Aurora Agent User Manual

Nextron Systems GmbH

Apr 02, 2024

CONTENTS:

CHAPTER

ONE

WHAT IS AURORA?

- Aurora is a lightweight endpoint agent that applies Sigma rules and IOCs on local event streams
- It uses Event Tracing for Windows (ETW) to subscribe to certain event channels
- It extends the Sigma standard with so-called "response actions" that trigger after a rule match
- It writes its own events to various outputs: the Windows Eventlog, a log file and remote UDP/TCP targets

CHAPTER

TWO

WHAT IS AURORA LITE?

Aurora Lite is our free version of Aurora which is free for private and commercial use. The only limitation defined in the TOS is that it cannot be sold or used as part of a paid service. We offer special licensing options for managed detection service providers.

Features and services that are not included in the Aurora Lite version:

- No comfortable Sigma rule management via ASGARD Management Center
- No additional detection modules (non-Sigma-based detection; e.g. Cobalt Strike beaconing, LSASS dumping)
- No private Nextron Sigma rule feed
- No private Nextron IOC rule feed
- No encrypted Sigma rules (protect rules from spying eyes or the AV)
- Only 5 rules with response actions allowed

For more details see the description on https://www.nextron-systems.com/aurora

CHAPTER

THREE

INSTALLATION

3.1 Requirements

Aurora runs on Windows 7 or newer and requires administrative privileges.

Other OS (Linux or macOS) are not supported.

3.1.1 Supported

- Windows 7 x86 / x64
- Windows Server 2008 R2 x64
- Windows 8.1
- Windows Server 2012
- Windows Server 2012 R2
- Windows 10
- Windows 11
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

3.1.2 Update Servers

To download the newest updates for Aurora and our signatures, you need an active internet connection. The endpoint performing the update needs to reach our update servers to do this.

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

Hint: You do not need an active internet connection to run Aurora on an endpoint. This is only needed if you want to update to the latest Aurora or signature versions.

3.1.3 Define an Antivirus / EDR Exclusion

It is recommended to exclude Aurora from your Antivirus / EDR solution.

Depending on your architecture and whether Aurora was installed or started interactively from a temporary directory, the exclusion paths are:

1. For an installed Aurora

```
C:\Program Files\Aurora-Agent\aurora-agent-64.exe
C:\Program Files\Aurora-Agent\aurora-agent.exe
```

2. For a interactively started Aurora the path you have used for extraction. For example:

```
C:\aurora\aurora-agent-64.exe
C:\aurora\aurora-agent.exe
```

3.2 Quick Installation

1. Extract the program package into a temporary folder (e.g. `C:\aurora`)
2. Make sure to place the license file (`*.lic`) into the extracted folder
3. Start a `cmd.exe` as administrator
4. Change directory to the extracted folder (`cd C:\aurora`)
5. Run one of the following commands (with/without GUI)

```
C:\aurora>aurora-agent.exe --install
C:\aurora>aurora-agent.exe --install --dashboard
```

6. Verify new events in the local "Application" event log (Event Viewer) or the Aurora Dashboard
7. Run the following commands to get details on the current status of the agent

```
C:\aurora>aurora-agent.exe --status
C:\aurora>aurora-agent.exe --status --trace
```

See the *Function Tests* section for ideas on how to test Aurora is working as expected.

3.3 Manual installation

3.3.1 Install Aurora

You can install the agent using the following command line from command line terminal that has been started "As Administrator".

```
C:\aurora>aurora-agent.exe --install
```

After the installation the agent, configuration files and rules reside in `C:\Program Files\Aurora Agent\`.

It automatically copies all rule files located in the sub-folders `signatures\sigma-rules` and `custom-signatures`. The `signatures\sigma-rules` folder contains the current open source rule set maintained in the Sigma repository. The `custom-signatures` folder can be used to add your own sigma rules.

Aurora comes with with 4 configuration presets that we encourage you to explore and use:

- Standard (`agent-config-standard.yml`)
- Reduced (`agent-config-reduced.yml`)
- Minimal (`agent-config-minimal.yml`)
- Intense (`agent-config-intense.yml`)

The different presets are explained in more detail in the chapter *Configuration*.

An installation that uses the preset named "reduced" would look like this:

```
C:\aurora>aurora-agent.exe --install -c agent-config-reduced.yml
```

3.3.2 Custom Settings

Adding your own Sigma rules or IOCs is described in chapter *Custom Signatures and IOCs*. The preferred way is to add them to the `custom-signatures` folder before you install Aurora.

All the flags that you use after `--install` get written to the configuration file named `agent-config.yml` in the `C:\Program Files\Aurora Agent\` folder and will be used by the service.

A typical command to install Aurora would look like this

```
C:\aurora>aurora-agent.exe --install --activate-responses
```

3.3.3 Uninstall Aurora

To uninstall the agent simply run the following command:

```
C:\Program Files\Aurora-Agent>aurora-agent.exe --uninstall
```

If the uninstaller fails due to unknown errors, you can uninstall Aurora manually with these commands (Run from an administrative shell)

```
C:\Users\nextron>sc stop aurora-agent
C:\Users\nextron>sc delete aurora-agent
C:\Users\nextron>rmdir /s /q "C:\Program Files\Aurora-Agent"
C:\Users\nextron>schtasks /Delete /F /TN aurora-agent-program-update
C:\Users\nextron>schtasks /Delete /F /TN aurora-agent-signature-update
```

3.4 Installation using ASGARD

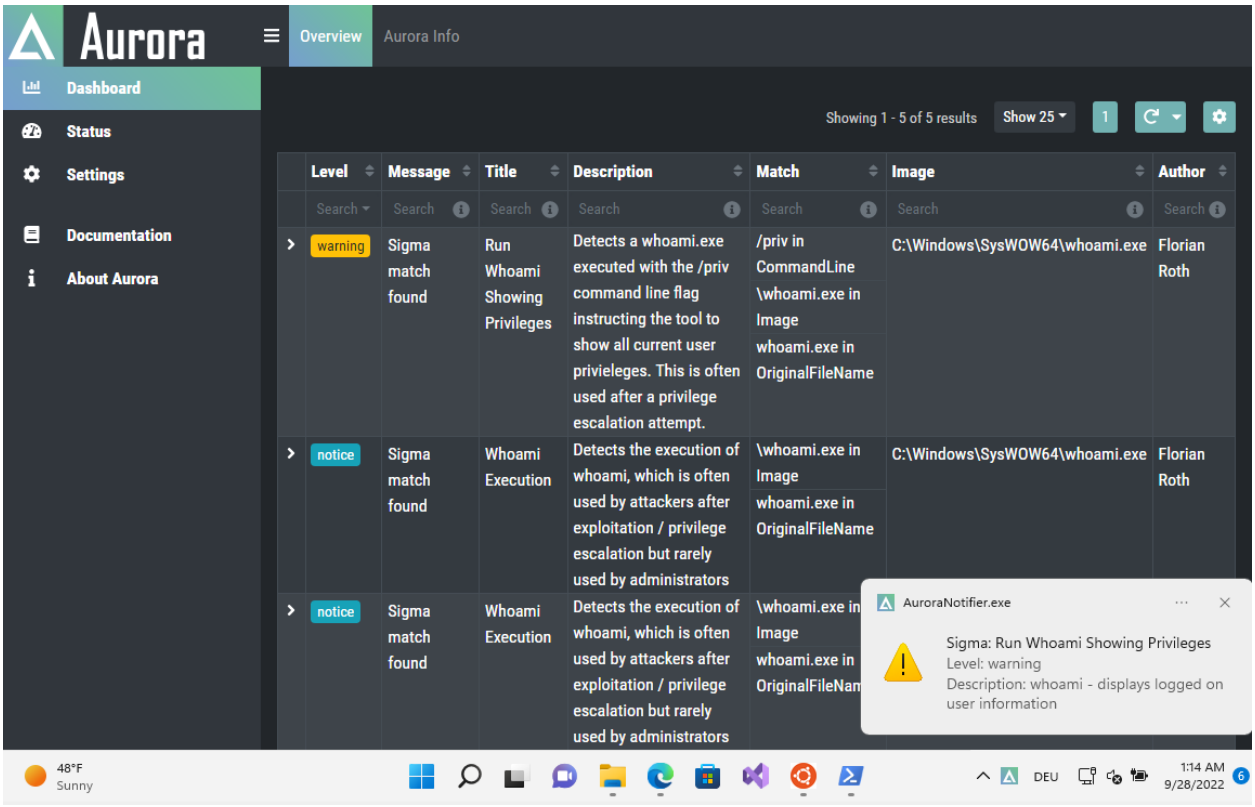
When using ASGARD Management Center, Aurora can be installed using the `Service Control` tab; see the relevant chapter in the ASGARD manual for details.

CHAPTER

FOUR

AURORA AGENT DASHBOARD

Aurora Agent Dashboard provides a way to review Aurora events and get notifications for them:



4.1 Dashboard Activation

Aurora can be started or installed with the dashboard feature using the `--dashboard` flag:

Examples:

```
C:\aurora>aurora-agent-64.exe --install --dashboard
C:\aurora>aurora-agent-64.exe --dashboard
```

4.2 Dashboard UI Access

The Dashboard can be accessed with your favorite browser using the following URL: `http://localhost:17494/ui/dashboard/overview`

There are also some shortcuts to open the dashboard such as:

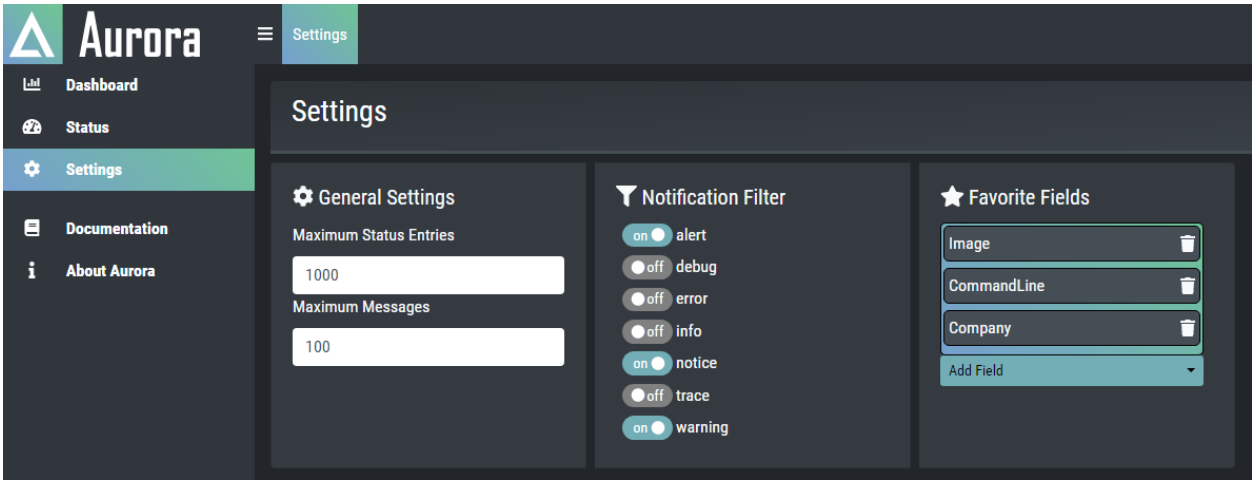
- Click on a notification message
- Right click the tray icon and then select `Open Dashboard`

4.3 Notifications

The filter for which events will produce notifications can be configured either in the tray icon in the `Options` sub menu or in the dashboard's `Settings`-section.

4.4 Settings

The settings can be accessed in the `Settings` Section in the dashboard.



4.4.1 General Settings

- Maximum Status Entries
  - Sets the Maximum Status Entries that will be saved in Memory
- Maximum Messages
  - Sets the Maximum Count of Aurora Event Messages that will be saved in Memory

4.4.2 Favorite Fields

You can set favorite fields in the `Settings` section. These fields are shown in the configured order in every event you expand. You can change the order by dragging and dropping the fields in the `Favorite Fields` list.

CHAPTER

FIVE

USAGE

This chapter explains some more frequently used command line options used by Aurora in more detail.

5.1 Run Aurora

If you simply run Aurora in your terminal, it'll use the default values for each flag and no dedicated config file:

```
C:\aurora>aurora-agent-64.exe
```

You can select one of the default config presets with the respective flag:

```
C:\aurora>aurora-agent-64.exe -c agent-config-reduced.yml
```

A typical command line that runs Aurora and prints messages and matches to the command line and the Windows `Application` eventlog looks like this:

```
C:\aurora>aurora-agent-64.exe --minimum-level low
```

5.2 Run Aurora as Service

To install Aurora as a service, use the `--install` flag and see the chapter *Installation* for more details.

A typical installation on systems that have limited hardware resources could look like this.

```
C:\aurora>aurora-agent-64.exe --install -c agent-config-reduced.yml
```

We ship Aurora with 4 presets that we recommend to use. See the chapter *Configuration* for more information.

5.3 Aurora Service Status Information

The `--status` flag can be used to query status information from the running service.

This flag can be combined with the `--json` and `--trace` flags for JSON formatted or more detailed output.

