THOR Thunderstorm

Architecture, Feature Overview and Use Cases

What is THOR?

A portable forensic scanner with a huge signature database optimised for compromise assessments and the detection of hacking activity

What is THOR Thunderstorm?

**A RESTful web service
that receives samples and
returns a result**

THOR Scan Service – The Idea

- Start THOR as a service (new mode)
- Simply submit samples from anywhere within the network
  - Malware analysis pipeline
  - Sandboxes (mail attachments, network traffic extracts)
  - Sample collections of your EDR
- Get results with
  - score (benign, suspicious, malicious)
  - meta data (reference link, tags including MITRE ATT&CK tags etc.)
- Simple Setup (Linux, Windows)
- High Performance (multi-threaded scanning), analyze thousands of samples per minute
- Special file type support
- Deployable in cloud containers (AWS, Docker ..)
- Users can include their own custom YARA rules

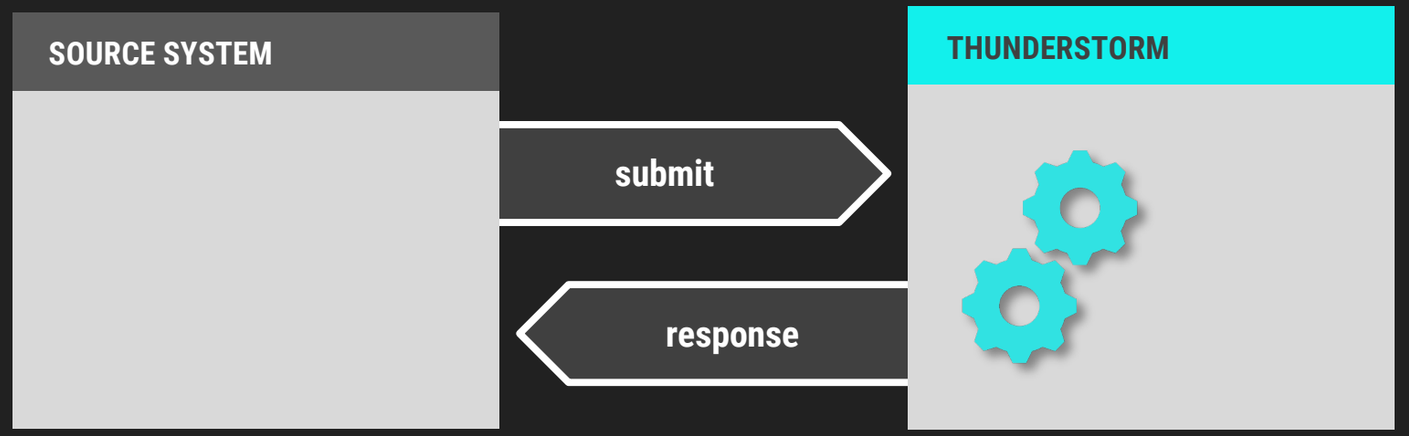
Extraordinary Features

- Includes THOR's 10,000+ hand-crafted YARA rules with focus on
  - APT related malware
  - Hack tools
  - Forensic artefacts
  - Obfuscation techniques
  - Web shells
- Special file types supported
  - Registry Hives (full walk and IOC application)
  - Memory Dumps (full YARA scan)
  - EVTX Eventlogs (log parsing and IOC application)
  - WER files (Error report analysis)

