

Java Agent Configurations (22.x.x)

JAVA Agent Configurations


Exported on 04/09/2021

Table of Contents

1	Java Agent Security Features Deployment (22.x.x)	4
1.1	Auto-reloading	4
2	Logging Configuration (22.x.x)	5
2.1	Overview	5
2.2	Setup Logging Format	5
2.2.1	List of Waratek properties for logging configuration	5
2.3	XML logging configuration.....	7
2.3.1	Examples	8
3	Logging Format (22.x.x)	10
3.1	CEF Message Format.....	10
3.1.1	Logging Devices	10
3.1.2	Device Vendor	11
3.1.3	Device Product	11
3.1.4	Device Version	11
3.1.5	Device Event	11
3.1.6	Event Name	11
3.1.7	Event Severity.....	12
3.1.8	Extensions	12
3.2	Syslog Format.....	14
3.2.1	Syslog Header Format	14
3.2.1.1	SP	14
3.2.1.2	PRI.....	14
3.2.1.3	VERSION.....	15
3.2.1.4	TIMESTAMP.....	15
3.2.1.5	HOSTNAME	16
3.2.1.6	APP-NAME.....	17
3.2.1.7	PROCID	17
3.2.1.8	MSGID	17
3.2.1.9	STRUCTURED-DATA	17
3.2.1.10	MSG.....	17
4	Log Message: Types and Examples (22.x.x)	18

4.1	Policy Summary Messages	18
4.2	Rule Lifecycle Messages.....	18
4.2.1	Load Rule.....	18
4.2.2	Link Rule	18
4.2.3	Unlink Rule	19
4.2.4	Unload Rule	19
4.2.5	Execute Rule.....	19
4.2.5.1	Execute ARMR Patch rule.....	19
4.3	Invalid ARMR Mod Messages.....	20
4.4	ARMR Events on Demand.....	20
4.5	Informational Messages.....	21

1 Java Agent Security Features Deployment (22.x.x)

 Please refer to the separate documentation for the **Waratek Proposed Directory Structure** for assistance with the steps below.

Security Features can be included in the content of `jvc.rules` file as follows:

```
-Dcom.waratek.rules.local=<path-to-waratek-agent-conf-folder>jvc.rules
```

Security patches (and ARMR security rules) can be loaded from a dedicated directory as follows:

```
-Dcom.waratek.rules.dir=<path-to-waratek-agent-conf-folder>
```

Once configured, the Waratek agent will load every patch/rule placed in the above directory.

1.1 Auto-reloading


The following flag can be added to `<absolute path to waratek agent>/conf_*/waratek.properties` file to ensure rules are auto-reloaded :

```
com.waratek.rules.autoreload=true
```



- Valid ARMR Security rules and Patch files must have a **.armr** file extension in the rules directory (files with other extension or no extension will be ignored)
- A syntax error in an individual file will result in that particular file being ignored.
- Other files in the directory will be loaded, provided they are free from syntax errors.
- Subdirectories of the rules directory will be ignored.

2 Logging Configuration (22.x.x)

 Please reference Waratek Proposed Directory Structure document for proposed location of log configuration files

2.1 Overview

Waratek allows the user to track down security events that occur when a security rule is triggered in the Java application. Each time a rule is triggered, an entry is written to the log file (unless logging has been turned off for that rule).

For example:

```
<10>1 2020-06-10T12:27:19.198+01:00 1-qa02 java 17097 - - CEF:0|ARMR:Path Traversal mod|Path Traversal mod|
2.2|Protect against relative and absolute path traversal attacks|Execute Rule|High|rt=Jun 10 2020
12:27:19.196 +0100 dvchost=1-qa02 procid=17097 outcome=success act=protect msg=Path Traversal attack
blocked path={"Path":"/home/spiracle/pathTraversal/testFilesParent/testFilesChild/../TestFile"}
metadata={"HeaderInfo":{"remoteAddr":"0:0:0:0:0:0:1", "requestURI":"/spiracle/FileServlet01", "sessionId":"3
767AF331E581A52923E6A274332EF72", "cookieNames":{"JSESSIONID":"3767AF331E581A52923E6A274332EF72", "CUSTOMER_U
UID":"05b7b9d7-2046-4014-b8c9-bc53c79790c5"}}}
<13>1 2020-06-17T15:42:50.264+01:00 1-qa02 java 12190 - - CEF:0|ARMR:TLS upgrade mod|Walter|2.2|forced TLS
on every connection|Execute Rule|Unknown|rt=Jun 17 2020 15:42:50.263 +0100 dvchost=1-qa02 procid=12190
outcome=success act=protect msg=TLS connection upgraded dst=0 SocketInfo={"port":0,"upgraded":true,"Support
edProtocols":["SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2"]}
```

In parallel with sending security log events to Elasticsearch, which are in turn consumed by the Management Console, the Waratek agent also has the option of logging to other different locations:

- locally, to a log file or series of rolled-over log files;
- to a remote Syslog server using either UDP or TCP protocols.