Note: If you've set a non-standard name when starting Aurora (using `--agent-name`), make sure to pass the same value here as well with `--agent-name`.

```
C:\aurora>aurora-agent-64.exe --status
Aurora Agent
Version: 0.9.1
Build Revision: 37fec81332531
Signature Revision: 2022/03/21-101412
Sigma Revision: 0.20-3331-gb4245c561
Status: running
Uptime (in hours): 0

Active Outputs:
    Windows Application Eventlog: enabled
    Stdout: enabled

Active Modules: LsassDumpDetector, BeaconHunter, EtwCanary, CommandLineMismatchDetector,␣
↪ProcessTamperingDetector, TemporaryDriverLoadDetector, ApplyIOCs, Rescontrol, Sigma,␣
↪ETWSource, ETWKernelSource, EventlogSource, PollHandles

Rule Statistics:
    Rule paths: C:\aurora\signatures\sigma-rules, C:\aurora\custom-signatures
    Loaded rules: 1285
    Rule reloads: 0
    Responses: 28

False positive filters: 4
Process excludes: 0

Events missed so far: 0
Sigma matches: 8
Suppressed Sigma matches of those: 0

Response Actions: disabled
```

This flag can be combined with the `--json` or `--trace` flags:

- JSON output is significantly more comprehensive, but is also more prone to changes (especially additions).
- Trace output contains more details, for example full event statistics.

5.4 Tracing Events

Using the `--trace` flag you can view all the events Aurora observes in the different subscribed channels.

It's a good idea to write the output to a file in order to search in it later.

```
C:\aurora>aurora-agent-64.exe --trace > d:\aurora-trace.log
```

CHAPTER

SIX

CONFIGURATION

Aurora uses the YAML format for its configuration file. All values set in the config file can also be used as command line flags.

There are two modes of operation:

1. Aurora started directly from the command line, optionally using a config passed with the `--config` / `-c` flag
2. Aurora started as a service (see chapter *Installation*) for more details) with a config file located in `C:\Program Files\Aurora-Agent\agent-config.yml`

6.1 Configuration Presets

To facilitate the use of Aurora, four configuration files are part of the Aurora package:

- Standard (`agent-config-standard.yml`)
- Reduced (`agent-config-reduced.yml`)
- Minimal (`agent-config-minimal.yml`)
- Intense (`agent-config-intense.yml`)

An installation that uses the preset named "reduced" would look like this:

```
C:\aurora>aurora-agent.exe --install -c agent-config-reduced.yml
```

The configuration presets effect the following settings:

Affected Setting	Minimal	Reduced	Standard	Intense
Deactivated Sources	Registry, Raw Disk Access, Kernel Handles, Create Remote Thread, Process Access, Image Loads	Registry, Raw Disk Access, Kernel Handles, Create Remote Thread, Process Access	Registry, Raw Disk Access, Kernel Handles, Create Remote Thread	
CPU Limit	20 %	30 %	35 %	100 %
Process Priority	Low	Normal	Normal	Normal
Minimum Reporting Level	High	High	Medium	Low
Deactivated Modules	LSASS Dump Detector, BeaconHunter	LSASS Dump Detector		

Warning: Intense preset uses the most system resources and can put the system under heavy load, especially if a process accesses many registry keys in a short amount of time.

We recommend using this preset only on a very selective set of systems or in cases in which maximum detection is required.

6.1.1 Custom Profiles

If you need a more specialized configuration than these predefined ones, you can also create your own configuration for maximal adaptability.

6.2 Output Options

The following output options are available

- Windows Eventlog (default)
- Log file
- UDP target
- TCP target

Output is usually formatted in a human readable way (with `KEY: value` pairs). For machine ingestion, using `--json` is recommended, which changes the format to JSON structs.

6.2.1 ASGARD Analysis Cockpit

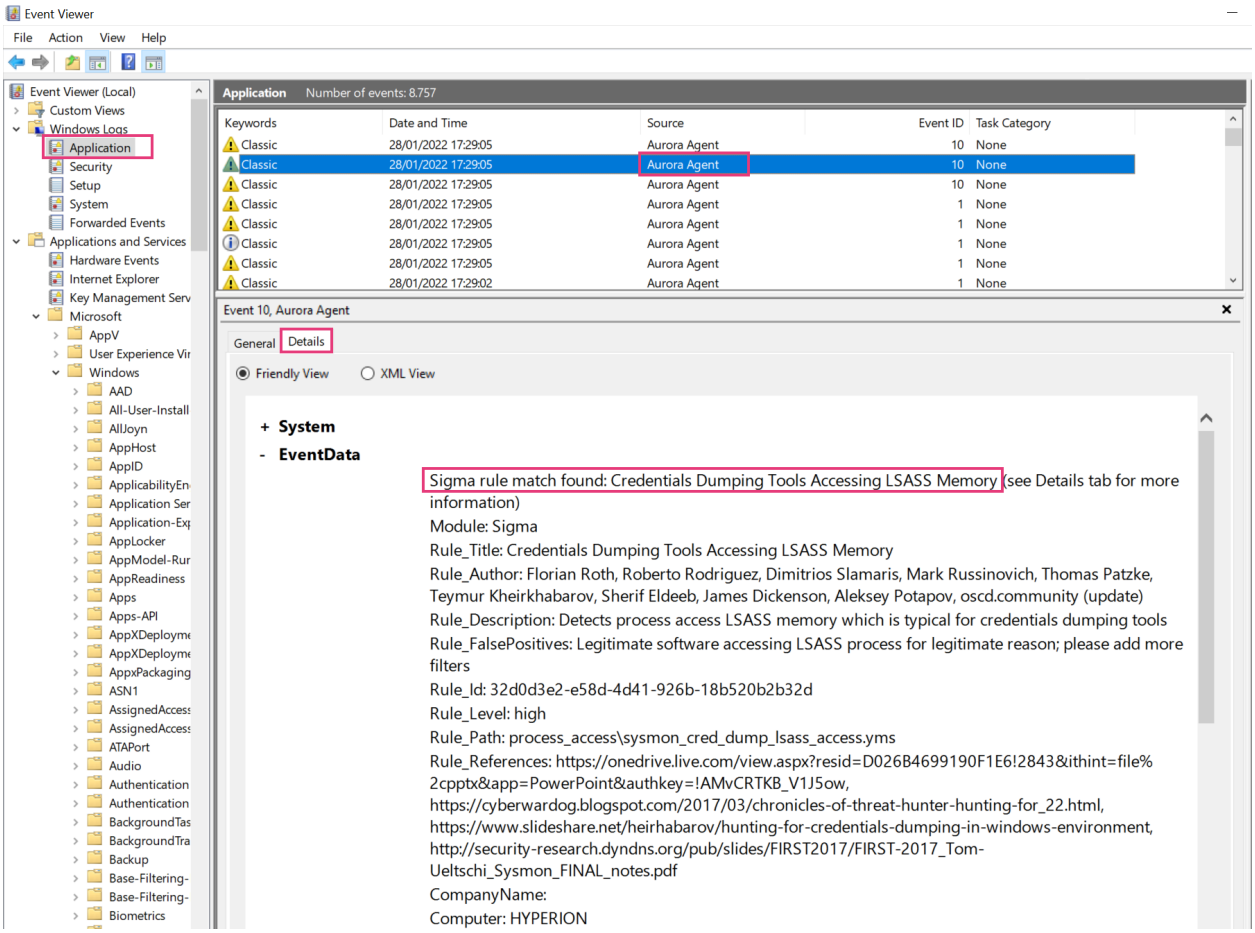
Whenever you install an ASGARD Agent, the controlled Aurora Agent Services gets its configuration automatically. In a default setup, all logs generated by an Aurora Agent will be relayed via an ASGARD to an Analysis Cockpit system.

6.2.2 Windows Eventlog

By default Aurora writes its event into the Windows event log "Application". To review the events use the Windows "EventViewer". Make sure to check the "Details" tab to see all fields and values.

6.2.3 UDP / TCP Targets

UDP or TCP log targets can be specified via the `--udp-target` and `--tcp-target` options. These options take an argument in the form `host:port`, e.g. `myloggingsystem.internal:8443`.


Event Viewer
File Action View Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services
Hardware Events
Internet Explorer
Key Management Serv
Microsoft
AppV
User Experience Vir
Windows
AAD
All-User-Install
AllJoyn
AppHost
AppID
ApplicabilityEn
Application Ser
Application-Exp
AppLocker
AppModel-Rur
AppReadiness
Apps
Apps-API
AppXDeployme
AppXDeployme
AppxPackaging
ASN1
AssignedAccess
AssignedAccess
ATAPort
Audio
Authentication
Authentication
BackgroundTas
BackgroundTra
Backup
Base-Filtering-
Base-Filtering-
Biometrics
Application Number of events: 8,757
Keywords Date and Time Source Event ID Task Category
Classic 28/01/2022 17:29:05 Aurora Agent 10 None
Classic 28/01/2022 17:29:05 Aurora Agent 10 None
Classic 28/01/2022 17:29:05 Aurora Agent 10 None
Classic 28/01/2022 17:29:05 Aurora Agent 1 None
Classic 28/01/2022 17:29:05 Aurora Agent 1 None
Classic 28/01/2022 17:29:05 Aurora Agent 1 None
Classic 28/01/2022 17:29:05 Aurora Agent 1 None
Classic 28/01/2022 17:29:02 Aurora Agent 1 None
Event 10, Aurora Agent
General Details
Friendly View XML View
+ System
- EventData
Sigma rule match found: Credentials Dumping Tools Accessing LSASS Memory (see Details tab for more information)
Module: Sigma
Rule_Title: Credentials Dumping Tools Accessing LSASS Memory
Rule_Author: Florian Roth, Roberto Rodriguez, Dimitrios Slamaris, Mark Russinovich, Thomas Patzke, Teymur Kheirkhabarov, Sherif Eldeeb, James Dickenson, Aleksey Potapov, oscd.community (update)
Rule_Description: Detects process access LSASS memory which is typical for credentials dumping tools
Rule_FalsePositives: Legitimate software accessing LSASS process for legitimate reason; please add more filters
Rule_Id: 32d0d3e2-e58d-4d41-926b-18b520b2b32d
Rule_Level: high
Rule_Path: process_access\sysmon_cred_dump_lsass_access.yms
Rule_References: https://onedrive.live.com/view.aspx?resid=D026B4699190F1E6!2843&ithint=file%2cpptx&app=PowerPoint&authkey=!AMvCRTKB_V1J5ow, https://cyberwardog.blogspot.com/2017/03/chronicles-of-threat-hunter-hunting-for_22.html, https://www.slideshare.net/heirhabarov/hunting-for-credentials-dumping-in-windows-environment, http://security-research.dyndns.org/pub/slides/FIRST2017/FIRST-2017_Tom-Ueltschi_Sysmon_FINAL_notes.pdf
CompanyName:
Computer: HYPERION

6.2.4 Log File

A log file can be specified using `--logfile`. By default, no log file is written.

The log file is automatically rotated by Aurora once more than `--log-size` bytes have been written to it (default is 10MB). `--log-rotate` can be used to specify the number of log rotations that are retained (defaults to 7).

Log rotation can be disabled by setting `--log-size` to 0.

6.3 Process Exclusions

To exclude specific processes from analysis, you can configure Aurora to ignore all events from specific image paths.

In order to do so, the excluded images must be specified (as regular expressions) in a file that is passed to `--process-excludes`. By default, `config\process-excludes.cfg` is used. This file contains further examples on how to specify the excludes.

The Process Exclusions are typically the values in your `PARENTIMAGE` field (Process Creation Event) or `IMAGE` (for all other Events - File Creation, etc.).

Adding the expressions in the file results in

1. Lower CPU load caused by the exclusion of the process
2. No more matches on events generated by the excluded process

The process exclusion file is loaded at startup. If you change the file, you'll need to restart Aurora to apply those changes.

Hint: Please be aware that adding process exclusions can cause malware that uses process hollowing or similar techniques to mask themselves as an excluded process to go unreported.

6.3.1 Exclusion Examples

To get get a full list of high volume event sources, use the following command:

```
aurora-agent-64.exe --status --trace

...
By process:
    866420 events from C:\Program Files (x86)\NoisyService\serv.exe
    66420 events from C:\Windows\System32\svchost.exe
    11369 events from C:\Program Files\Microsoft VS Code\Code.exe
```

You identify the first entry in the list as the top speaker that you'd like to exclude from the observation. Accordingly, the exclusion should look like this:

```
C:\\Program Files \(x86\)\\NoisyService\\serv\.exe
```

The expressions are applied:

- as **contains**, so there is no need to add `.*` at the beginning or the end of it
- case-sensitive

Make sure that escape every character that has a meaning in regular expressions.

6.4 False Positive Filtering

When encountering false positives or known anomalies, besides reporting them, you can also exclude them using a false positive filter file. By default, `config\false-positives.cfg` is used.

The file passed should contain a regular expression per line; any log lines where any of these false positive regexps matches will not be logged.

If you want to exclude all events from a specific process, process exclusions might be a better choice than a false positive filter since they also cancel any analysis on those events; see *Process Exclusions* for more details.

CHAPTER

SEVEN

UPGRADING AND UPDATING AURORA

7.1 Management Aurora using ASGARD

When using ASGARD Management Center, you can update Aurora Agent and its signatures for all end systems from the Management Center. Doing so is described in more detail in this section of the ASGARD Management Center manual.

7.2 Automatic Updates via Scheduled Tasks

The installer creates two scheduled tasks during the installation of the service.

One task is scheduled to update the signatures on a daily basis and after each login. The other task is scheduled to update the Aurora agent itself on a weekly basis. Both tasks can be disabled separately.