Two Modes of Submission

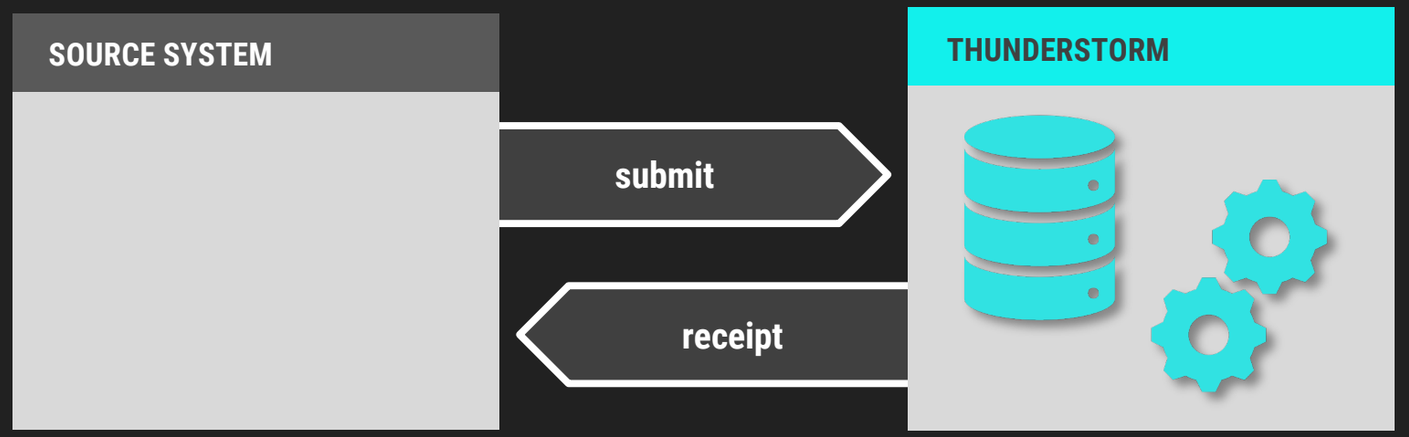
Synchronous

- Response contains result
- Number of concurrent analyses is limited to number of threads
- Service overload results in HTTP 403 Retry in X seconds errors
- Used in low volume analysis setups



Asynchronous

- Response contains receipt with id
- Number of concurrent submissions is limited to number of threads
- Sample analysis from queue
- Service overload is less likely
- Used in low volume analysis setups



Overview – Analysis Setup

ANALYSIS SYSTEM

Sample Submission

```
POST /api/scan/
Host: scan.thor.int

... File Payload
```

submits

Server Response

```
{
  “level”: “Alert”,
  “matched_rules”: [
    {
       “rule_name”: “HKTL_PwdDump_1”,
       “score”: 85
       “reference”: “https://blog.fu.bar”,
       ...
```

responds

SERVER SYSTEM

THUNDERSTORM

- 13,000+ YARA Rules
- Special Format Support: EVTX, Registry Hive, Memory Dumps etc.
- High Performance Sample Processing
- Sample Queuing