2.2 Setup Logging Format

1. Open the <absolute path to waratek agent>/conf_*/waratek.properties file.
2. Add the following Waratek properties and make an adjustment according to the real-world requirement.

2.2.1 List of Waratek properties for logging configuration

- `com.waratek.log.mode`: type of location for logging security events. Can be one of `local`, `remote` or `both`, indicating the location that the Waratek agent will choose to log.
- `com.waratek.log.host`: mandatory property when the `com.waratek.log.mode` property is set to `remote` or `both`. The value should adhere to the following syntax:

```
[tcp:]<ip_address|hostname>:<port>
```

 The default protocol is UDP. Please use the `tcp:` prefix to connect remotely via TCP protocol.

- `com.waratek.log.DomainNameHost`: specifies the fully qualified domain name (FQDN) for the hostname of host. The value can be either a hostname or an IP address.

- i**
1. When the provided hostname is not resolved, in other words, there's no DNS server to query on the network, the IP address of the selected interface will be used.
 2. When the contacted domain name host is not available. the hostname from local configuration (`/etc/hostname`) will be used.

- `com.waratek.log.file`: the security log file location. If this log file is not provided, security logging will be turned off.

- i** Note : the value for security log file location may be an absolute or relative path, for example either of the following values may be used to specify the log file :-

- `com.waratek.log.file=/opt/waratek/conf_1/security.log`
- `com.waratek.log.file=security.log`

- `com.waratek.log.file.maxsize`: specifies the maximum file size (with a margin of some KBs) of a security log file. As soon as the file reaches the maximum size, the next write will cause the log to roll-over to another file.
 - As an example if the security log file is configured with the name `events.log` then the rolled-over file name will be `events.log.1` and subsequently `events.log.2` will be the next name, and so forth.
 - The allowed formats of the flag `com.waratek.log.file.maxsize` are as follows :
 - `<number>KB`
 - `<number>MB`
 - `<number>GB`
 - the default is bytes if KB/MB/GB extension is not present
 - The default value of this flag is 10MB - soon after the file reaches the limit the security log file is either rotated, if rotation is not disabled, or truncated.
- `com.waratek.log.file.maxindex`: defines the maximum number of backed-up security log files that will be created. For `N=3`, the following log files will be created: `events.log`, `events.log.1`, `events.log.2` and `events.log.3`. Files are selected in a round-robin fashion.
 - The default value of this flag is 1, this means that a single backup file is created. When the log file size exceeds the limit, the backup file is removed, log file is moved to `<filename>.1` and a new log file is created.
 - To disable rotation, set the value of this flag to 0
- `com.waratek.log.file.rotatedaily`: specifies if the log file is rotated every 24hours, values accepted are true or false.
- `com.waratek.log.file.redaction` : this flag lists a comma separated list of CEF extension names which are never included in the CEF event messages produced by the Agent, see example below


 Here is a typical log message with no CEF extensions redacted

```
<10>1 2021-01-22T12:22:45.181Z l-dev java 5041 - - CEF:0|ARMR:Walter|Walter|2.2|
xss_detect|Execute Rule|High|rt=Jan 22 2021 12:22:45.181 +0000 dvchost=l-dev
procid=5041 appVersion=1 act=protect msg=XSSTest payload=<img foo
error='100'='100' /> httpSessionId=A7E2E39171952A19199E407DC1090746
taintSource=HTTP_SERVLET httpRequestUri=/spiracle/xssContextMatrix.jsp
httpCookies=JSESSIONID=A7E2E39171952A19199E407DC1090746 remoteIpAddress=127.0.0.1
```

If we use

`com.waratek.log.cef.redaction=payload,httpSessionId,remoteIpAddress,httpCookies,`
the output is altered with information redacted

```
<10>1 2021-01-22T12:22:45.181Z l-dev java 5041 - - CEF:0|ARMR:Walter|Walter|2.2|
xss_detect|Execute Rule|High|rt=Jan 22 2021 12:22:45.181 +0000 dvchost=l-dev
procid=5041 appVersion=1 act=protect msg=XSSTest taintSource=HTTP_SERVLET
httpRequestUri=/spiracle/xssContextMatrix.jsp
```

 Alternatively, `com.waratek.rules.log` can be used instead of `com.waratek.log.file`, but this is now deprecated and will be removed in a future release.