Note: The Aurora agent update also updates the signatures. (`upgrade` command includes a signature update)

7.3 Manual Upgrades and Updates

7.3.1 Aurora Agent Program Updates

Aurora can be upgraded using the Aurora Agent Util binary that is distributed as part of the Aurora Agent package. To upgrade Aurora to the latest version, use:

```
C:\Program Files\Aurora-Agent>aurora-agent-util.exe upgrade --restart-service
```

When upgrading Aurora while it is installed, make sure to:

- Use `--restart-service` to automatically stop and start the service (otherwise the service has to be manually stopped and started)
- Upgrade the installed version using the Aurora Agent Util under `C:\Program Files\Aurora-Agent`

7.3.2 Signature Updates

When Aurora Agent is installed, it adds a scheduled task that checks daily for signature updates and automatically restarts the service. Usually, this is sufficient and no manual action is necessary.

To manually update Aurora's built-in signatures, use the `Aurora Agent Util` binary that is distributed as part of the Aurora Agent package:

```
C:\Program Files\Aurora-Agent>aurora-agent-util.exe update
```

You can specify `--auto-reload` when starting or installing Aurora to automatically reload built-in or custom signatures after you have manually updated them (see the *Configuration* chapter for more details).

```
C:\aurora>aurora-agent-64.exe --install --auto-reload
```

If you do not use `--auto-reload`, make sure to restart Aurora for the new signatures to take effect.

If you haven't set `--auto-reload` during installation, use the `--restart-service` flag to stop and start the service.

```
C:\Program Files\Aurora-Agent>aurora-agent-util.exe update --restart-service
```

7.4 Update Servers

Aurora connects to the following Servers to download updates:

- https://update-aurora.nextron-systems.com/
- https://update-lite.nextron-systems.com/

Please ensure that these servers can be reached.

CHAPTER

EIGHT

RESPONSES

Responses are an extension to the Sigma standard used in Aurora agents.

They can be used to perform certain actions on an event match and therefore immediately respond to a Sigma rule match. Responses can help you contain a threat or limit its damage, but they can also lead to serious problems when they're not handled with care.

> **Warning:** Only use in cases in which you are absolutely sure that a rule doesn't create false positives and test your custom actions carefully.

Intended use cases:

- Worm containment
- Ransomware containment
- Hard blocking of certain uses of a tool (otherwise use AppLocker)

8.1 Types of Actions

Aurora agent supports two types of responses:

1. Predefined
2. Custom

The actions can be a list of predefined actions or commands (see examples below).

8.1.1 Predefined Responses

- `suspend` (suspends the target process)
- `kill` (kills the target process)
- `dump` (creates a dump file in folder configured as `dump-path`)

8.1.2 Custom Responses

A custom response must be a command line calling an executable available from `PATH`.

You can use the values of fields in the corresponding log line encased with single percent signs (e.g. `%ProcessId%`). Windows environment variables can be used encased with double percent signs (e.g. `%%ProgramData%%`).

Note: Be aware that the variable values correspond to the environment of the Aurora Agent process that runs as SYSTEM and not an observed user process.

8.2 Response flags

Responses can be modified by a set of flags that are specified within the YAML as key/value structures. The following response flags exist:

8.2.1 Simulate

`simulate` specifies that a response will not be triggered on a match. Instead, a log entry will be created that notes which response would be triggered. This is the same behavior as when `--activate-responses` is not set.

`simulate` is supported for all responses.

8.2.2 Recursive

`recursive` specifies that a response will also affect all child and descendant processes.

It is supported for the predefined responses and is `true` by default.

8.2.3 Low privilege only

`lowprivonly` specifies that the response will only be triggered if the target process does not run as `LOCAL SYSTEM` or is similarly elevated.

It is supported for the predefined responses and is `true` by default.

8.2.4 Ancestor

`ancestors` specifies that a response will affect a process's ancestor instead of the process itself. `ancestors: 1` causes the response to affect the process's parent instead, `ancestors: 2` causes it to affect the process's grandparent, and so on.

As a special case, `ancestors: all` can be used to affect all ancestors up to the first invalid ancestor (see the `lowprivonly` flag).

`ancestors` is supported for the predefined responses. It is 0 by default.

8.2.5 Process ID field

`processidfield` specifies the field that contains the process ID that shall be affected.

It is `ProcessId` by default and is supported for the predefined responses.

8.3 Response Examples

```
response:
   type: predefined
   action: kill
```

Kill the parent process from a process creation event:

```
response:
   type: predefined
   action: kill
   processidfield: ParentProcessId
```

Kill the process, the parent and the grandparent:

```
response:
   type: predefined
   action: kill
   ancestors: 2
```

```
response:
   type: predefined
   action: suspend
```

Copy the executed image to a backup folder, then kill the target process:

```
response:
   - type: custom
     action: cmd /c copy %Image% "%%ProgramData%%\Aurora\Image-%ProcessId%.bin"
   - type: predefined
     action: kill
```

Simulate a process kill:

```
response:
   type: predefined
   action: kill
   simulate: true
```

8.4 Specifying a Response for a Sigma rule

Responses can be specified for a Sigma rule in two ways. Both have different advantages and disadvantages.

8.4.1 Inline responses

A response can be declared inline in the sigma rule.

This is useful for testing and provides response and sigma rule in a single file.

However, it is also inflexible since all targets where the sigma rules are deployed will have the same responses active. Also, there is no easy way to list all active responses.

```
title: Example rule with inline response
logsource:
   product: windows
   category: process_creation
detection:
   selection:
      Image|endswith: '\example.exe'
   condition: selection
response:
   type: predefined
   action: kill
```

8.4.2 Response sets

Responses can be declared in a separate *response set* file. This file contains a response in combination with a list of rule IDs that identify the rules where the response should be applied.

Response set files can be passed at startup using the `--response-set` option. Multiple response set files can be passed.

If a response is defined in multiple ways for the same rule (e.g. inline and in multiple response sets), the response from the response set that was specified last is used.

```
description: My example response set
response:
   type: predefined
   action: kill
   lowprivonly: true
   ancestors: all
rule-ids:
   - '87df9ee1-5416-453a-8a08-e8d4a51e9ce1'  # Delete Volume Shadow Copies Via WMI
   - 'ae9c6a7c-9521-42a6-915e-5aaa8689d529'  # CobaltStrike Load by Rundll32
```

8.5 Action Results

The results of the actions are logged as part of a log message that lists the executed action and the rule that triggered it. This log message is written into the respective output channels.

CHAPTER

NINE

MODULES

This chapter lists all modules included in the full version of Aurora.

9.1 Providers

Providers are modules that do not detect anything on their own. Instead, they provide events to consumers. Each consumer requests a list of *log sources*, which are passed to the providers. The providers then send the events from that log source to the consumers that requested it.

9.1.1 ETW Source

Starts an ETW session and registers for ETW providers to receive events from them.

9.1.2 ETW Kernel Source

Starts a SystemTraceProvider ETW session (see Microsoft's Documentation) and registers for System Providers to receive events from them.

9.1.3 Eventlog Source

Regularly polls event logs for new events.

9.1.4 Poll Handles

Regularly polls all handles on a system.

9.2 Consumers

Consumers contain detection and self-protection logic. They register for specific *log sources* that they require in order to work.

9.2.1 Cobalt Strike Beacon Hunter

Detects suspicious processes beaconing to remote systems based on certain communication patterns often found in C2 frameworks, especially Cobalt Strike.

9.2.2 LSASS Dump Detector

Generic detection of LSASS process dumping.

9.2.3 ETW Canary

A detector module that tries to detect tampering with the ETW channels. (self defense mechanism)

9.2.4 Command Line Mismatch Detector

Detects process ghosting and similar process creation anomalies.

9.2.5 Process Tampering Detector

Detects privilege escalation to LOCAL_SYSTEM within a process context and PPL protection changes (e.g. MimiDrv process manipulation)

9.2.6 Temporary Driver Load Detector

Detects driver loading events in which a driver is loaded and quickly unloaded afterwards, which could be a sign of malicious activity.

CHAPTER

TEN

FUNCTION TESTS

There are easy ways to test Aurora and see if it matches suspicious / malicious events.

10.1 Sigma Matching

10.1.1 Sigma Matching - Process Creation

Included in profiles: Minimal, Reduced, Standard, Intense

This should create a `WARNING` level message for a Sigma rule with level `high`.

```
C:\Users\nextron>whoami /priv
```

This should create a `WARNING` level message for a Sigma rule with level `high`.

```
C:\Users\nextron>certutil.exe -urlcache http://test.com
```

10.1.2 Sigma Matching - Network Communication

Included in profiles: Minimal, Reduced, Standard, Intense

This should create a `ALERT` level message for a Sigma rule with level `critical`.

```
C:\Users\nextron>ping aaa.stage.123456.test.com
```

10.1.3 Sigma Matching - File Creation

Included in profiles: Minimal, Reduced, Standard, Intense

This should create a `WARNING` level message for a Sigma rule with level `high`.

```
C:\Users\nextron>echo "test" > %temp%\lsass.dmp
```

10.1.4 Sigma Matching - Process Access

Included in profiles: Standard, Intense

This should create a `WARNING` level message for a Sigma rule with level `high`.

```
PS C:\Users\nextron>$id = Get-Process lsass; rundll32.exe C:\Windows\System32\comsvcs.
↪dll , MiniDump $id.Id $env:temp\lsass.dmp full
```

Cleanup:

```
C:\Users\nextron>del /f %temp%\lsass.dmp
```

10.1.5 Sigma Matching - Registry

Included in profiles: Intense

This should create a `WARNING` level message for a Sigma rule with level `high`.

```
C:\Users\nextron>reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\AuroraTest" /V
↪"AuroraTest" /t REG_SZ /F /D "vbscript"
```

Cleanup:

```
C:\Users\nextron>reg delete "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\AuroraTest" /
↪F
```

10.2 IOC Matching

Note: The Aurora Lite version uses only a very limited set of IOCs.

10.2.1 IOC Matching - Filenames

```
C:\Users\nextron>echo "test" > %temp%\loader.ps1
```

Cleanup:

```
C:\Users\nextron>del %temp%\loader.ps1
```

10.2.2 IOC Matching - C2

Warning: This could trigger an alert in your internal monitoring (old Sofacy C2)

```
C:\Users\nextron>ping drivres-update.info
```

10.2.3 IOC Matching - Hash

TBD

10.2.4 IOC Matching - NamedPipe

Start a named pipe using the following PowerShell commands:

```
PS C:\Users\nextron>$npipeServer = New-Object System.IO.Pipes.NamedPipeServerStream(
↪'testPipe', [System.IO.Pipes.PipeDirection]::InOut)
PS C:\Users\nextron>$npipeServer.Close()
```

Included in profiles: Intense

10.2.5 IOC Matching - Mutex

Create a mutex using the following PowerShell commands:

```
PS C:\Users\nextron>$mtx = New-Object System.Threading.Mutex($true, "agony")
```

Matching might take some time (outside of the Intense profile) since mutexes are polled.

10.3 CommandLineMismatchDetector

Download Process Ghosting PoC release package named "proc_ghost.zip" by `@hasherezade`

Extract the package and then run:

```
C:\Users\nextron>proc_ghost.exe %comspec% c1.exe
```

Note: Only available in the full version (not Aurora Lite)

CHAPTER

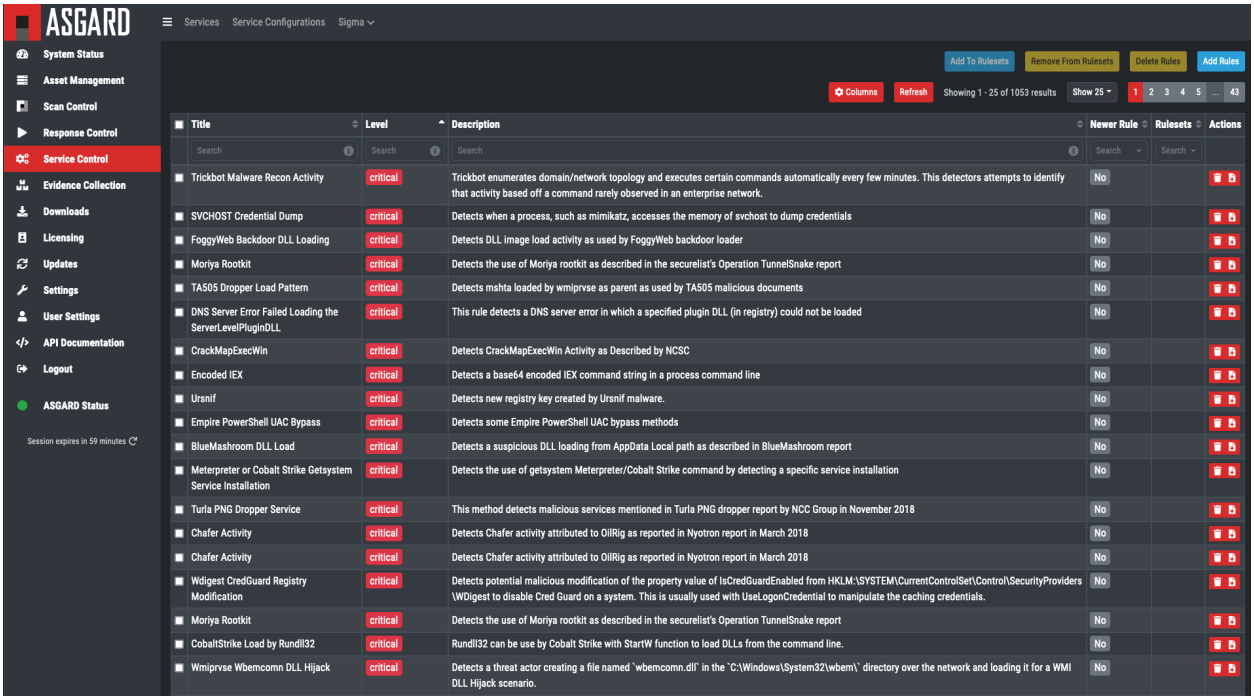
ELEVEN

CUSTOM SIGNATURES AND IOCS

11.1 Management using ASGARD

ASGARD Management Center allows you to create rule sets of sigma rules and apply hem to groups of end systems. It also features a "difference view" that shows you rules that have changed in the remote Sigma repository and allows you to accept or deny the changes. It also provides ways to filter false positives right at the source.