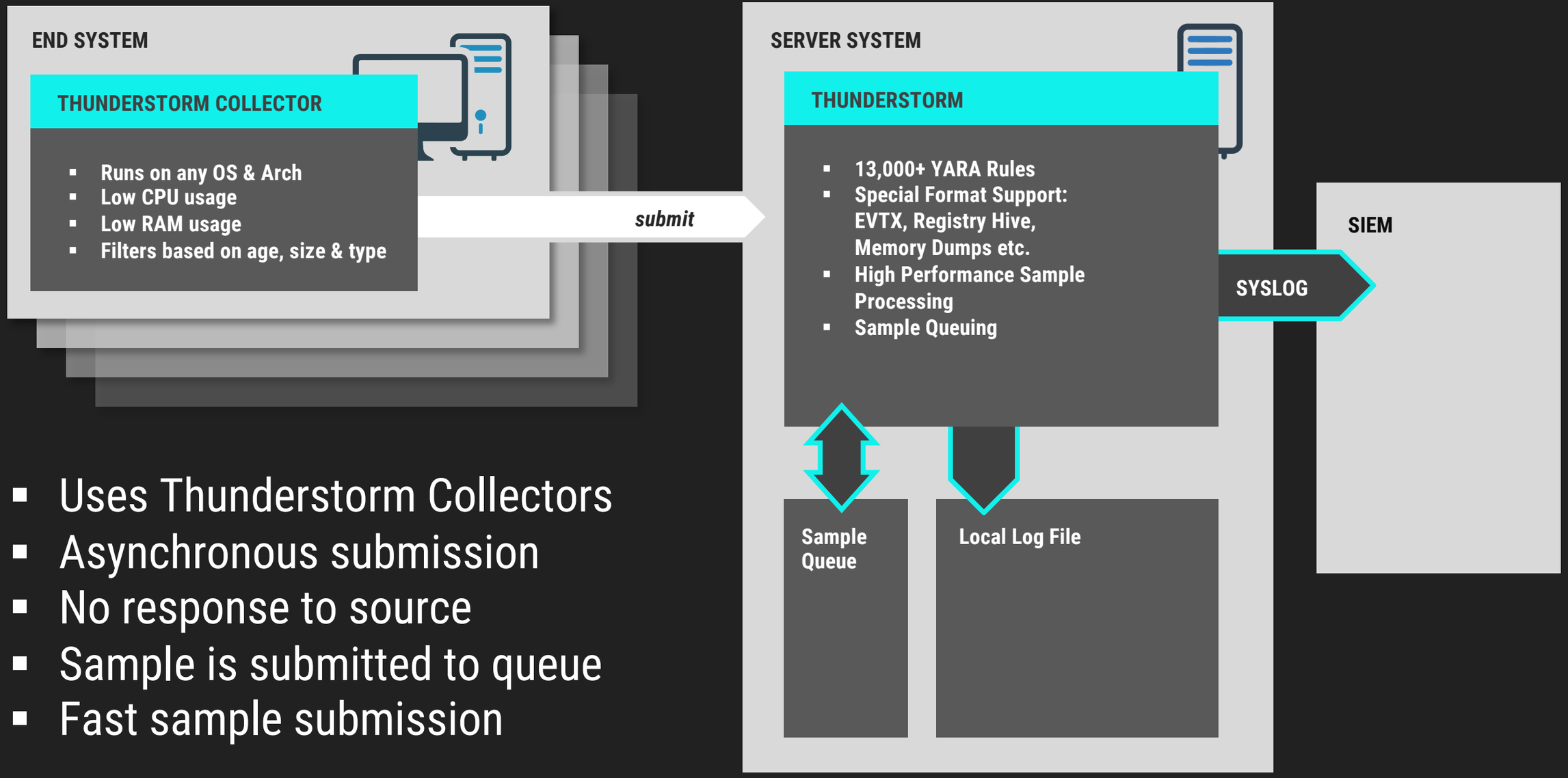
SYSLOG

SIEM

LOCAL LOG FILE
- Text (SYSLOG format)
- JSON

- Uses Thunderstorm API
- Synchronous submission
- Response contains result
- Processing slots equals number of threads
- No queuing (overload: HTTP 403 Retry in X seconds)

Overview – Mass Collection Setup

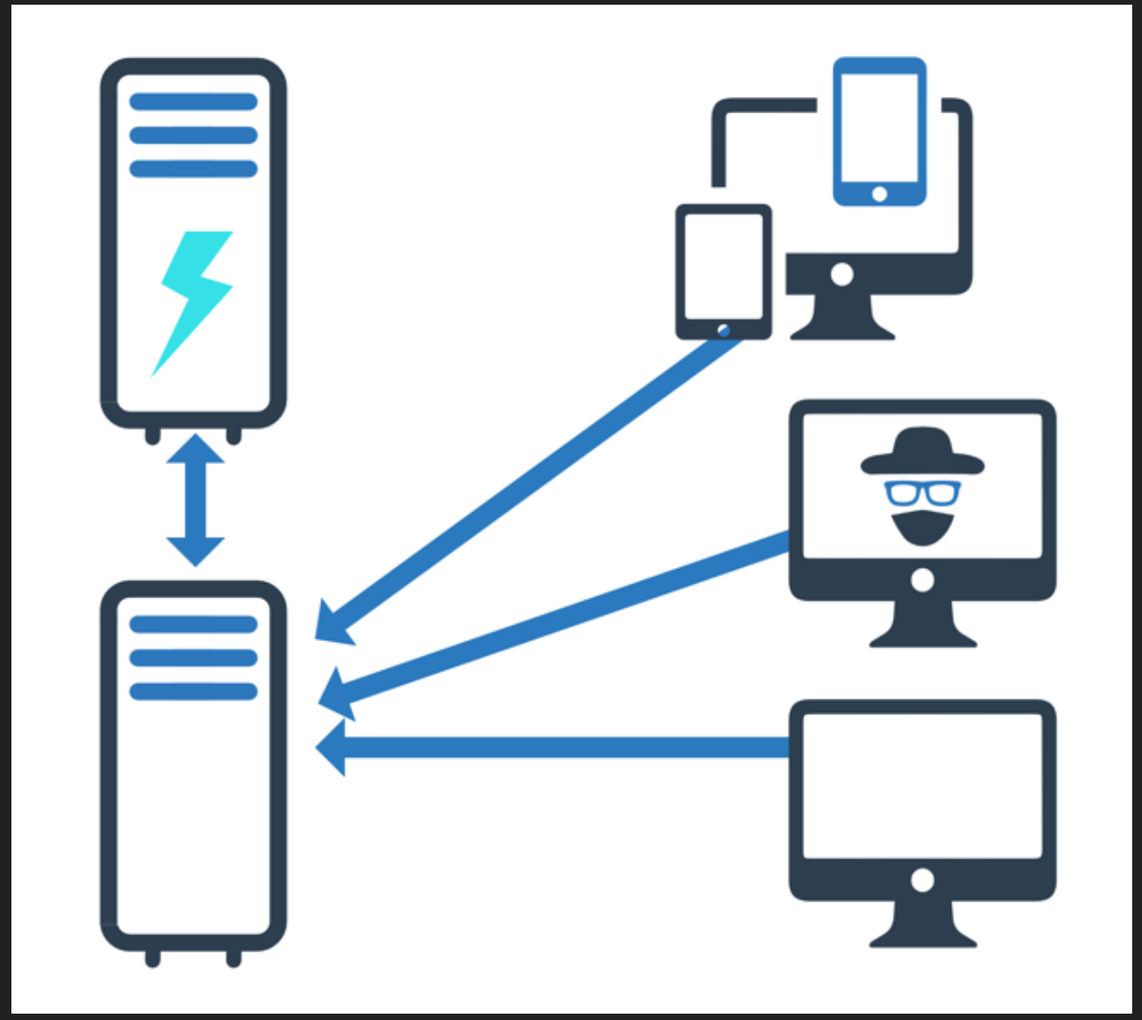


- Uses Thunderstorm Collectors
- Asynchronous submission
- No response to source
- Sample is submitted to queue
- Fast sample submission