2.3 XML logging configuration

 Note: XML logging configuration is deprecated and will be removed in a future release

Additionally, logging can be configured using an XML file provided to the `com.waratek.log.properties` system property.

1. Go to the folder `<path to waratek agent>/config_*`.
2. Create a `logProps.xml` file in that folder.
3. Copy the following content into `logProps.xml`, and make required adjustments that meet your use case.

```
<logProps-array>
  <logProps>
    <logMode>LOCAL</logMode>
    <logFile>SECURITYLOG</logFile>
    <fileName>/var/opt/waratek/rules.log</fileName>
    <priorityLevel>DEBUG</priorityLevel>
    <maxFileSize>10MB</maxFileSize>
    <maxBackupIndex>1</maxBackupIndex>
  </logProps>
</logProps-array>
```

4. Open the `<absolute path to waratek agent>/config_*/<name of property file>` file.
5. Add the following flag.

```
com.waratek.log.properties=<absolute path to logProps.xml>
```

- If the security rules are triggered, there should be a log file in the directory specified by the `fileName` tag in the `logProps.xml` file.

The list of valid XML elements within the file:

- `<LogFile>`
It is mandatory and must be set to `SECURITYLOG`.
- `<LogMode>`
It is optional and defines the location type of security events. The `<LogMode>` property values that are available are:
LOCAL: Logs messages only locally
REMOTE: Logs messages only to the remote log server.
BOTH: Logs messages both locally and to the remote log server.
- `<FileName>`
This is the location of the actual security log. It is optional when the `<LogMode>` property is set to LOCAL or BOTH because the default value will be set automatically. Note : the location of security log , if given by relative path, will resolve to absolute location based on XML logging configuration file location.
- `<remoteHost>`
It is mandatory when the `LogMode` property is set to REMOTE or BOTH. The value of the `<remoteHost>` property should adhere to the following syntax:

```
[tcp:]<ip_address|hostname>:<port>
```

- `maxFileSize`
It specifies the maximum file size (with a margin of some KBs) that the active log file and back-up log files will have. As soon as the log file reaches the maximum file size, the next write to the file will cause the log to roll-over to the "rules.log.1" file and subsequent logging is continues in a new "rules.log" file.
- `maxBackupIndex`
It defines the maximum number of backed-up log files that will be created.

2.3.1 Examples

How to set up remote logging with UDP protocol:

```
<logProps-array>
<logProps>
<logMode>REMOTE</logMode>
<logfile>SECURITYLOG</logfile>
<remoteHost>syslog.example.com:514</remoteHost>
</logProps>
</logProps-array>
```

How to set up local and remote logging using the TCP protocol:

```
<logProps-array>  
<logProps>  
<logMode>BOTH</logMode>  
<logFile>SECURITYLOG</logFile>  
<fileName>/var/opt/waratek/rules.log</fileName>  
<remoteHost>tcp:192.168.0.10:1234</remoteHost>  
</logProps>  
</logProps-array>
```

3 Logging Format (22.x.x)

Waratek agents use the CEF format when logging security events, wrapped around in a Syslog header envelope. The joint format (Syslog + CEF) is always transmitted regardless of the destination (security log file, Syslog server, Elasticsearch), split by a white space (**SP**).

```
[SYSLOG PREFIX] SP [CEF MESSAGE]
```

3.1 CEF Message Format

CEF stands for *Common Event Format* - this format is used in every security message event created by agents. Waratek CEF messages follow the [CEF specification](#)¹ from ArcSight.

