv --rebase-dir C:\ProgramData\Microsoft\Windows Defender Advanced Threat Protection\Downloads --noscanid --nofserrors
Signature Database	2023/04/11-124115
Start Time	Mon Apr 17 06:38:25 2023
End Time	Mon Apr 17 06:58:51 2023
IP Addresses	172.28.28.102
Run as user	NT AUTHORITY\SYSTEM
Admin rights	yes
Platform	Windows Server 2022 Standard
Log File Name	[redacted]_thor_2023-04-17_0637.txt
False Positive Filters Applied	0
Scan ID	-

Modules	
Eventlog	5
Filescan	1
HotfixCheck	1
LoggedIn	1
ScheduledTasks	2
ServiceCheck	2

Statistics	
Alerts	0
Warnings	8
Notice	5
Info	878
Errors	0

Help	
Shortcuts	Use Ctrl+⬆ (Windows/Linux) or ⌘+⬆ (macOS) to return to the top of the page
Filters	You can provide a file (--filter file) with regular expressions to suppress false positives
Hint 1	Select text and use the context menu to filter / select / lookup strings
Hint 2	Click on a module to filter for all events from that module.

Errors

Alerts

Warnings

Warning 1

Apr 17 13:38:46 [redacted]

MODULE: HotfixCheck
MESSAGE: Outdated System - No hotfixes installed for the last 90 days.
LAST_HOTFIX: Wed Dec 7 00:00:00 2022
SCORE: 70

Fig. 2: HTML report generated by thor-util

CHAPTER

SEVEN

VERIFY BINARIES (VERIFY)

Use the "verify" command to verify the authenticity of the included binaries. Signature verification is based on a public key encryption algorithm and requires the `*.sig` files shipped with the packages.

Fig. 1: Verify thor.exe signature using THOR Util

To verify the integrity of THOR Util, download the public key from Nextron's website: https://www.nextron-systems.com/pki/ You can then use the public key with the following commands to verify `thor-util`:

On Windows:

```
C:\thor>openssl dgst -sha256 -verify codesign.pem -signature thor-util.exe.sig thor-util.
↪exe
```

On Linux:

```
C:\thor>openssl dgst -sha256 -verify codesign.pem -signature thor-util.sig thor-util
```

CHAPTER

EIGHT

DECRYPT REPORTS AND LOG FILES (DECRYPT)

Use the "decrypt" command to decrypt HTML reports or text log files that THOR encrypted when the scan completed.

```
Decrypt log- and csv-files from THOR/SPARK

Usage:
  thor-util decrypt <file|files> [flags]

Examples:
  thor-util decrypt --keyfile /path/to/privatekey.pem scan-results.log

Flags:
      --debug            debug mode
  -h, --help             help for decrypt
      --privkey string   RSA Private Key in PEM Format (privkey="<key>" or privkey="<file>")
```

Fig. 1: THOR Util's decryption feature options

CHAPTER

NINE

LOG CONVERSION (LOGCONVERT)

The log conversion feature allows you to convert THOR logs between different formats. Choose the format that fits your needs:

Format	Convert From	Convert To
Log[1]	Yes	Yes
JSON	Yes	Yes
Key-Value	No[2]	Yes
CSV	No	Yes
ZIP CSV	No	Yes

```
C:\nextron\thor>thor-util.exe logconvert --help
    ________ ______  ___    __  _______________
   /_  __/ // / __ \/ _ \  / / / /_  __/  _/ /
    / / / _  / /_/ / , _/ / /_/ / / / _/ // /__
   /_/ /_//_/\____/_/|_|  \____/ /_/ /___/____/

   Copyright by Nextron Systems GmbH, 2023
   v1.11.0+thor10.7.6

Convert log file into another format

Usage:
  thor-util logconvert [flags]

Examples:
  thor-util logconvert --from-json --to-log --file example.json --output example.log

Flags:
  -f, --file string     Input file
      --from-json       Convert from JSON
      --from-kv         Convert from KV
      --from-log        Convert from Log
  -h, --help            help for logconvert
  -o, --output string   Output file
      --to-csv          Convert to CSV
      --to-csv-zip      Convert to ZIP containing one CSV log per module
```

(continues on next page)

[1] This is the default THOR log format `<hostname>_timestamp.txt`.
[2] The help menu shows this flag, but the function is not implemented yet.

(continued from previous page)

```
    --to-json         Convert to JSON
    --to-kv           Convert to KV
    --to-log          Convert to Log
```

Note

The feature to convert logs into CSV and ZIP CSV was introduced in THOR Util version 1.11.0.

9.1 Conversion Examples

The following examples show how to convert logs to different formats.

