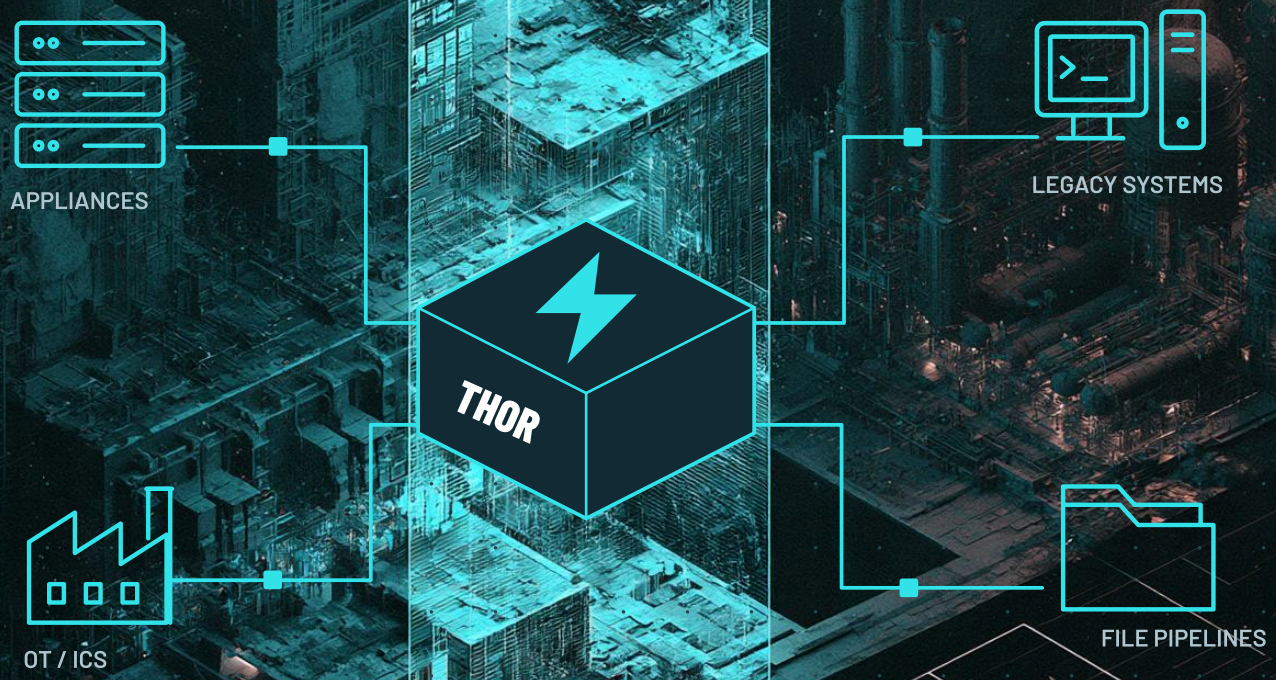
nextron systems

THOR THUNDERSTORM

SCAN BEYOND THE ENDPOINT.



COLLECT FILES.
SCAN CENTRALLY.
INVESTIGATE.

MOVE THE FILES. NOT THE SCANNER.

The source collects. Your central THOR service analyses.

AT THE SOURCE

Collector or script

Choose files. Read them.
Submit copies.

Original files stay here.

HTTP(S)

ON YOUR SERVER

THOR Thunderstorm

Apply detections.
Process forensic artefacts.

Analysis runs here.

NEED THE ANSWER NOW?

Synchronous submission returns the scan result to the caller.

COLLECTING IN BULK?

Asynchronous submission returns an ID. Files enter the queue; results follow.

YOUR FILES. YOUR NETWORK. YOUR ANALYSIS.

WHERE ENDPOINT COVERAGE STOPS.

Appliances. Old operating systems. Systems with no room for another scanner.

01

APPLIANCES & HYPERVISORS

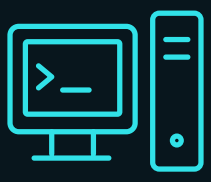
Collect files from network devices, embedded Linux or virtualization hosts using a suitable tool or export method.

02

INDUSTRIAL ENVIRONMENTS

Keep the analysis on a separate server. Scope collection around availability requirements and device restrictions.

03



LEGACY OPERATING SYSTEMS

Use a compatible binary or an available runtime. Older Python and PowerShell environments have dedicated script options.