CEF :0	PIPE	Device vendor	PIPE	Device Product	PIPE	Device Version	PIPE	Device Event	PIPE	Event Name	PIPE	Event Severity	PIPE	Extensions
--------	------	---------------	------	----------------	------	----------------	------	--------------	------	------------	------	----------------	------	------------

3.1.1 Logging Devices

There are 2 types of devices logging security messages:

- **ARMR mod:** For specific ARMR mods log security events and, at a more granular level, an ARMR rule.
 - An example of a Load Rule event for an ARMR patch rule where the ARMR mod is named CVE-2017-10102:

```
CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|RegistryImpl_Skel|Load Rule|Low|rt=May 05 2020 15:02:23.053 +0100 dvchost=I-dev05 procid=46210 outcome=success
```

- **Waratek agent:** The Waratek agent itself acts as a logging device for all other security events not sent by an ARMR mod.
 - An example of new security policies applied.

```
CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|Reload Rules|Low|rt=Jul 03 2020 01:44:30.199 +0100 dvchost=I-dev05 procid=1130132 outcome=success msg=New ARMR policy has been applied
```

- An example of no security policies applied.

```
CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|Reload Rules|Low|rt=Jul 03 2020 01:44:30.172 +0100 dvchost=I-dev05 procid=1130132 msg=Waratek rules file '/tmp/armr.rules' defined in 'com.waratek.rules.local' does not exist or is inaccessible. No security rules were loaded!
```

- An example of invalid syntax in security log files.

¹ <https://community.microfocus.com/t5/ArcSight-Connectors/ArcSight-Common-Event-Format-CEF-Implementation-Standard/ta-p/1645557?attachment-id=68077>

```
CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|Syntax Error|Very-High|rt=Jul 03 2020 01:44:30.172 +0100 dvchost=I-dev05 procid=1130132 msg=duplicate app name detected in rules file. Offending app name: "SQL mod" reason=Error loading ARMR App at line 38 : app("SQL mod"):
```

3.1.2 Device Vendor

There are 2 types of device vendor:

- **ARMR mod:** The value is ARMR:<ARMR Mod Name>.
- **Waratek agent:** The value is ARMR:Waratek.

3.1.3 Device Product

- **ARMR mod** - the ARMR mod name, as an example if the mod is named CVE-2017-10102, then this will be the name printed in this field as follows:

```
CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|RegistryImpl_Skel|Load Rule|Low|rt=May 05 2020 15:02:23.053 +0100 dvchost=I-dev05 procid=46210 outcome=success
```

- **Waratek agent** - the Waratek agent name that is currently running with the JVM. As the names of Waratek product, the value can be one of Patch Agent, Secure Agent or Upgrade Agent.

```
CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|Syntax Error|Very-High|rt=Jun 30 2020 19:26:28.703 +0100 dvchost=I-dev05 procid=840923 msg=invalid location specifier, patch must contain only one of [call, callreturn, callsite, entry, error, exit, instruction, line, read, readreturn, readsite, write, writesite, writereturn] reason=Error loading ARMR App at line 20 : app("myapp"):
```

3.1.4 Device Version

- **ARMR mod:** The version of the ARMR mod as declared in the requires declaration
- **Waratek agent:** The version of the Waratek agent (19.0.1)

3.1.5 Device Event

The particular part of the device where the event occurred. In the case of :

- **ARMR mod:** this is set to the name of the rule that triggered the security event
- **Waratek agent:** this is simply set to Engine

3.1.6 Event Name

The name of the security event type

- **ARMR mod:** different event names in the context of ARMR rule lifecycle events and upon execution: Load Rule, Link Rule, Execute Rule, Unload Rule, Unlink Rule.

- **Waratek agent:**

- Syntax Error: when an ARMR mod with invalid syntax is not recognized by the agent:

```
CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|Syntax Error|Very-High|rt=Jun 30 2020 19:26:28.703 +0100 dvchost=I-dev05 procid=840923 msg=invalid location specifier, patch must contain only one of [call, callreturn, callsite, entry, error, exit, instructio
Secure Agent|19.0.1|Engine|Syntax Error|Very-High|rt=Jun 30 2020 19:26:28.703 +0100 dvchost=I-dev05 procid=840923 msg=invalid location specifier, patch must contain only one of [call, callreturn, callsite, entry, error, exit, instruction, line, read, readreturn, readsite, write, writesite, writereturn] reason=Error loading ARMR App at line 20 : app("myapp"): n, line, read, readreturn, readsite, write, writesite, writereturn] reason=Error loading ARMR App at line 20 : app("myapp"):
```

- ReLoad Rules: events related to loading/unloading rules in general, but not specifically associated with any ARMR mod:

```
CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|Reload Rules|Low|rt=Jun 30 2020 19:26:28.703 +0100 dvchost=I-dev05 procid=1356675 outcome=success msg=No ARMR policy is in effect
```

3.1.7 Event Severity

One of the five severity values defined by CEF: Unknown, Low, Medium, High, Very-High.