Thunderstorm - Use Case 1

Integration into Sample Processing Pipeline

- Remote File Collection
- Use THOR Thunderstorm as
  - Fast processing of large amounts of files
  - Pre-selector for deeper analysis
  - 2nd or 3rd engine in your setup (this engine has focus on forensic artefacts and APTs)



Thunderstorm - Use Case 2

ICS Networks

- Highly critical environments
- Stability / Availability has highest priority
- Endpoint agents are frowned upon (consume memory, threaten system stability, warranty claims)

- Collect samples with simple scripts or already available tools
- Analyze thousands of them per minute with a single THOR Thunderstorm server

Thunderstorm - Use Case 3

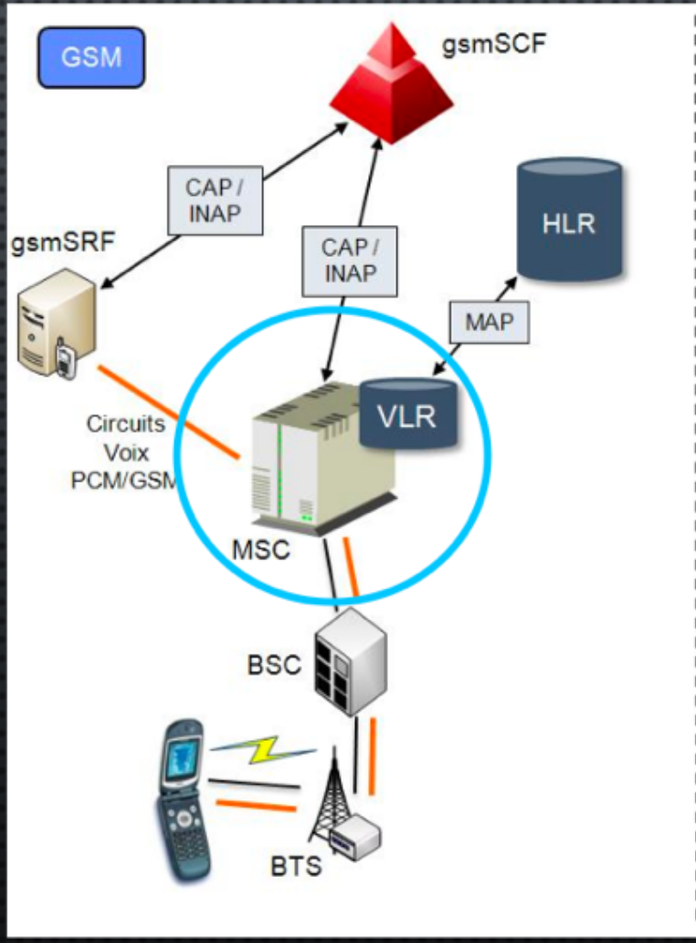
Out of Reach Devices

- Network devices
- PBX
- Embedded devices
- IOT

So far: “We see them being used in APTs but cannot scan them!”

TELCO BASICS

- GSM/3G/4G SEPARATE SPEECH FROM SIGNALLING
- VOICE CALLS INTERCEPTION DONE THROUGH LIG
- INTERCEPTION FROM THE NETWORK IS DIFFICULT
- LIG CONTROLLED THROUGH AUTHORITIES (WARRANTS, COURT ORDERS)
- VOICE CALLS ARE CONTROLLED BY THE MSC/MSS/MGW
- SUBSCRIBERS DATA IS STORED IN THE HLR