OLDER SYSTEMS. CURRENT DETECTION.

A WEBSHELL IS STILL A FILE.

Even when you cannot run THOR on the system that holds it.

1

```
SOURCE / appliance-01

/web/
  index.jsp
  login.jsp
  shell.jsp
```

Collect selected web-root files and submit copies.

2

THOR THUNDERSTORM

Apply detection logic on the central service.

3

ILLUSTRATIVE FINDING

Webshell detected

```
Source: appliance-01
File: /web/shell.jsp
```

Investigate the source and file. Keep relevant samples and send findings to your SIEM through configured outputs.

THOR DETECTION. AS A WEB SERVICE.

Apply detection content and forensic processing to the files you submit.

YARA

Match Nextron and custom rules against submitted file content.

SIGMA

Apply Sigma detections to supported event-log data.

IOCs

Check incoming samples against custom hash and filename indicators.

FORENSIC FILES HAVE STRUCTURE. THOR CAN USE IT.

- EVTX event logs
- Registry hives
- Memory dumps

PURE YARA

Focused rule matching.

FULL-FEATURED

Adds parsing and forensic processing.

SMALL COLLECTOR. BIG REACH.

Use the tool that fits the system you actually have.

GO

THE DEFAULT CHOICE

A standalone compiled collector. No extra runtime on the source. YAML configuration, dry-run and collection statistics.

WHEN A BINARY DOESN'T FIT

Choose an existing runtime. Adapt the collection logic.

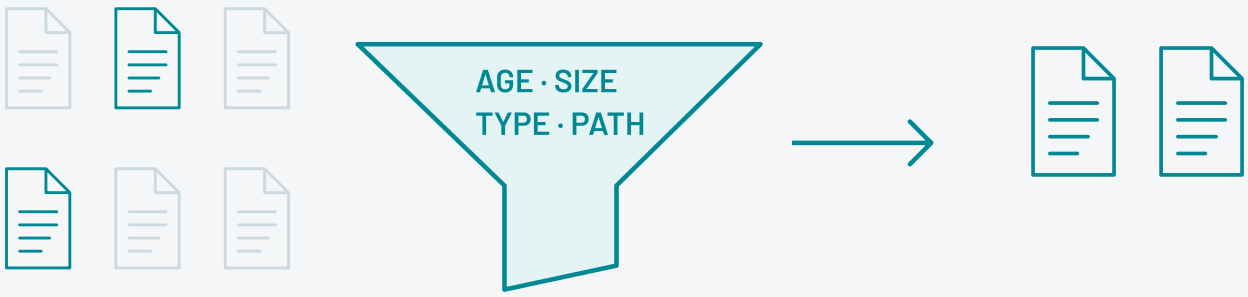
Unix & embedded	`Bash / POSIX sh · ash`
Cross-platform	`Python 3 / Python 2`
Older Unix	`Perl`
Windows	`PowerShell 3+ / PowerShell 2 / Batch`

Open source. Adaptable. Built for file submission.

Match the collector to the target OS, architecture and dependencies.
Features and compatibility vary between implementations.

YOU CHOOSE WHAT LEAVES.

Select the files that matter to your assessment.



SOURCE FILES | YOUR SELECTION | SELECTED COPIES

PREVIEW BEFORE UPLOAD

```
./amd64-linux-thunderstorm-collector \
  --dry-run --debug \
  --path /srv/www \
  --max-age 3d --max-filesize 10
```

Example: Linux Go collector. The configured file-type filters also apply.

CONTROL THE LOAD

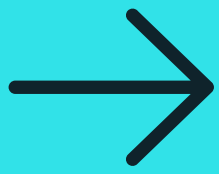
Limit upload concurrency and rate. Use HTTPS with verified certificates.

REPEAT ON YOUR SCHEDULE

Use existing automation to collect recent files. Scheduled checks are not real-time monitoring.

START WITH THE SYSTEM YOU CAN'T SCAN.

Choose one blind spot.
Bring its files to THOR.



01 DEPLOY THE SERVICE

Run a licensed THOR Thunderstorm instance on your own infrastructure.

02 CONNECT A COLLECTOR

Start with a scoped dry-run, then submit a small file set.

03 FOLLOW THE FINDINGS

Verify source attribution and results, then expand the workflow.

EXPLORE THOR THUNDERSTORM

nextron-systems.com/thor-thunderstorm/

Collectors: github.com/NextronSystems/thunderstorm-collector