Signature updates and Aurora upgrades can be executed for all end points from the Management Center.



The rule management is described in more detail in this section of the ASGARD Management Center manual.

11.2 Manual signature management

Signatures can be specified when starting Aurora using the `--rules-path` and `--ioc-path` parameters. These parameters default to the built-in rules and IOCs at `signatures\sigma-rules` and `signatures\iocs` and the provided paths for custom signatures at `custom-signatures\sigma-rules` and `custom-signatures\iocs` respectively. Aurora traverses the directories that are specified with these parameters recursively and initializes all signature files it finds.

In order to add new sigma rules or IOCs, you can either:

- Add them to the corresponding subfolder in *custom-signatures*
- Specify the folder where they are located using `--rules-path` or `--ioc-path`

Warning: If you specify `--rules-path` or `--ioc-path`, if you want to use the Aurora built-in rules and IOCs, you need to add them manually as well. E.g.:

```
C:\Program Files\Aurora-Agent>aurora-agent.exe --install --rules-path .\signatures\
↪sigma-rules --rules-path .\my-rules
```

If paths are configured, only the configured paths are used.

11.2.1 Signature format

IOCs follow the same format that THOR IOCs do; the full description can be found in the THOR manual.

Sigma rules must adhere to the specification found in the Sigma repository.

11.2.2 Encrypted signatures

Both IOCs and sigma rules can be encrypted using the `encrypt` function in Aurora Agent Util. Aurora will automatically decrypt encrypted signatures at startup. This functionality is only available in the full version of Aurora.

11.3 Signature Application

11.3.1 Sigma rules

Sigma rules must contain a `logsource` element which (indirectly) determines on which events the sigma rule is applied.

Aurora utilizes a number of *log sources* which map between these `logsource` elements and the actual sources. The log source definitions which can be found in the `log-sources` folder. Rules are applied on every log source which has a matching `logsource` definition.

Log sources may also utilize:

- `conditions` to filter events from the given sources
- `fieldmappings` to rename specific fields in the events that occur. This is useful to have all events with the same `logsource` appear to have the same fields, even if the underlying sources and field names differ.
- `rewrite` to reference each other. `rewrite` is meant to be used in combination with the other elements: For example, Sysmon events are split into different categories using `conditions` and `rewrite`.

11.3.2 IOCs

Hashes

Hash IOCs are applied to:

- Process creation events
- Image load events
- Driver load events

Filenames

Filename IOCs are applied to:

- Process creation events
- Image load events
- File creation events
- Handle events that reference files
- Driver load events

C2

C2 IOCs are applied to:

- DNS query events
- TCP connection events

Named Pipe

Named Pipe IOCs are applied to handle events that reference named pipes.

Handle

Handle IOCs (which include mutex and event IOCs) are applied to handle events.

CHAPTER

TWELVE

AURORA AGENT UTIL

Aurora Agent Util provides utility functions that are not centered around the detection logic:

- Updating Aurora and its signatures
- Encrypting custom signatures
- Excluding "noisy" processes
- Creating a diagnostics pack with debugging information

12.1 Updating Aurora

There are two commands for Aurora Agent Util to update:

- `update` updates Aurora's signatures. The program files are unaffected and will not be updated.
- `upgrade` upgrades the Aurora fully, both program files and signatures.

Both commands only affect the directory that contains the executable unless `--restart-service` is specified. `--restart-service` requires Aurora to have been installed with `--install` previously and will restart the service to reload the updated files.

Examples:

```
C:\Program Files\Aurora-Agent>aurora-agent-util.exe update
C:\Program Files\Aurora-Agent>aurora-agent-util.exe upgrade --restart-service
```

12.2 Encrypting Signatures

You can encrypt your custom signatures (either IOCs or sigma rules) with the `encrypt` command to avoid them being flagged by an Antivirus or to protect them from hostile reads on a potentially compromised system. Aurora will decrypt them at startup.

The encrypted versions of the passed signatures will be placed next to the unencrypted signatures, with a different extension.

Encrypted signatures can be passed to Aurora just like unencrypted ones: Via the `custom-signatures` folder, or by specifying them with `--rules-path` / `--ioc-path`.

Examples:

```
C:\Program Files\Aurora-Agent>aurora-agent-util.exe encrypt path/to/my/sigmarule.yml␣
↪path/to/my/other/sigmarule.yml
```

12.3 Excluding Processes

The `exclude` command requires a running Aurora Agent. It will connect to that Agent for status information about the processes that created the most events and will run a dialogue to comfortably add exclusions for some of these processes.

Examples:

```
C:\Program Files\Aurora-Agent>aurora-agent-util.exe exclude
```

This function is just a more comfortable alternative to adding the exclusions manually in the respective *Process Exclusions* file.

12.4 Creating a Diagnostics Pack

The `diagnostics` command creates a ZIP file with several files that can be analyzed by us in case of issues. If you encounter an issue, the first step is usually sending us this diagnostics pack along with a description of the issues.

Examples:

```
C:\Program Files\Aurora-Agent>aurora-agent-util.exe diagnostics
```

The diagnostics pack includes the status output, service startup logs (if available) and memory profiles that can be analyzed with the help of pprof.

CHAPTER

THIRTEEN

LIST OF EVENT IDS

This list contains all event IDs that Aurora can produce. Event IDs are used when logging to the Windows Eventlog, and can also be included in the log message using *--print-event-id*.

13.1 Sigma related event IDs

Event ID	Description
1	A **process creation** Sigma rule matched.
2	A **set file creation time** sigma rule matched.
3	A **network connection** sigma rule matched.
4	A **sysmon status** Sigma rule matched.
5	A **process termination** Sigma rule matched.
6	A **driver loaded** Sigma rule matched.
7	An **image loaded** Sigma rule matched.
8	A **create remote thread** Sigma rule matched.
9	A **raw disk access** Sigma rule matched.
10	A **process access** Sigma rule matched.
11	A **file creation** Sigma rule matched.
12	A **registry event** Sigma rule matched.
15	A **create stream hash** Sigma rule matched.
17	A **pipe event** Sigma rule matched.
19	A **WMI event** Sigma rule matched.
12	A **registry event** Sigma rule matched.
22	A **DNS query** Sigma rule matched.
23	A **file deletion** Sigma rule matched.
95	An error occurred while loading the Sigma rules.
96	Sigma rules were reloaded.
97	No Sigma rule files were found.
98	Unspecified log message from Sigma module.
99	Another Sigma rule (that did not belong to one of the above categories) matched.
6000	A response for a sigma match was executed.
6001	A response for a sigma match was simulated.

13.2 Internal event IDs

Event ID	Description
100	A license file was found.
101	Status message (from `--report-stats`)
102	Aurora Agent started.
103	Aurora Agent is terminating.
104	The current license expired.
105	No valid license file was found.
107	A process created a large amount of events.
108	An internal panic occurred.

13.3 Event IDs for other modules

Event ID	Module
200	BeaconHunter
300	Lsass Dump Detector
400	ETW Canary
500	Process Tampering Detector
600	Temporary Driver Load Detector
700	Command Line Mismatch Detector
800	Event Distributor
900	ETW Provider
1000	Eventlog Provider
1100	Handle Polling Provider
1200	Resource Control
1301	Filename IOC Match Found

CHAPTER

FOURTEEN

DEBUGGING

The best way to debug Aurora is to run it directly in the command line and don't use it as service.

If you have already installed it, use the following command:

```
C:\Program Files\Aurora Agent\>aurora-agent-64.exe --debug
```

Otherwise just run it from the folder to which you have extracted the Aurora program package:

```
aurora-agent-64.exe --debug
```

The verbosity can even be increased by using the `--trace` flag. With `--trace`, a log entry will be generated for every incoming event.

`--debug` and `--trace` apply to all outputs (log file, UDP / TCP, command line) except for the Windows Eventlog,

14.1 Status Information

The `--status` can give you information from a running Aurora service.

```
C:\Program Files\Aurora Agent>aurora-agent-64.exe --status
Aurora Agent
Version: 0.1.6
Build Revision: 5fef68a1
Sigma Revision: 0.20-1884-ga4a26540
Status: running
Uptime (in hours): 0

Active Outputs:
Eventlog: enabled
Stdout: enabled

Rule Statistics:
Loaded rules: 734
Number of rule reloads: 0

Event Statistics:
Events observed so far: 89419
Events lost so far: 0
Sigma matches: 4
Suppressed Sigma matches of those: 0
```

(continues on next page)

(continued from previous page)

```

Response Actions: disabled
```

It displays the number of events that the agent was able to see and process, the number of initialized rules and rule matches.

Adding the flag `--trace` includes more information in the output, e.g. the number of processed events per ETW event channel.

```
C:\Program Files\Aurora Agent>aurora-agent-64.exe --status --trace
Aurora Agent
Version: 0.1.6
Build Revision: 5fef68a1
Sigma Revision: 0.20-1884-ga4a26540
Status: running
Uptime (in hours): 0

Active Outputs:
Eventlog: enabled
Stdout: enabled

Rule Statistics:
Loaded rules: 734
Number of rule reloads: 0

Event Statistics:
Events observed so far: 111138
        88650 events from WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls
        20094 events from WinEventLog:Microsoft-Antimalware-Engine
        1265 events from PollNamedPipes
        400 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_
↪IMAGE
        306 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_
↪PROCESS
        174 events from WinEventLog:Microsoft-Windows-Kernel-Registry/CreateKey
        164 events from SystemLogger:Process
        46 events from WinEventLog:Microsoft-Windows-DNS-Client
        19 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_
↪CREATE_NEW_FILE
        11 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_
↪DELETE_PATH
        9 events from WinEventLog:Microsoft-Windows-TCPIP/ut:ConnectPath
Events lost so far: 0
Sigma matches: 4
        Run Whoami Showing Privileges: 2
        Suspicious Certutil Command: 2
Suppressed Sigma matches of those: 0

Response Actions: disabled
```

The JSON output includes all the information plus the agent configuration settings.

```
C:\Program Files\Aurora Agent>aurora-agent-64.exe --status --json
```

(continues on next page)

(continued from previous page)


```
{
    "Parameters": {
        "SigmaFolders": [
            "C:\\aurora\\rules"
        ],
        "AutoReload": false,
        "LogFile": "",
        "LogSources": [
            "log-sources\\etw-log-sources-standard.yml"
        ],
        "Debug": false,
        "Trace": false,
        "NoEventlog": false,
        "ReportingLevel": "high",
        "Json": false,
        "LicensePath": "C:\\aurora\\",
        "UdpTarget": "",
        "Silent": false,
        "CpuLimit": 100,
        "ReportStats": false,
        "ReportStatsInterval": 3600000000000,
        "LogRotateCount": 7,
        "LogSize": 10485760,
        "AgentName": "aurora-agent",
        "ActivateModules": null,
        "DeactivateModules": null,
        "NoStdout": false,
        "EventThrottling": 0,
        "LowPrio": false,
        "PrintEventId": false,
        "ConsumerParameters": {
            "ActivateResponses": false,
            "DumpFolder": ".",
            "SigmaMatchThrottling": 60000000000,
            "SigmaMatchBurst": 5
        },
        "ProviderParameters": {
            "NoHashes": false
        }
    },
    "Uptime": 334601989600,
    "Version": "0.1.6",
    "SigmaRevision": "0.20-1884-ga4a26540",
    "BuildRevision": "5fef68a1",
    "CurrentAction": "running",
    "LoadedRules": 734,
    "ReloadCounter": 0,
    "EventsProcessed": {
        "PollNamedPipes": 1815,
        "SystemLogger:Process": 175,
        "WinEventLog:Microsoft-Antimalware-Engine": 27847,
        "WinEventLog:Microsoft-Windows-DNS-Client": 57,
```


(continues on next page)

(continued from previous page)

```
        "WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls": 124269,
        "WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_CREATE_NEW_FILE": 22,
        "WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_DELETE_PATH": 11,
        "WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_IMAGE": 645,
        "WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_PROCESS": 314,
        "WinEventLog:Microsoft-Windows-Kernel-Registry/CreateKey": 342,
        "WinEventLog:Microsoft-Windows-TCPIP/ut:ConnectPath": 26
    },
    "EventsLost": 0,
    "SigmaMatches": {
        "Run Whoami Showing Privileges": 2,
        "Suspicious Certutil Command": 2
    },
    "SuppressedSigmaMatches": {},
    "ActiveModules": null
}
```

14.2 Diagnostic information

14.2.1 Diagnostic pack

You can create a diagnostic pack to detect and debug performance problems.