Thunderstorm - Use Case 4

Out of Reach Operating Systems

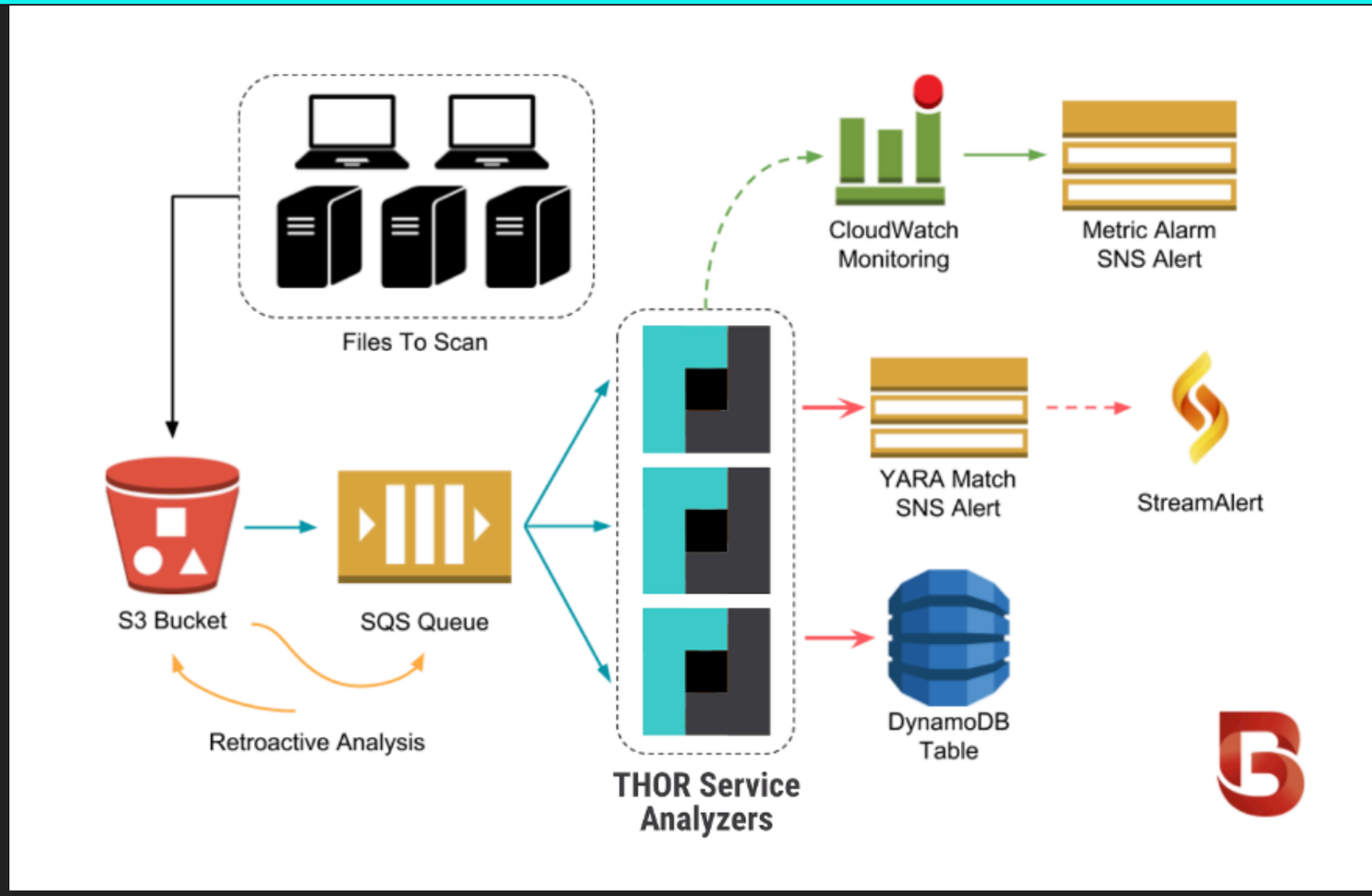
- File collection scripts for many old or usually unsupported operating systems
- Upload samples for analysis
- Select files based on size, age or type
- Schedule frequent upload tasks to analyse only new or modified files

Thunderstorm - Use Case 5

S3 Bucket Scanning

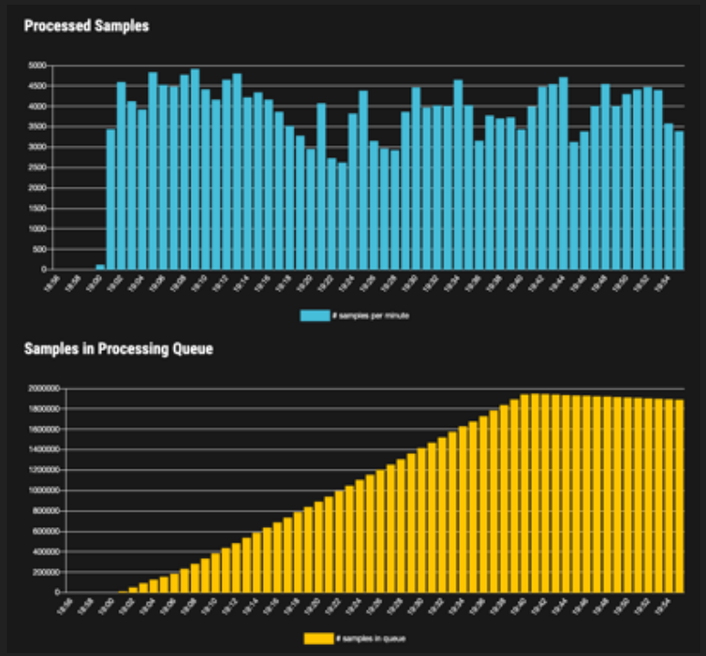
- Process millions of samples per minute with the scalability of the cloud
- Apply 13,000+ hand-crafted high quality YARA rules

First Proof-of-Concept with BinaryAlert by AirBnb in which we' replaced the standard YARA analyzer with THOR Thunderstorm service