3.1.8 Extensions

An extension is a key-value pair, where these are separated by whitespace. To avoid ambiguity between different extensions and for security reasons some characters need to be escaped (=, \, \n, \r), as mentioned in the CEF specification.

The order of extensions is not guaranteed and these can appear shuffled in a future release. The following table lists all supported extensions and their usage scope.

Extension	Description
rt	<p>Time of the triggered event. It has the following format MM DD YYYY HH:mm:ss.SSS zzzz, as follows :</p> <ul style="list-style-type: none"> • MM <ul style="list-style-type: none"> • English language abbreviation for the month of the year with the first character in uppercase and the other two characters in lowercase (Jan, Feb, Mar, Jun) • DD <ul style="list-style-type: none"> • day of the month, e.g., (03, 12, 30, ...) • YYYY <ul style="list-style-type: none"> • full year, e.g., 1975 • HH <ul style="list-style-type: none"> • hour represented in a 24-hour format • mm <ul style="list-style-type: none"> • minutes from 00 to 59 • ss <ul style="list-style-type: none"> • seconds from 00 to 59 • SSS <ul style="list-style-type: none"> • milliseconds from 000 to 999 • zzzz <ul style="list-style-type: none"> • timezone comprised of a “+“/”-” sign and other four digits for the time difference to GMT: • first 2 digits indicate the hour difference • last 2 digits indicate the minute difference
dvchost	The hostname of the machine the Java process is running on. It is the hostname as seen from the local host configuration or the FQDN depending on the agent configuration
procid	The process ID is taken from the Operating System where the process is running
nodeid	The ID of the Java process relative to the Management Console. This is the id assigned by the Management Console to the agent upon registration
outcome	Value is either success or failure

Extension	Description
act	The action as defined in the ARM rule. Examples of actions are: protect, detect or allow
msg	User-defined message as declared in the ARM rule

i If unable to retrieve values for any of the extensions `dvchost`, `procid` or `nodeid` then these extensions are omitted from the CEF message.

The `rt` extension is mandatory in every CEF message logged.

3.2 Syslog Format

Syslog format wraps the actual log message, just like a network frame/datagram and it's widely used in enterprise IT. Unix systems have adopted this format for their internal logging events when they are logging to local files.

3.2.1 Syslog Header Format

PRI	SP	VERSION	SP	TIME STAMP	SP	HOSTNAME	SP	APP-NAME	SP	PROCID	SP	MSGID	SP	STRUCTURED-DATA	SP	MSG
-----	----	---------	----	------------	----	----------	----	----------	----	--------	----	-------	----	-----------------	----	-----

3.2.1.1 SP

Space Character

3.2.1.2 PRI

The PRI part must have three digits and will be bound with angle brackets as the first and last characters. The number contained within these angle brackets is known as the 'Priority' value and is computed by combining both the Facility and Severity.

Facility

The Waratek agent only uses a Facility of USER for its security logging.

Numerical Code	Facility
1	user-level messages

Severity

The following Syslog severities are used for the Waratek’s agent security logging. They map directly to the CEF message severity if the message payload is indeed in the CEF format. For other message types, the Syslog severity of Notice is chosen.

Syslog Numerical Code	Syslog Severity	CEF Severity
1	Alert: action must be taken immediately	Very-High
2	Critical: critical conditions	High
4	Warning: warning conditions	Medium
5	Notice: normal but significant condition	Unknown
6	Informational: informational messages	Low

The Priority value is calculated by first multiplying the Facility number by 8 and then adding the numerical value of the Severity. For example, a user-level message (Facility=1) with a Severity of Notice (Severity=5) would have a Priority value of 13.

3.2.1.3 VERSION

The VERSION field denotes the version of the Syslog protocol specification. VERSION equals '1' indicating that the standard documented by [RFC 5424](https://tools.ietf.org/html/rfc5424)² is being followed.