Each command uses the same structure: a `--from` format, a `--to` format, an input file with `-f`, and an output file with `-o`.

```
user@unix:~/thor$ ./thor-util logconvert --from-log --to-json -f thor.txt -o thor-
↪converted.json
user@unix:~/thor$ ./thor-util logconvert --from-log --to-csv -f thor.txt -o thor-
↪converted.csv
user@unix:~/thor$ ./thor-util logconvert --from-json --to-log -f thor.json -o thor-
↪converted.log
user@unix:~/thor$ ./thor-util logconvert --from-log --to-csv -f thor.txt -o thor-
↪converted.csv
user@unix:~/thor$ ./thor-util logconvert --from-log --to-csv-zip -f thor.txt -o thor-
↪converted.zip
```

CHAPTER

TEN

TEMPLATES

THOR Util reads a default configuration from `config/thor-util.yml`.

Use this file to set default parameters in YAML format.

Command-line flags override these default parameters.

All global flags for THOR Util are supported in the configuration file. You can view these flags with:

```
user@unix:~/thor$ ./thor-util --help
```

10.1 Proxy configuration

To use a specific HTTP proxy, add it to your configuration file:

```
proxy: http://myproxy:8080
```

10.2 TechPreview configuration

To always download the latest TechPreview instead of the standard THOR version, add:

```
techpreview: True
```

CHAPTER

ELEVEN

DIAGNOSTICS

If THOR does not behave as expected, for example if it uses more resources than expected, takes a long time to finish a scan, or exits unexpectedly with a generic error, you can create a diagnostics pack for Nextron Support.

Create the diagnostics pack with THOR Util's diagnostics command.

```
C:\thor>thor-util.exe help diagnostics

Create diagnostics pack

Usage:
  thor-util diagnostics [flags]

Flags:
  -h, --help        help for diagnostics
  --output string   File to write diagnostics pack to (default "[...]\diagnostics.zip")
  --run             Rerun last THOR scan with debug logging before collecting␣
↪diagnostics pack
```

By default, THOR Util writes the `diagnostics.zip` file to THOR's working directory. The location is printed on the command line after data collection finishes and can be changed with the `--output` flag.

11.1 Get diagnostics for a running THOR scan

The preferred way to collect THOR diagnostics is to run THOR Util's diagnostics command while the issue is occurring. Use this method if you suspect that THOR is stuck during a scan, that THOR has high memory or CPU usage, or that another issue occurs during runtime.

```
C:\thor>thor-util.exe diagnostics
```

11.2 Get diagnostics for a finished THOR scan

If the THOR run has already finished or stopped unexpectedly, you can also use the diagnostics command shown above. However, only a limited amount of diagnostic data can be collected after the scan has ended. In those cases, use the `--run` flag to rerun the last THOR scan. The `--run` flag is the preferred method if THOR exits unexpectedly or intermittently.

```
C:\thor>thor-util.exe diagnostics --run
```

11.3 What data is collected

THOR Util's diagnostics function collects the following data:

- A log of the THOR Util diagnostics run
- Go profiles for CPU, memory, and goroutines, see: https://go.dev/blog/pprof
- THOR's running configuration parameters
- A process list of all running processes on the machine. This helps identify processes that might interfere with THOR, such as an AV/EDR.
- A process dump of the running THOR instance
- The progress state of the running THOR instance
- A dump of the THOR DB
- The latest THOR log

Hint

Critical or personal information may be present in the THOR log, THOR DB dump, running process list, in the THOR process dump, and in the progress report (working item details like path information). The profiles may indicate what type of data is being scanned but do not contain specific pieces of data.

The diagnostics pack is only used to investigate the issue you are facing with THOR and will be deleted from our systems after the root cause has been identified.

CHAPTER

TWELVE

YARA FORGE

YARA-Forge (https://yarahq.github.io/) is an open-source project that bundles YARA rules from different open-source projects. Rules are offered in different **rulesets** that differ in their false positive (FP) ratio and detection rate trade-off.

THOR Util supports downloading YARA Forge with:

```
C:\thor>thor-util.exe yara-forge download --ruleset <ruleset>
```

The **ruleset** value can be one of the following:

- core
- extended
- full

> **Note**
>
> Only one ruleset at a time can be used. When you download a new ruleset, the old one is overwritten.

A downloaded YARA Forge ruleset is stored in `custom-signatures/yara-forge` and is automatically updated with `thor-util update`.

If you no longer want to use YARA Forge, you can run:

```
C:\thor>thor-util.exe yara-forge remove
```

CHAPTER

THIRTEEN

INDICES AND TABLES

- search