Thunderstorm – Web Interface

- Web GUI with status information and statistics
- Helps you to monitor load, processing queues and performance indicators
- Includes links to API documentation, API client and Thunderstorm Collectors



Server Load Statistics
- Processed Samples
- Samples in Processing Queue

THOR THUNDERSTORM

4458 scanned samples at an average scan speed of 5063 samples per minute

License Expiration Date	2021/01/30
Pure Yara	true
Sigma Version	0.18.1-50-gaf3b93a5
Signature Version	2020/09/07-152238
Sync Request Only Threads	3
Thor Timestamp	2020-09-10 11:35:00
Thor Version	10.6.0
Threads	40
Yara Version	4.0.2
Avg Scan Time Milliseconds	473
Avg Total Time Milliseconds	474
Denied Request Proportion	0.000000
Denied Requests	0
Queued Async Requests	6436
Quota Wait Time Milliseconds	0
Quota Waits	0
Scanned Samples	4458
Uptime Seconds	59

API Documentation

API Client Library

Live Demo: Web GUI

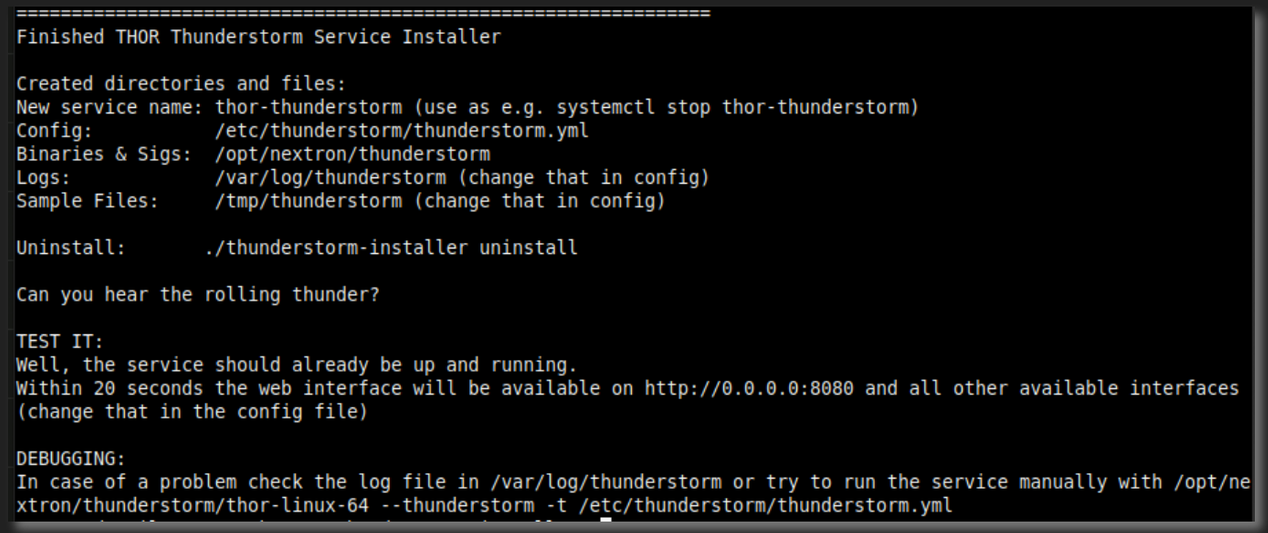
THOR Thunderstorm WebInterface

Thunderstorm - Components

- Thunderstorm Server
  - Is THOR run with “--thunderstorm” flag
  - Documentation: THOR Manual
  - Installation (Linux): via Installer Script
- Thunderstorm Collector
  - Go-based
  - Hosted on Github: https://github.com/NextronSystems/thunderstorm-collector
  - Documentation: Github Readme
  - Download from Release section
- ThunderstormAPI Client
  - Python-based
  - Hosted on Github: https://github.com/NextronSystems/thunderstormAPI
  - Documentation: Github Readme
  - Installation: pip install thunderstormAPI
- Thunderstorm Helper Scripts
  - Installer, Updater
  - Hosted on Github: https://github.com/NextronSystems/nextron-helper-scripts/thunderstorm

Thunderstorm Installer Script

- Shell Script for Linux
- Retrieves THOR, prepares directories, default configuration, registers a service, starts the service
- Uninstall function included
- Many command line hints
- Requires: Bash, Wget
- Simple Installation
  1. Download license
  2. Run thunderstorm-installler.sh