Simply run:

```
C:\Program Files\Aurora Agent>aurora-agent-util.exe diagnostics
```

This creates a ZIP file with debugging information (such as heap usage, stack traces, ...) that we can use to analyze these issues.

14.2.2 Profiling server

If Aurora has been started with `--pprof`, information can also be gathered manually via a web interface:

```
curl.exe http://localhost:8080/debug/pprof/profile?seconds=20 --output aurora-debug.pprof
curl.exe http://localhost:8080/debug/pprof/heap --output aurora-heap.pprof
curl.exe http://localhost:8080/debug/pprof/goroutine --output aurora-stack-traces.pprof
```

This is the same information that is included in the diagnostic pack.

14.3 Crashes

In cases of unexpected crashes, the following command lines can help you identify the source of the problem.

```
C:\Program Files\Aurora Agent>aurora-agent.exe -c agent-config.yml > aurora-crash.log 2>&
↪1
```

```
C:\Program Files\Aurora Agent>aurora-agent.exe -c agent-config.yml --trace > aurora-
↪crash-trace.log 2>&1
```

14.4 Error Messages

Check the configured log outputs for error messages. A faulty rule would e.g. lead to error messages like this one in the `Application` eventlog with EventID

```
Could not reload sigma rules
Module: Aurora-Agent
Changed_files: C:\Program Files\Aurora-Agent\myrules\my-ransomware.yml
Error: could not parse rule response in file "C:\\Program Files\\Aurora-Agent\\myrules\\
↪my-ransomware.yml": invalid predefined response action kil
```

14.5 Performance Tuning

14.5.1 Event Source Tuning

Event Sources and Consumers

Internally, Aurora has a number of event consumers. The event consumers are:

- Aurora's built-in modules
- Sigma log sources

Each event consumer consists of:

- A number of requested event sources
- Logic to handle incoming events from these sources

Performance is primarily determined by the number of incoming events that Aurora has to process; The impact of Sigma rule matching, in comparison, is fairly low.

Therefore, to optimize performance, choose your event sources wisely and avoid event sources that produce an extreme number of events.

Event Source Analysis

When executing `aurora-agent.exe --status --trace` while Aurora is running, an overview of events that was received for each event source is generated. The performance impact of each source scales roughly linear with the number of events.

To see which built-in modules requests which event source, the requested log sources can be listed with `aurora-agent.exe --module-info --trace`.

For sigma log sources, inspect the sigma configurations that are used by Aurora. By default, `etw-log-sources.yml` and `default-log-sources.yml` from the Aurora directory are used.

Each sigma log source in these files that has a `sources` element requests the event sources listed in that element.

Event Source Definitions

Aurora Agent supports the following event source prefixes:

- `WinEventLog:` Events from an Eventlog channel or ETW provider.

  The schema for these sources is: `WinEventLog:Provider/Channel?Options`

  Channels and options are optional and add further restrictions on events from the provider that is requested. A full list of Eventlog channels on a system can be found using the Event Viewer. A full list of ETW providers on a system can be found using e.g. https://github.com/zodiacon/EtwExplorer.

- `SystemLogger:` Events from the System Trace Provider (see https://learn.microsoft.com/en-us/windows/win32/etw/configuring-and-starting-a-systemtraceprovider-session for details).

  The schema is: `SystemLogger:SystemLoggerFlag` where the supported `SystemLoggerFlag` flags are:

  - FileIO
  - Process
  - Thread
  - Registry
  - Image
  - Network-TCP-IP
  - Handle

- `PollHandles`: This event source is handled by a provider in Aurora that regularly creates an event for each handle that exists on a system.

Example: Disabling a Noisy Log Source

In this example, say that `aurora-agent.exe --status --trace` results in this event overview:

```
Events observed so far: 50657
     36783 events from WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls
     7611 events from WinEventLog:Microsoft-Windows-Kernel-File?eventids=14
     1842 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_IMAGE
     1273 events from WinEventLog:Microsoft-Windows-Kernel-Registry/CreateKey
     1058 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_
↪THREAD
     995 events from WinEventLog:Microsoft-Windows-DNS-Client
```

(continues on next page)

(continued from previous page)

```
    585 events from PollNamedPipes
    235 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_
↪FILENAME
    169 events from SystemLogger:Process
    39 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_CREATE_
↪NEW_FILE
    32 events from WinEventLog:Microsoft-Windows-Sysmon/Operational
    22 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_PROCESS
    5 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_DELETE_
↪PATH
    3 events from WinEventLog:Security
    3 events from WinEventLog:Application
    2 events from WinEventLog:Microsoft-Windows-Kernel-Registry/DeleteKey
```

As we can see, the by far noisiest event source is `WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls`.

If we want to disable this event source to lessen Aurora's CPU usage, we must find the event consumers that request it.

`aurora-agent.exe --module-info --trace` shows these modules which use this event source:

```
Aurora Agent Modules:
    LsassDumpCheck
            Requested sources:
            ...
            WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls
    Beaconhunter
            Requested sources:
            ...
            WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls
            ...
```

Searching in `etw-log-sources.yml`, we find that there is also a Sigma log source definition which uses this event source:

```
windows-api-call-auditing:
   product: windows
   service: api-call-auditing
   sources:
      - "WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls"
```

To deactivate this log source, we therefore need to deactivate both modules which use this source using *--deactivate-module* and remove the log source definition from the sigma configuration.

Obviously, this will also impact Aurora's detection capabilities to some degree. Choose your trade-off between detection and performance carefully.

Examples

```
# Exclude a specific process
^C:\\Program Files\\My Antivirus\\antivirus\.exe$

# Exclude Windows Defender
^C:\\ProgramData\\Microsoft\\Windows Defender\\Platform\\[^\\]{5,20}\\MsMpEng\.exe$
```

CHAPTER

FIFTEEN

KNOWN ISSUES

15.1 AUR#002: Missing `exclude` Feature in Aurora Util Lite

Introduced Version	Fixed Version
0.8.0	Open

The process exclusion feature that allows to add a process automatically to the `conf/process-exclusions.cfg` automatically is missing in Aurora Lite Util and only available in the full version. This is an error.

15.2 AUR#001: Missing Self-Defense

Introduced Version	Fixed Version
N/A	Open

- Folder permission checks
- ETW manipulations (the ETW canary module in the full Aurora version already covers some of them)
- Warning events on configuration changes

CHAPTER

SIXTEEN

DETECTION GAPS

Aurora Agent uses ETW to observe a system. However, there are some parts of the system where there are no ETW events or the available events are not easily usable. Also, an attacker might try to prevent ETW events from reaching Aurora.

16.1 Named Pipes

There is no ETW provider that provides information about creation of or connection to named pipes. The only way to observe named pipe events is the Kernel Object Handle provider which provides information about all handles that are opened and closed, but which is therefore very "noisy" and only enabled in the intense configuration.

We've integrated a polling for named pipes that would detect opened named pipes if they exist for more than 10 seconds. In the default configuration we miss named pipes that only exist for a very short amount of time.

16.1.1 Named Pipes - Solution

To close this detection gap:

- Use Aurora with the "Intense" configuration preset (could cause high CPU load on systems)
- Additionally install and use Sysmon with
- this configuration.

16.2 Registry Events

While there are a couple of ETW providers for registry events such as creating keys or writing values (primarily `Microsoft-Windows-Kernel-Registry`), their information is not directly usable. Each event references other keys by their handle, meaning that all registry handles must be tracked for these events to be useful. Since doing so is fairly expensive in CPU time, Aurora does so in the intense configuration, but not in the standard configuration.

The events for setting values here also are apparently broken; while the manifest contains a field for the written data, it appears to be empty. To query the new value of the registry key, it is necessary to read the registry after receiving the event (which is less reliable since the value might have been changed again).

```
EventID:2 Provider_Name:Microsoft-Windows-Kernel-Registry
BaseName: BaseObject:0xFFFFB700E2344C40 Disposition:0 KeyObject:0xFFFFB700E23442C0
RelativeName:{34ce6e13-aec9-43ca-a80c-7ee47260ef84} Status:0x0

EventID:7 Provider_Name:Microsoft-Windows-Kernel-Registry
```

(continues on next page)

(continued from previous page)

```
CapturedDataSize:0 DataSize:16 InfoClass:2 KeyName: KeyObject:0xFFFFB700E23442C0
Status:0x0 ValueName:EnableDhcp

Could not parse event ERROR: "failed to parse \"CapturedData\" value; TdhFormatProperty␣
↪failed; The parameter is incorrect."
```

These captured events display the issues here: The first event (with Event ID 2) is an *OpenKey* event. While we see the relative name, the `BaseName` field is empty. Therefore, in order to determine the full name of the key, we need to correlate this with previous *OpenKey* events for the key referenced as `BaseObject`.

The second event (with Event ID 7) is a *QueryValue* event. Again, the `KeyName` is empty; instead, the `KeyObject` field needs to be correlated with previous *OpenKey* events. The data that was returned from the *QueryValue* is also missing. There is a field for it, (`CapturedData`) but it is apparently empty based on the `CapturedDataSize` and querying its value fails with the displayed error message.

16.2.1 Registry Events - Solution

To close this detection gap:

- Use Aurora with the "Intense" configuration preset (could cause high CPU load on systems with a lot of registry access events)
- Additionally install and use Sysmon with
- this configuration.

We have been in contact with Microsoft to get the registry related ETW events fixed and extended in future Windows versions. However, Microsoft responded that it is hard to determine the real demand for a solution to these issues.

16.3 ETW disabling

Since ETW events partially originate from user space, an attacker can disable user space ETW events from its own process by patching the syscalls that Windows uses to create ETW events. Doing so is, in fact, common for attacker frameworks.

While this does not make Aurora useless, you should be aware of this when writing detection rules that are based on these providers. Usually, any event that originates from the process and is caused by a provider that does not start with `Microsoft-Windows-Kernel` can be suppressed and should be handled with care.

16.3.1 ETW disabling - Solution

- The full version of Aurora uses a ETW Canary module to detect ETW manipulations.
- The flag `--report-stats` allows you to report the status of the agent to your central log collector (SIEM). This status includes statistics of the observed, process and dropped events that can be used to detect manipulations. (e.g. number of observed events doesn't increase over time)

16.4 Handle polling and asynchronous handles

Some handles are asynchronous, meaning that it is impossible to query their named and type; trying to do so causes the querying thread to permanently hang, ultimately leading to resource leaks. Unfortunately, it is impossible to query from user space whether a handle is asynchronous, so there is no guaranteed way of avoiding these handles either.

In practice, this means that Aurora tries to predict based on a handle's other properties whether a handle would hang and skips those. However, this also causes Aurora to skip some "valid" handles, meaning they won't be listed by the `Handle-Polling` module.

16.4.1 Handles - Solution

- Use Aurora with the "Intense" configuration preset (could cause high CPU load on systems)
- Additionally install and use Sysmon with
- this configuration
- to at least guarantee events about named pipes

CHAPTER

SEVENTEEN

FREQUENTLY ASKED QUESTIONS

17.1 Why does Aurora use a lot of memory?

The short answer is: because it can.

The long answer is related to the way the go runtime manages the memory. There are many articles that describe the way how the garbage collector works but only a few that describe situations in which a program used a unexpectedly high amount of memory.

Note: It turns out that there was a change in Go 1.12 regarding how the runtime signals the operating system that it can take unused memory. Before Go 1.12, the runtime sends a `MADV_DONTNEED` signal on unused memory and the operating system immediately reclaims the unused memory pages. Starting with Go 1.12, the signal was changed to `MADV_FREE`, which tells the operating system that it can reclaim some unused memory pages if it needs to, meaning it doesn't always do that unless the system is under memory pressure from different processes.

So, yes, it is possible that the Aurora agent uses much more memory than the usual 200-300 MB, but only in cases in which there is a lot of free available memory. The operating system should be able to claim that excessive memory whenever needed.

If you notice that this is not the case, please provide a diagnostics pack, which also includes a complete memory profile of a running Aurora agent.

See the section *Creating a Diagnostics Pack* of the Aurora Agent Util chapter for details.

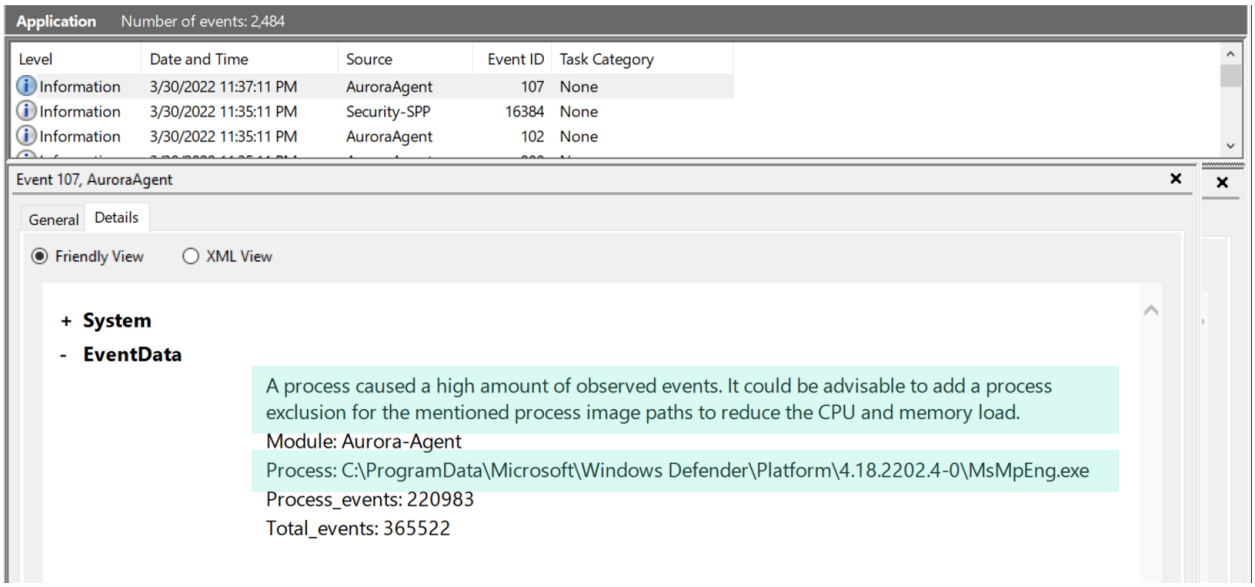
17.2 What's the impact of Sigma rule matching on the agent's performance?