3.2.1.4 TIMESTAMP

The TIMESTAMP is of the following format :

DATE-FULL-YEAR-DATE-MONTH-DATE-MDAYTTIME-HOUR:TIME-MINUTE: TIME-SECOND.TIME-MILLISECOND[Z]
 [+/-UTC-TIME-HOUR: UTC-TIME-MINUTE]

The TIMESTAMP value must follow these restrictions:

- The "T" and "Z" characters in this syntax must be upper case.
- Usage of the "T" character is required.
- Leap seconds must not be used.
- DATE-FULL-YEAR
 - 4 digits.
- DATE-MONTH
 - 2 digits; 01-12
- DATE-MDAY
 - 2 digits; 01-28, 01-29, 01-30, 01-31 based on month and year.

² <https://tools.ietf.org/html/rfc5424#page-8>

- TIME-HOUR
 - 2 digits; 00-23
- TIME-MINUTE
 - 2 digits; 00-59
- TIME-SECOND
 - 2 digits; 00-59
- TIME-MILLISECOND
 - 3 digits.
- UTC-TIME-HOUR
 - timezone difference from UTC; 2 digits; 00-23
- UTC-TIME-MINUTE
 - timezone difference from UTC; 2 digits; 00-59

Examples

Example 1

1985-04-12T23:20:50.52Z

This represents 20 minutes and 50.52 seconds after the 23rd hour of 12 April 1985 in UTC.

Example 2

1985-04-12T19:20:50.52-04:00

This represents the same time as in example 1 but expressed in US Eastern Standard Time (observing daylight savings time).

Example 3

2003-10-11T22:14:15.003Z

This represents 11 October 2003 at 10:14: 15 pm, 3 milliseconds into the next second. The timestamp is in UTC. The timestamp provides millisecond resolution. The creator may have actually had a better resolution, but providing just three digits for the fractional part of a second does not tell us.

Example 4

2003-08-24T05:14:15.003-07:00

This represents 24 August 2003 at 05:14: 15 am, 3 microseconds into the next second. The timestamp indicates that its local time is -7 hours from UTC. This timestamp might be created in the US Pacific time zone during daylight savings time.

Example 5: An Invalid TIMESTAMP

2003-08-24T05:14:15.000000003-07:00

This example is nearly the same as Example 4, but the second is followed by nanoseconds which is invalidate.

3.2.1.5 HOSTNAME

The HOSTNAME field identifies the machine that originally sent the Syslog message or – if not available.

3.2.1.6 APP-NAME

The APP-NAME field should identify the device or application that originated the message. It is intended for filtering messages on a relay or collector.

In the Waratek agent, the APP-NAME value is simply java.

3.2.1.7 PROCID

The PROCID field is often used to map log entries to the process from where they were logged. The Waratek agent prints the corresponding process ID from the Operating System. If the process ID cannot be identified, a random one is generated and used for as long as the process is alive.

3.2.1.8 MSGID

The hyphen (-) is used for all Waratek Agents.

3.2.1.9 STRUCTURED-DATA

The hyphen (-) is used for all Waratek Agents.

3.2.1.10 MSG

The MSG field is, in fact, the CEF message. Examples:

```
<13>1 2020-05-11T19:02:53.188+01:00 I-dev05 java 394700 - - CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|
Reload Rules|Low|rt=May 11 2020 19:02:53.188 +0100 dvchost=I-dev05 procid=394700 outcome=success msg=No
ARMR policy is in effect
```

```
<9>1 2020-05-11T14:39:23.965+01:00 I-dev05 java 433109 - - CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|
RegistryImpl_Skel|Execute Rule|Very-High|msg=java.lang.Exception outcome=failure procid=433109 dvchost=I-
dev05 rt=May 11 2020 14:39:23.965 +0100
```

```
<14>1 2020-07-07T16:39:52.540+01:00 I-dev05 java 1398713 - - CEF:0|ARMR:Mod with filesystem rule|Mod with
filesystem rule|2.2|block file.txt|Load Rule|Low|rt=Jul 07 2020 16:39:52.538 +0100 dvchost=I-dev05
procid=1398713 nodeid=1 outcome=success
<14>1 2020-07-07T16:39:52.558+01:00 I-dev05 java 1398713 - - CEF:0|ARMR:Mod with filesystem rule|Mod with
filesystem rule|2.2|block file.txt|Link Rule|Low|rt=Jul 07 2020 16:39:52.557 +0100 dvchost=I-dev05
procid=1398713 nodeid=1 outcome=success
```

4 Log Message: Types and Examples (22.x.x)

4.1 Policy Summary Messages

The Waratek agent will notify the user whether new ARMR policies have or have not been applied to the application. This message type is complementary to the Rule Lifecycle messages.