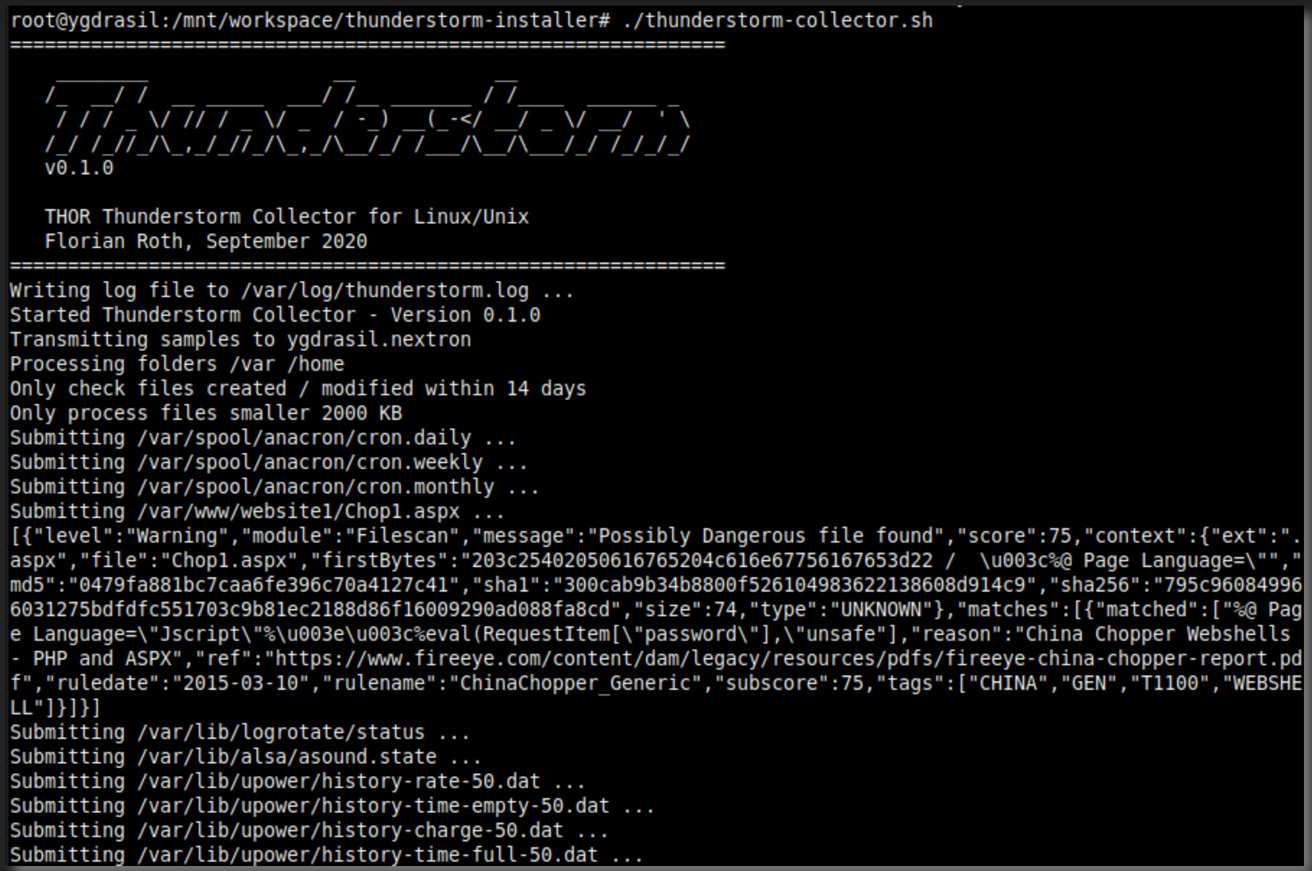
On Github https://github.com/NextronSystems/nextron-helper-scripts/blob/master/thunderstorm/thunderstorm-installer.sh

Live Demo: Installer

THOR Thunderstorm Installation on Linux

Thunderstorm Collector Scripts

- Script for Linux/Unix and Windows
- Collects and submits files to Thunderstorm Server
- Requires: Curl executable
- Allows to select files based on age, size, and extension
- Use Case: Add this script to crontab to submit every hour all newly created files in your webserver root to THOR Thunderstorm for analysis