Users are often interested in the matching process and ask questions like: "What's the performance impact with such a design? How scalable it is, let's say with 100 / 1000 / 10,000 sigma rules?"

The answer is that the impact isn't proportionate and thus much lower than expected. We use a matching logic that is very similar to the one used in i.e. YARA and therefore adding 1,000 rules to existing 1,000 rules would only slow down the agent by around 1-3%.

The most CPU cycles are spent on reading and parsing the events from the different ETW event channels. This means that a process that produces a disproportionately high number events in these channels causes much more impact than adding 1,000 or 10,000 sigma rules.

Aurora has some detection logic to detect and report such processes in separate log messages with ID 107. In the release version Aurora reports all processes that are responsible for more than 50% of the total number of events.



17.3 Why does Aurora Lite use the newest rules while Aurora doesn't?

The rules used by our commercial product go through an intensive internal testing process before we release them to our customers. The rule set used by Aurora includes the Github repository maintained by the Sigma community and Nextron's own private Sigma rules.

The tests include:

- Sigma tests against EVTX files exported from 30+ different test systems
- Live tests on 4+ Windows machines with simulated user activity
- 24h live endurance tests on 30+ different test systems

The Aurora Lite version always uses the current `master` of the Github repository maintained by the Sigma community. This set goes through some rudimentary testing against exported EVTX files but isn't tested on live systems.

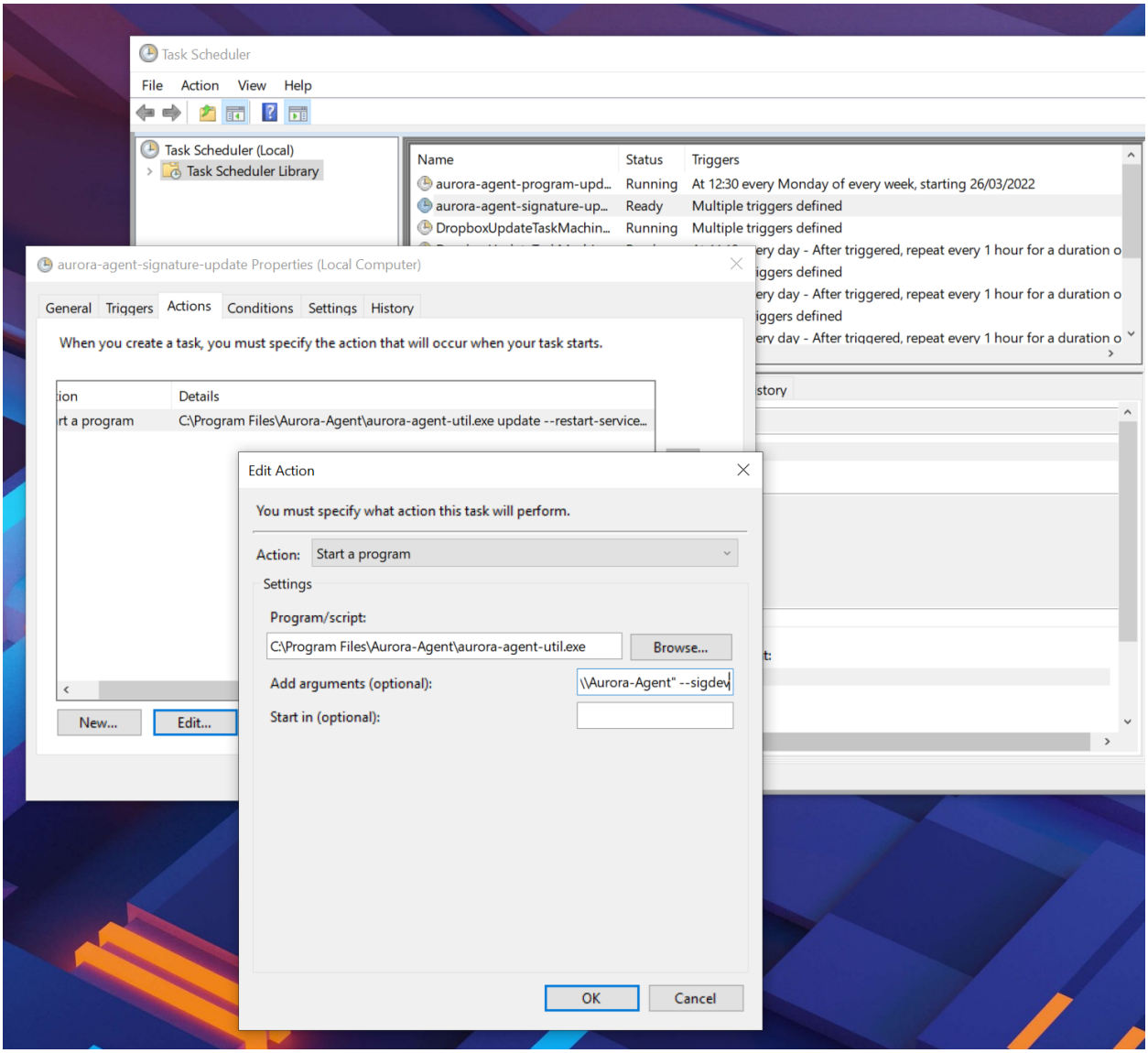
If you want to use the most current and untested rule set, you can add the `--sigdev` flag to the command line flags used by the update tasks (add it right after the `update` or `upgrade` command). Please be aware that support cases caused by the use of that untested rule set may not be covered by the existing maintenance or support contract.

17.4 Why does Aurora generate two alerts for a single event?

Aurora registers to different event channels that sometimes contain the same information. It is possible that Aurora notices the same activity in two different channels and generates two alerts for a single event.

In these cases, the alerts should have different values set in the `Provider_Name` field, e.g. `Provider_Name: Microsoft-Windows-Kernel-Process` and `Provider_Name: Microsoft-Windows-Sysmon`.

It is unclear which of the two alerts should be suppressed in order to avoid these duplicate notifications, as they do not include identical information and in some situations one is preferred over the other or vice versa.


Task Scheduler
File Action View Help
Task Scheduler (Local)
Task Scheduler Library
Name Status Triggers
aurora-agent-program-upd... Running At 12:30 every Monday of every week, starting 26/03/2022
aurora-agent-signature-up... Ready Multiple triggers defined
DropboxUpdateTaskMachin... Running Multiple triggers defined
aurora-agent-signature-update Properties (Local Computer)
General Triggers Actions Conditions Settings History
When you create a task, you must specify the action that will occur when your task starts.
Details
rt a program C:\Program Files\Aurora-Agent\aurora-agent-util.exe update --restart-service...
New... Edit...
Edit Action
You must specify what action this task will perform.
Action: Start a program
Settings
Program/script:
C:\Program Files\Aurora-Agent\aurora-agent-util.exe
Browse...
Add arguments (optional):
\\Aurora-Agent" --sigdev
Start in (optional):
OK Cancel

17.5 How do I view the suppressed Sigma matches?

In some messages, e.g. in the reported statistics (`--report-stats`) or the status message (`--status`), you may find a number of suppressed Sigma matches

```
Event Statistics:
Events observed so far: 23483429
Events lost so far: 0
Sigma matches: 13
Suppressed Sigma matches of those: 6
```

Use the flag combination `--status --trace` to view which Sigma rule matches have been suppressed.

```
C:\Program Files\Aurora Agent>aurora-agent-64.exe --status --trace
Aurora Agent
Version: 0.9.9
Build Revision: 9280d44aef722
Signature Revision: 2022/03/25-161029
Sigma Revision: 0.20-3393-g952f14d8
Status: running
Uptime (in hours): 0

Active Outputs:
Windows Application Eventlog: enabled

Resource Usage:
CPU Cores: 2
Total Memory: 4.00GB
Used Memory: 2.65GB
Used by Aurora: 346.47MB

Log Messages:
Errors: 0
Alerts: 0
Warnings: 6
Notices: 13

Active Modules: LsassDumpDetector, BeaconHunter, EtwCanary, CommandLineMismatchDetector,␣
↪ProcessTamperingDetector, TemporaryDriverLoadDetector, ApplyIOCs, Rescontrol, Sigma,␣
↪ETWSource, ETWKernelSource, EventlogSource, PollHandles

Rule Statistics:
Rule paths: C:\Program Files\Aurora-Agent\signatures\sigma-rules, C:\Program Files\
↪Aurora-Agent\custom-signatures
Loaded rules: 1299
        custom: 2
        private: 18
        public: 1279
Rule reloads: 0
Responses: 0
Process dump path: C:\Program Files\Aurora-Agent\process-dumps

Loaded IOCs:
```

(continues on next page)

(continued from previous page)

```
Domain IOCs: 8425
        internal: 8425
Filename IOCs: 6894
        internal: 6894
Handle IOCs: 581
        internal: 581
Hash IOCs: 8448
        custom: 1
        internal: 8447
Namedpipe IOCs: 100
        internal: 100

Event Statistics:
Events observed so far: 4003363
By source:
        1432318 events from PollHandles
        1108254 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_
↪KEYWORD_CREATE
        872554 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_
↪FILEIO?eventids=14
        353165 events from WinEventLog:Microsoft-Windows-Sysmon/Operational
        162140 events from WinEventLog:Microsoft-Windows-Kernel-Audit-API-Calls
        30112 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_
↪IMAGE
        15275 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_
↪THREAD
        8113 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_
↪DELETE_PATH
        4738 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_
↪CREATE_NEW_FILE
        4717 events from WinEventLog:Microsoft-Windows-Kernel-File/KERNEL_FILE_KEYWORD_
↪RENAME_SETLINK_PATH
        3826 events from WinEventLog:Microsoft-Windows-DNS-Client
        3558 events from WinEventLog:Microsoft-Windows-TCPIP/ut:ConnectPath
        1302 events from SystemLogger:Process
        1236 events from WinEventLog:Security
        756 events from WinEventLog:Microsoft-Windows-TaskScheduler/Operational
        639 events from WinEventLog:Microsoft-Windows-Kernel-Process/WINEVENT_KEYWORD_
↪PROCESS
        257 events from WinEventLog:Microsoft-Windows-WinINet/WININET_KEYWORD_HANDLES
        127 events from WinEventLog:{fbb4fbaa-2ae9-5b86-6d76-09930a11a03d}?fromownpid=1
        120 events from WinEventLog:System
        57 events from WinEventLog:Microsoft-Windows-Windows Firewall With Advanced␣
↪Security/Firewall
        27 events from WinEventLog:Microsoft-Windows-WMI-Activity/Operational
        25 events from WinEventLog:Microsoft-Windows-PowerShell
        23 events from WinEventLog:Application
        8 events from WinEventLog:Windows PowerShell
        6 events from WinEventLog:Microsoft-Windows-Kernel-PnP/DriverLoad
        5 events from WinEventLog:Microsoft-Windows-Windows Defender/Operational
        4 events from WinEventLog:Microsoft-Windows-Kernel-PnP/DriverUnload
        1 events from WinEventLog:Microsoft-Windows-SmbClient/Security
```

(continues on next page)

(continued from previous page)

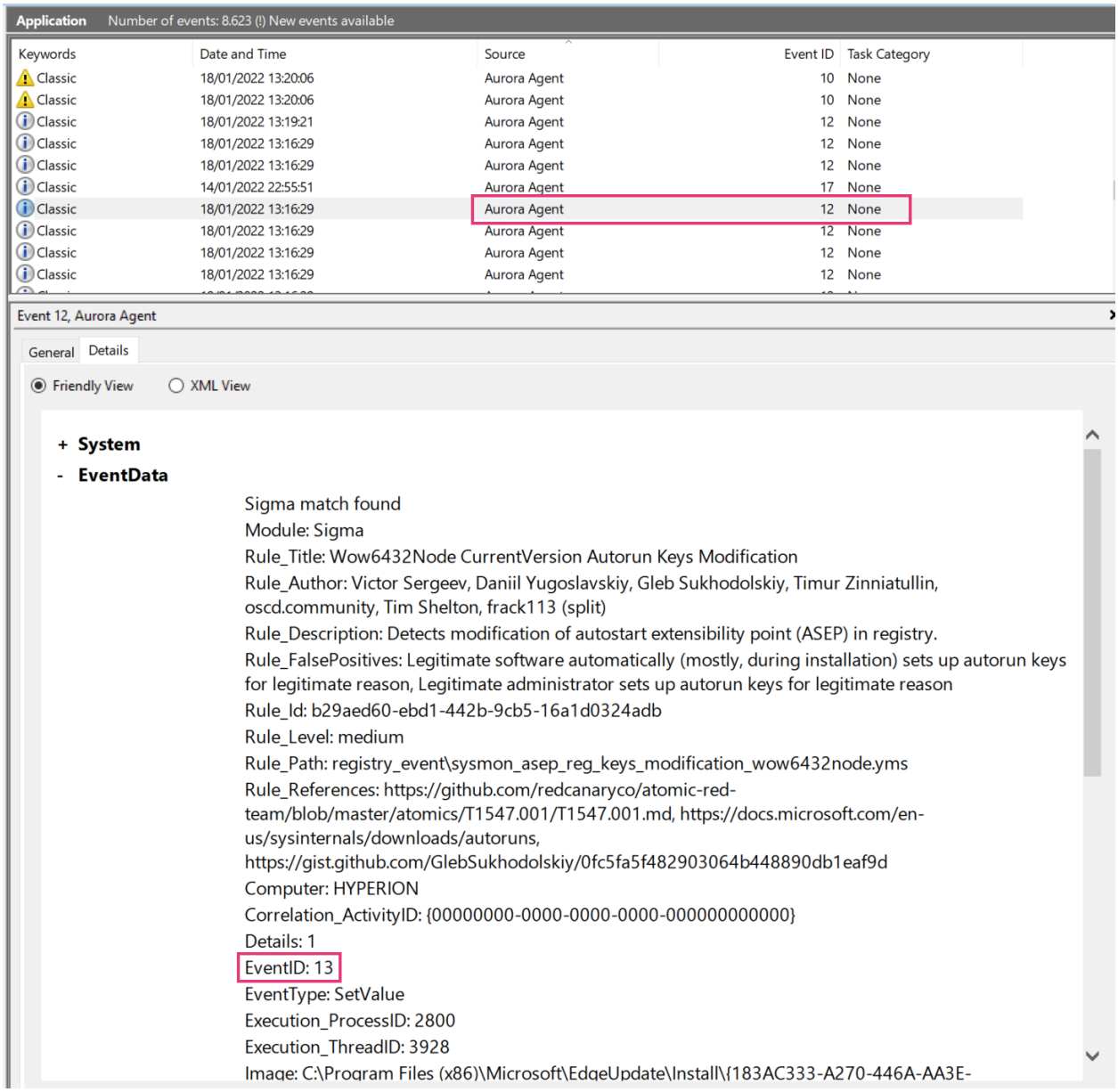
```
By process:
        1146976 events from C:\Windows\System32\svchost.exe
        875516 events from C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2202.
↪4-0\MsMpEng.exe
        519059 events from C:\Windows\Sysmon64.exe
        142271 events from C:\Windows\System32\RuntimeBroker.exe
        110926 events from C:\Windows\explorer.exe
        108878 events from System
        99896 events from C:\Users\neo\Downloads\ProcessExplorer\procexp64.exe
        77899 events from C:\Users\neo\AppData\Local\Programs\Microsoft VS Code\Code.exe
        64256 events from C:\aurora-beta\aurora-agent-util.exe
        ...

False positive filters: 0
Process excludes: 0

Events missed so far: 0
Sigma matches: 28
Whoami Execution: 12
Run Whoami Showing Privileges: 9
Suspicious WSMAN Provider Image Loads: 4
New TaskCache Entry: 2
Run Once Task Configuration in Registry: 1
Suppressed Sigma matches of those: 9
Whoami Execution: 6
Run Whoami Showing Privileges: 3

Response Actions: disabled
```

The match throttling can be configured with the flags `--match-burst` and `--match-throttling`. We recommend keeping the default. It does not suppress matches of a rule that you haven't already noticed in the defined time frame (each rule triggers at least `--match-burst` number of times before being throttled). It only throttles numerous matches of a single rule; cases in which a single rule causes numerous matches in the defined time frame, which is typically the cause of a noisy / too sensitive rule.

17.6 Why does the Event ID in the Windows Eventlog differ from the one in the Event Data?

There's a difference between the Event IDs in the source channels and the Event IDs that we use to write into the various output channels.

The Event ID that you find in the event data is the one provided in the ETW channel that Aurora subscribes to. The Event ID used to write these events into the local Windows Eventlog differ from these Event IDs and are controlled by Aurora.


Application Number of events: 8,623 (!) New events available
Keywords Date and Time Source Event ID Task Category
Classic 18/01/2022 13:20:06 Aurora Agent 10 None
Classic 18/01/2022 13:20:06 Aurora Agent 10 None
Classic 18/01/2022 13:19:21 Aurora Agent 12 None
Classic 18/01/2022 13:16:29 Aurora Agent 12 None
Classic 18/01/2022 13:16:29 Aurora Agent 12 None
Classic 14/01/2022 22:55:51 Aurora Agent 17 None
Classic 18/01/2022 13:16:29 Aurora Agent 12 None
Classic 18/01/2022 13:16:29 Aurora Agent 12 None
Classic 18/01/2022 13:16:29 Aurora Agent 12 None
Classic 18/01/2022 13:16:29 Aurora Agent 12 None
Event 12, Aurora Agent
General Details
Friendly View
XML View
+ System
- EventData
Sigma match found
Module: Sigma
Rule_Title: Wow6432Node CurrentVersion Autorun Keys Modification
Rule_Author: Victor Sergeev, Daniil Yugoslavskiy, Gleb Sukhodolskiy, Timur Zinniatullin, oscd.community, Tim Shelton, frack113 (split)
Rule_Description: Detects modification of autostart extensibility point (ASEP) in registry.
Rule_FalsePositives: Legitimate software automatically (mostly, during installation) sets up autorun keys for legitimate reason, Legitimate administrator sets up autorun keys for legitimate reason
Rule_Id: b29aed60-ebd1-442b-9cb5-16a1d0324adb
Rule_Level: medium
Rule_Path: registry_event\sysmon_asep_reg_keys_modification_wow6432node.yms
Rule_References: https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1547.001/T1547.001.md, https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns, https://gist.github.com/GlebSukhodolskiy/0fc5fa5f482903064b448890db1eaf9d
Computer: HYPERION
Correlation_ActivityID: {00000000-0000-0000-0000-000000000000}
Details: 1
EventID: 13
EventType: SetValue
Execution_ProcessID: 2800
Execution_ThreadID: 3928
Image: C:\Program Files (x86)\Microsoft\EdgeUpdate\Install\{183AC333-A270-446A-AA3E-

17.7 Why does Aurora take so long to start?

Almost all of the startup time comes from loading and compiling the IOCs and Sigma rules. `--debug` gives more information on what Aurora is doing during startup.

If you don't need all IOCs and Sigma rules, it can be helpful to use `--deactivate-module`, `--ioc-path` and `--rules-path` to significantly reduce the startup time:

- `--deactivate-module ApplyIOCs --rules-path my-custom-rule.yml` deactivates IOCs completely and only loads the specified sigma rule.
- `--deactivate-module Sigma --ioc-path my-custom-filename-ioc.txt` deactivates Sigma rules completely and only loads the specified filename IOC file.

17.8 Why doesn't Aurora report Registry matches?

The reason is that ETW provides only insufficient data in the respective event channels. Aurora has to perform some resource intensive check whenever processes access the Windows registry. We have therefore activated these checks only in the "intense" preset.

See chapter *Detection Gaps* for more details.

CHAPTER

EIGHTEEN

CHANGELOG

This chapter contains all new changes of Aurora.

18.1 Aurora Agent 1.2

18.1.1 Aurora Agent Version 1.2.1

Type	Description
Bugfix	Fixed an issue where NT paths starting with `\Windows\` were not correctly converted to DOS paths

18.1.2 Aurora Agent Version 1.2.0

Type	Description
Feature	Added temporary excludes for process / provider / event ID combinations with very high event rates
Feature	Improved speed of sigma rule matching
Feature	Added support for polling log files
Feature	Removed support for custom actions
Bugfix	Fixed an issue where Aurora could crash if too many handles were open
Feature	Added Trace Event Tab in Dashboard

18.2 Aurora Agent 1.1

18.2.1 Aurora Agent Version 1.1.5

Type	Description
Bugfix	Fixed an issue where the handle polling provider did not handle large PIDs correctly

18.2.2 Aurora Agent Version 1.1.4

Type	Description
Bugfix	Fixed an issue where some new log sources were not loaded properly

18.2.3 Aurora Agent Version 1.1.3

Type	Description
Bugfix	Fixed an issue where small files could cause issues with magic header detection

18.2.4 Aurora Agent Version 1.1.2

Type	Description
Bugfix	Fixed an issue where Aurora could leak process handles when analyzing stack traces, possibly leading to high memory load when running for a long time
Feature	Added functionality to the ResControl module to terminate Aurora if a handle leak is detected

18.2.5 Aurora Agent Version 1.1.0

Type	Description
Feature	Added a field for process creation events that indicates whether process parent spoofing took place
Feature	Added support for call traces from ETW events. Extended call traces (with full symbols) are also possible.
Feature	Added CallTrace (and, in intense configuration, CallTraceExtended) fields to OpenProcess events
Feature	Added new flags to set output format specifically for some outputs
Feature	Added a '--dashboard' option that starts an interactive notifier for checking recent Aurora events
Bugfix	Fixed an issue where rules with multiple wildcards could cause extremely high memory usage

18.3 Aurora Agent 1.0

18.3.1 Aurora Agent Version 1.0.7

Type	Description
Bugfix	Fixed an issue in go-sigma that can lead to high memory usage during initialization
Feature	Added default exclusion for Windows Defender

18.3.2 Aurora Agent Version 1.0.6

Type	Description
Feature	Improved performance of Sigma rule matching
Feature	Added a new log source for 'NtSetInformationKey' calls
Feature	Added a timeout for receiving the agent status

18.3.3 Aurora Agent Version 1.0.5

Type	Description
Change	Removed unnecessary files from ZIP package
Bugfix	Fixed an issue where memory was leaked over time

18.3.4 Aurora Agent Version 1.0.4

Type	Description
Change	Improved logging of file hash calculation

18.3.5 Aurora Agent Version 1.0.3

Type	Description
Change	Improved handling of signals received during Aurora startup

18.3.6 Aurora Agent Version 1.0.2

Type	Description
Feature	Made additional flags available from ASGARD
Feature	Added a warning when using keyword IOCs

18.3.7 Aurora Agent Version 1.0.1

Type	Description
Bugfix	Fixed a bug where an invalid rule caused the full ruleset to not be loaded

18.3.8 Aurora Agent Version 1.0.0

Type	Description
Bugfix	Fixed a bug where diagnostics pack creation could hang
Feature	Added specific 'registry_*' categories in log source mappings

18.4 Aurora Agent 0.9

18.4.1 Aurora Agent Version 0.9.9

Type	Description
Feature	Extended 'diagnostics' information to cover broken configurations better
Feature	Incorrect flags used in configuration file now cause Aurora startup to fail

18.4.2 Aurora Agent Version 0.9.8

Type	Description
Change	Upgraded to Golang 1.17
Feature	Added a new 'diagnostics' command to Aurora Util that extracts information useful for debugging

18.4.3 Aurora Agent Version 0.9.7

Type	Description
Bugfix	Fixed a bug where registry events weren't applied correctly
Bugfix	Fixed a bug where the TemporaryDriverLoadDetector did not contain useful information

18.4.4 Aurora Agent Version 0.9.6

Type	Description
Change	Improved formatting of events displayed by '--trace'
Change	Improved handling of custom responses with nonexistent fields
Change	Improved handling of invalid log source files

18.4.5 Aurora Agent Version 0.9.5

Type	Description
Feature	Added explicit information about enabled modules to '--module-info'

18.4.6 Aurora Agent Version 0.9.4

Type	Description
Bugfix	Fixed a bug where backslashes in custom responses were parsed incorrectly
Bugfix	Fixed a bug where events that indirectly originated from Aurora (e.g. via Sysmon) were processed
Bugfix	Fixed a bug where some response events had an incorrect log ID
Feature	Added IOC counts to status
Bugfix	Fixed a bug where explorer.exe could be terminated even if 'lowprivonly' was set

18.4.7 Aurora Agent Version 0.9.3

Type	Description
Change	Decreased time needed to stop Aurora
Bugfix	Fixed a bug where certain responses could lead to a deadlock in response execution
Bugfix	Fixed a bug where the log file was not recreated when it was deleted
Bugfix	Fixed a bug where '--restart-service' did not work as intended
Bugfix	Fixed a bug where faulty hash IOCs were silently ignored

18.4.8 Aurora Agent Version 0.9.2

Type	Description
Change	TCP log target now tries to reconnect if the connection is interrupted
Feature	Aurora Agent Util's 'upgrade' now also upgrades the installed agent when run with '--restart-service'

18.4.9 Aurora Agent Version 0.9.1

Type	Description
Change	Renamed providers to have similar naming patterns for all modules
Change	Changed indentation for '--status'
Bugfix	Fixed a bug where ProcessTree contained incorrect elements

18.4.10 Aurora Agent Version 0.9.0

Type	Description
Feature	Improved performance for many matching operations

18.5 Aurora Agent 0.8

18.5.1 Aurora Agent Version 0.8.3

Type	Description
Bugfix	Fixed a bug regarding decision making whether a process is considered high privileged