root@ygdrasil:/mnt/workspace/thunderstorm-installer# ./thunderstorm-collector.sh
v0.1.0
THOR Thunderstorm Collector for Linux/Unix
Florian Roth, September 2020
Writing log file to /var/log/thunderstorm.log ...
Started Thunderstorm Collector - Version 0.1.0
Transmitting samples to ygdrasil.nextron
Processing folders /var /home
Only check files created / modified within 14 days
Only process files smaller 2000 KB
Submitting /var/spool/anacron/cron.daily ...
Submitting /var/spool/anacron/cron.weekly ...
Submitting /var/spool/anacron/cron.monthly ...
Submitting /var/www/website1/Chop1.aspx ...
[{"level":"Warning","module":"Filescan","message":"Possibly Dangerous file found","score":75,"context":{"ext":".aspx","file":"Chop1.aspx","firstBytes":"203c25402050616765204c616e67756167653d22 / \u003c%@ Page Language=\"","md5":"0479fa881bc7caa6fe396c70a4127c41","sha1":"300cab9b34b8800f526104983622138608d914c9","sha256":"795c960849966031275bdfdfc551703c9b81ec2188d86f16009290ad088fa8cd","size":74,"type":"UNKNOWN"},"matches":[{"matched":["%@ Page Language=\"Jscript\"%\u003e\u003c%eval(RequestItem[\"password\"],\"unsafe"],"reason":"China Chopper Webshells - PHP and ASPX","ref":"https://www.fireeye.com/content/dam/legacy/resources/pdfs/fireeye-china-chopper-report.pdf","ruledate":"2015-03-10","rulename":"ChinaChopper_Generic","subscore":75,"tags":["CHINA","GEN","T1100","WEBSHELL"]}]}]
Submitting /var/lib/logrotate/status ...
Submitting /var/lib/alsa/asound.state ...
Submitting /var/lib/upower/history-rate-50.dat ...
Submitting /var/lib/upower/history-time-empty-50.dat ...
Submitting /var/lib/upower/history-charge-50.dat ...
Submitting /var/lib/upower/history-time-full-50.dat ...

On Github https://github.com/NextronSystems/thunderstorm-collector/tree/master/scripts

Thunderstorm Collector

- Compiled binaries for any well-known architecture and operating system
- OS: Linux, Windows, Android, FreeBSD, OpenBSD, AIX …
- Arch: 386, amd64, arm, …
- Support for outdated operating systems like Windows 2003 Server and Windows XP
- Low CPU usage (0.5-2%)
- Low RAM usage (20 Mbyte)

On Github https://github.com/NextronSystems/thunderstorm-collector

Live Demo: Collector

THOR Thunderstorm Sample Submission on Linux

THOR Thunderstorm Licensing

- New “Service” License Type
- Two variants
  1. Unlimited
  2. Limited to samples per minute (cheaper)

THOR Thunderstorm Roadmap

- Available in THOR TechPreview: September 2020
- ICAP Support
- Thunderstorm Installer and Collector as PowerShell Scripts for Windows platform
- Docker image
- Support for other file formats
  - PCAP files
  - MFT files
- Caching proxy service

Get Started

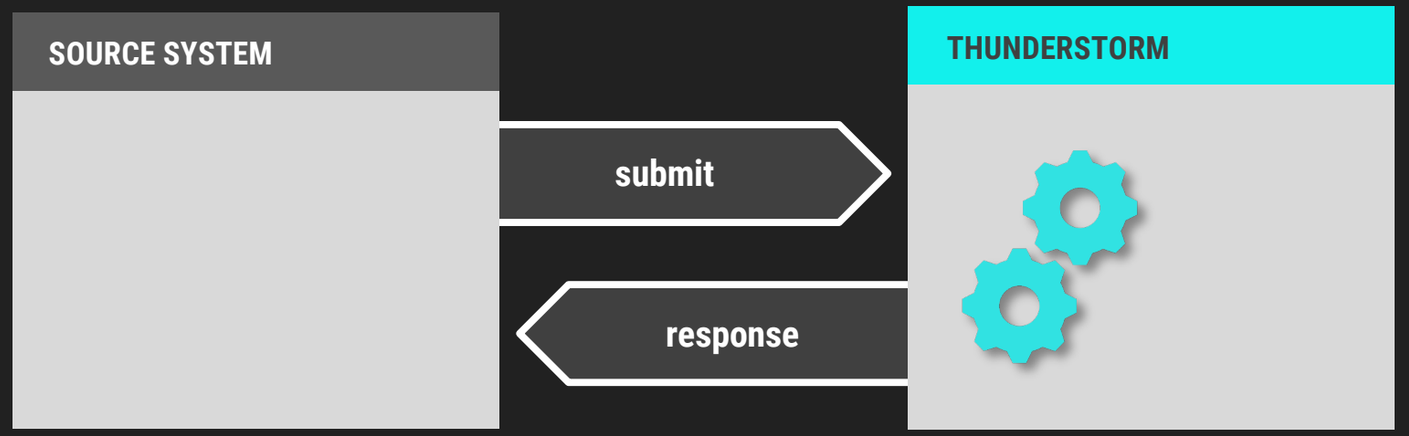
Visit the contact form an mention “THOR Service”
https://www.nextron-systems.com/get-started/

Extra Slides

Two Modes of Submission

Synchronous

- Response contains result
- Number of concurrent analyses is limited to number of threads
- Service overload results in HTTP 403 Retry in X seconds errors
- Used in low volume analysis setups



Asynchronous

- Response contains receipt with id
- Number of concurrent submissions is limited to number of threads
- Sample analysis from queue
- Service overload is less likely
- Used in low volume analysis setups