18.5.2 Aurora Agent Version 0.8.2

Type	Description
Change	Clarified log messages if responses are simulated
Change	Clarified log messages for IOC matches

18.5.3 Aurora Agent Version 0.8.1

Type	Description
Change	Renamed the 'FileAge' field to 'ImageAge' for many events
Feature	Improved debug logging
Feature	Added 'ParentCommandLine' field to some file events
Feature	Added information about grandparent process to process creation events
Feature	Added 'ProcessTree' field to process creation events

18.5.4 Aurora Agent Version 0.8.0

Type	Description
Change	The default locations for process exclude and false positive exclusion files have been moved to the new 'config/' directory
Change	The number of process excludes and false positive exclusions is now part of the agent status
Feature	Added 'exclude' command to Aurora Agent Util for a dialogue to exclude processes causing many events

18.6 Aurora Agent 0.7

18.6.1 Aurora Agent Version 0.7.0

Type	Description
Feature	Added '--process-exclude' parameter that can be used to filter out events from specific processes early
Bugfix	Fixed a bug that could potentially lead to deadlocks
Feature	Added ResControl module to terminate Aurora if memory usage is too excessive
Feature	Added information about events received per process to '--status --trace' output

18.7 Aurora Agent 0.6

18.7.1 Aurora Agent Version 0.6.4

Type	Description
Bugfix	Fixed a bug where some content information was missing from events

18.7.2 Aurora Agent Version 0.6.3

Type	Description
Change	Improved output for response execution

18.7.3 Aurora Agent Version 0.6.2

Type	Description
Change	Deprecated 'killparent' which was replaced by 'processidfield'
Feature	Added lookup of parent process using cached data for responses
Feature	Added 'emp' response action

18.7.4 Aurora Agent Version 0.6.1

Type	Description
Feature	Added 'processidfield' flag for responses using 'kill', 'suspend' or 'dump'
Change	Change '--deactivate-all-modules' to '--deactivate-all-consumers'
Feature	Added support for 'response: none' to explicitly overwrite a response with one that does nothing

18.7.5 Aurora Agent Version 0.6.0

Type	Description
Feature	Added '--response-set' flag for external definitions of responses for sigma rules
Bugfix	Fixed a bug where some events did not contain the process ID as expected by responses
Feature	Added 'all' as a valid value for the 'ancestors' flag

18.8 Aurora Agent 0.5

18.8.1 Aurora Agent Version 0.5.8

Type	Description
Change	Added descriptions for all modules

18.8.2 Aurora Agent Version 0.5.7

Type	Description
Feature	Added additional information for ASGARD's parameter representation
Change	Unified module list for Windows and Linux builds
Change	Included providers in '--module-list'

18.8.3 Aurora Agent Version 0.5.6

Type	Description
Change	Allowed deactivation of providers
Bugfix	Fixed an issue where some sigma rule matches were reported as Info level instead of Notice

18.8.4 Aurora Agent Version 0.5.5

Type	Description
Feature	Added '--quiet' flag for ASGARD
Feature	Added more log IDs for identification
Bugfix	Fixed a bug where '--restart-service' would fail if the Aurora service was stopped

18.8.5 Aurora Agent Version 0.5.4

Type	Description
Change	Improved identification of processes for correlation purposes

18.8.6 Aurora Agent Version 0.5.3

Type	Description
Change	Improved handling of allocations, reduced temporary allocations during event analysis

18.8.7 Aurora Agent Version 0.5.2

Type	Description
Feature	Added exclusions to intrusive tampering detectors
Feature	'--json' now also applies to eventlog output
Bugfix	Fixed a bug where Aurora Agent Util downloaded upgrades / updates even when not necessary

18.8.8 Aurora Agent Version 0.5.1

Type	Description
Feature	Added log source for 'WinEventLog:Microsoft-Windows-Windows Firewall With Advanced Security/Firewall'
Change	Removed unnecessary completion command in Aurora Agent Util

18.8.9 Aurora Agent Version 0.5.0

Type	Description
Feature	Added detection for 'EtwEventWrite' patches to process tampering detector
Bugfix	Fixed a bug where hash order was not constant

18.9 Aurora Agent 0.4

18.9.1 Aurora Agent Version 0.4.4

Type	Description
Change	Changed the scheduled task names to be better understandable
Feature	Added an additional log source for virtual disk mounts
Change	Administrator tokens now count as low privileged for 'lowprivonly' (only LOCAL SYSTEM and similar tokens are protected)

18.9.2 Aurora Agent Version 0.4.3

Type	Description
Bugfix	Fixed a bug where installation paniced in certain race conditions
Feature	Added better support for file names in events from 'Microsoft-Windows-Kernel-File'

18.9.3 Aurora Agent Version 0.4.2

Type	Description
Feature	Added 'Alert' and 'Notice' log levels to better distinguish internal error / info messages and matches
Bugfix	Fixed a bug where a handle was not correctly closed
Change	Improved error message when receiving a Sigma correlation rule
Change	Improved output when failing to parse the command line

18.9.4 Aurora Agent Version 0.4.1

Type	Description
Bugfix	Fixed a bug where Aurora installation timed out
Change	Improved output if Aurora service failed to start after installation
Bugfix	Fixed a bug where '--uninstall' failed when run from the installed Aurora executable
Bugfix	Fixed a bug where a segmentation fault in the eventlog API was visible to the user

18.9.5 Aurora Agent Version 0.4.0

Type	Description
Change	Startup errors when running as a service are now written to 'service-startup.log' next to the executable
Change	There are now two scheduled tasks: one for upgrades, one for updates
Feature	Added '--report-stats-verbose' flag for more information in '--report-stats' output
Bugfix	Fixed a bug where signatures where updated even when this was unnecessary
Change	Installation now adds the installation path to the PATH environment variable

18.10 Aurora Agent 0.3

18.10.1 Aurora Agent Version 0.3.0

Type	Description
Bugfix	Fixed a bug where Aurora indefinitely tried to restart after a startup error
Bugfix	Fixed a bug where the installed service still referred to the paths as they were prior to installation
Bugfix	Fixed a bug where Aurora didn't update the signatures daily
Change	Updated description for many flags in '--help'
Change	Process dumps are now written to the 'process-dumps' folder by default instead of the working directory
Feature	Added banner display for interactive runs
Feature	Added a default file for '--false-positive-filter' that includes a usage example
Feature	Added rule paths to '--status' output
Change	Specifying positional arguments (which were ignored before) now causes an error

18.11 Aurora Agent 0.2

18.11.1 Aurora Agent Version 0.2.4

Type	Description
Feature	Added support for DestinationIsIpv6 in Microsoft-Windows-TCPIP events
Change	Improved installation procedure to account for user interrupts
Feature	Added custom-signatures folder that is on the search list by default
Change	Improved handling of panics and runtime faults

18.11.2 Aurora Agent Version 0.2.3

Type	Description
Feature	Active and Inactive modules are listed at startup
Feature	Added more verbose output to installation success

18.11.3 Aurora Agent Version 0.2.2

Type	Description
Feature	Signature revision is now included in status and initial message
Change	Events from the named pipe poller now include the process that has a handle to the named pipe
Change	The named pipe polling provider now provides polling for all handles on the system
Change	Command lines from existing processes at Aurora startup are now properly cached

18.11.4 Aurora Agent Version 0.2.1

Type	Description
Bugfix	Fixed bug that caused the version numbers to be empty in Eventlog
Bugfix	Fixed overlaps with Event IDs of different modules (default ID 199)
Change	Lowered score of driver loads from System32 folder (TemporaryDriverLoadDetector)

18.11.5 Aurora Agent Version 0.2.0

Type	Description
Change	Disabled EtwCanary for x86 systems due to issues with Windows 10 x86
Bugfix	Fixed a bug where the prodcess tampering detector caused panics on Windows 7
Change	Errors in single sigma rules no longer cause the Aurora Agent startup to fail
Feature	Added '--false-positive-filter-file' for custom exclusions
Change	Aurora now installs all files to C:Program FilesAurora Agent and none to C:ProgramData
Feature	Added '--force' flag to Aurora Agent Util for forced upgrades
Feature	Aurora Agent Util is now installed and can be used to update the installed version directly
Feature	Aurora Agent now adds a daily update scheduled tasks on installation

18.12 Aurora Agent 0.1

18.12.1 Aurora Agent Version 0.1.12

Type	Description
Bugfix	Fixed a bug in Sigma matching that could cause false negatives
Change	Unified startup log lines into a single message
Feature	Added module for process tampering detection
Feature	Added module for temporary driver detection
Feature	Added '--deactivate-all-modules' for easier debugging
Feature	Added '--sigdev' option for Aurora Agent Util
Feature	Added module for IOC (filenames, domains, hashes, ...) application
Change	Renamed '--no-content-info' to '--no-content-enrichment'

18.12.2 Aurora Agent Version 0.1.11

Type	Description
Feature	Added an ETW Canary module that checks whether ETW events are received
Feature	Added content information via correlation to many events
Change	Restricted number of active responses to 2 for Aurora Agent Lite
Feature	Added FileAge field for content information
Feature	Added Aurora Signature pack, Aurora Signatures can be updated with Aurora Util

18.12.3 Aurora Agent Version 0.1.10

Type	Description
Feature	Added a whitelist as beaconhunter excludes
Bugfix	Fixed a bug where the UDP socket permanently broke down
Feature	Added more context information to beaconhunter messages
Change	Sigma can now be deactivated with '--deactivate-module Sigma'
Change	BeaconHunter no longer activates expensive event sources by default, but still uses them if others activate them
Change	Renamed '--no-hashes' to the more accurate '--no-content-info'

18.12.4 Aurora Agent Version 0.1.9

Type	Description
Feature	Added log id for status messages
Bugfix	Fixed a FP in LSASS dump check
Feature	Added more information for TCP connections

18.12.5 Aurora Agent Version 0.1.8

Type	Description
Change	Moved log source mappings to a separate file that is shared for all configurations
Bugfix	Fixed a bug where process information could be misinterpreted when a process ID was reused
Feature	Added more content information for PE files (version resource information)

18.12.6 Aurora Agent Version 0.1.7

Type	Description
Feature	Added registry kernel logger as default source, values and paths are now parsed correctly
Bugfix	Fixed a bug where process information was discarded too early
Bugfix	Fixed a bug where Aurora didn't register properly for kernel providers if it was terminated harshly

18.12.7 Aurora Agent Version 0.1.6

Type	Description
Feature	Added '--print-event-id' option
Bugfix	Fixed a bug where errors in other ETW sessions could affect Aurora

18.12.8 Aurora Agent Version 0.1.5

Type	Description
Feature	Added '--no-hashes' option
Bugfix	Fixed a race condition where log sources were not updated properly on sigma log source change
Bugfix	Fixed a bug where hash calculation didn't close its file mapping properly
Change	Log sources are now in a separate folder
Feature	Added four agent configurations (minimal, reduced, standard, intense) for common use cases
Change	Renamed 'sigma-config' to '--log-source'
Bugfix	Fixed a bug where debugging output from the imphash calculation was visible
Change	Disabled quick edit mode in a console while Aurora is running

18.12.9 Aurora Agent Version 0.1.4

Type	Description
Feature	Added MD5, SHA1, SHA256 hashes as well as imphashes to process creation, image load, and driver load events
Feature	Added Aurora Util for Aurora upgrades and rule encryption
Feature	Added example for proper named pipe detection using SystemLogger:Handle
Change	Expanded Log IDs, defined different Log ID ranges for the different modules

18.12.10 Aurora Agent Version 0.1.3

Type	Description
Change	Renamed '--event-throttling' to '--output-throttling', it now drops events instead of slowing Aurora
Bugfix	Fixed a bug where the log file wasn't written after installation
Feature	Added '--low-prio' for reduced process priority, changed default priority to normal
Change	Added '--sigma-match-throttling' and '--sigma-match-burst' for limiting sigma matches on a per-rule basis
Change	aurora-agent now calls aurora-agent-64 when called on a 64 bit platform
Feature	Added missing log source rewrite for systemlogger-process
Change	Grouped "source not found" messages
Change	Rules may now define multiple responses
Change	Event Log IDs are now equal to Sysmon Event IDs for common sigma categories
Change	Custom fields are now marshaled to YAML in string form
Change	CPU limit now measures only CPU usage of Aurora

18.12.11 Aurora Agent Version 0.1.2

Type	Description
Feature	Added '--event-throttling' option for slowed output
Feature	Added '--no-stdout' option for no logging to stdout
Feature	Added '--module-info' option to enumerate existing modules
Bugfix	Fixed a bug where some parameters weren't written to the installed config
Change	Expanded '--status' output
Feature	Added support for response options: recursive, ancestors, and simulate
Feature	Added output for simulated responses
Bugfix	Fixed a bug where Aurora could match events that it wrote itself
Bugfix	Fixed a bug where fields available for sigma matching and responses were inconsistent
Feature	Added Aurora Agent Icon

18.12.12 Aurora Agent Version 0.1.1

Type	Description
Feature	Added support for activating and deactivating single consumers
Change	Allowed query syntax with ETW channels to request only specific event IDs
Feature	Added build revision support

18.12.13 Aurora Agent Version 0.1.0

Type	Description
Major Release	Initial Release

CHAPTER

NINETEEN

INDICES AND TABLES

- search