When a new ARMR policy is applied:

```
<14>1 2020-07-06T23:35:36.393+01:00 I-dev05 java 1356675 - - CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|
Reload Rules|Low|rt=Jul 06 2020 23:35:36.393 +0100 dvchost=I-dev05 procid=1356675 outcome=success msg=New
ARMR policy has been applied
```

When ARMR policies have been cleared or there was no ARMR policies to load by the agent:

```
<14>1 2020-07-06T23:36:06.405+01:00 I-dev05 java 1356675 - - CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|
Reload Rules|Low|rt=Jul 06 2020 23:36:06.405 +0100 dvchost=I-dev05 procid=1356675 outcome=success msg=No
ARMR policy is in effect
```

The logging device of this type of messages is the Waratek agent.

4.2 Rule Lifecycle Messages

Rule lifecycle messages occur when the Waratek agent applies any ARMR rule. They are labelled as Load Rule, Link Rule, Unload Rule, Unlink Rule and Execute Rule CEF events.

4.2.1 Load Rule

A rule is parsed and validated correctly:

```
<9>1 2020-05-11T19:02:22,551+01:00 I-dev05 java 46259 - - CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|
RegistryImpl_Skel|Load Rule|Low|rt=May 05 2020 15:02:23.053 +0100 dvchost=I-dev05 procid=46210
outcome=success
```

4.2.2 Link Rule

A rule is added to the runtime and a hook is created for the rule to execute. After this event, if all the conditions laid out in the rule are satisfied, the rule will execute the configured action:

```
<9>1 2020-05-11T19:02:22,551+01:00 I-dev05 java 46259 - - CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|
RegistryImpl_Skel|Link Rule|Low|rt=May 05 2020 15:02:28.257 +0100 dvchost=I-dev05 procid=46259
outcome=success
```

4.2.3 Unlink Rule

The rule's hook attached to the runtime is removed. Events for this rule will no longer trigger:

```
<9>1 2020-05-11T19:02:22,551+01:00 I-dev05 java 46259 - - CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|
RegistryImpl_Skel|Unload Rule|Low|rt=May 05 2020 15:02:37.098 +0100 dvchost=I-dev05 procid=46259
outcome=success
```

4.2.4 Unload Rule

A rule is completely removed from the rules engine and the system:

```
<9>1 2020-05-11T19:02:22,551+01:00 I-dev05 java 46259 - - CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|
RegistryImpl_Skel|Unlink Rule|Low|rt=May 05 2020 15:02:37.094 +0100 dvchost=I-dev05 procid=46259
outcome=success
```

4.2.5 Execute Rule

Upon execution, an ARMR rule logs a security event indicating what action was taken in the face of the security threat. The act extension shown in the log entry is as what is configured for the rule along with its configured custom message (msg extension). Additional metadata associated with the context of the attack is also provided in some cases. This log message type uses the Execute Rule event type exclusively.

The following is a log entry example of an XSS attack that was blocked by an http rule:

```
<9>1 2020-07-03T12:50:17.738+01:00 l-qa02 java 17023 - - CEF:0|ARMR:XSS Mod|XSS Mod|2.2|XSS|Execute Rule|
Very-High|rt=Jul 03 2020 12:50:17.738 +0100 dvchost=l-qa02 procid=17023 outcome=success act=protect msg=XSS
attacked identified and blocked payload=<script>alert(1)</script> metadata={"HeaderInfo":{"remoteAddr":"0:0:
0:0:0:0:1","requestURI":"/spiracle/xss.jsp","sessionId":"1648908FE92AE9F6FF13691985B3B849","cookieNames":
{"CUSTOMER_UUID":"05b7b9d7-2046-4014-b8c9-bc53c79790c5"}}
```

4.2.5.1 Execute ARMR Patch rule

Apart from the ARMR patch rule, all ARMR rules will create a log event when they execute, unless logging has been intentionally turned off. The patch rule will only log a message if its code block has to be rolled back due to a problem that occurred during execution. The following example shows a log entry generated by a triggered patch rule in which an exception was not caught, hence, as a result, the rule was deactivated:


```
<9>1 2020-05-11T14:39:23.965+01:00 I-dev05 java 433109 - - CEF:0|ARMR:CVE-2017-10102|CVE-2017-10102|2.2|
RegistryImpl_Skel|Execute Rule|Very-High|msg=java.lang.Exception outcome=failure procid=433109 dvchost=I-
dev05 rt=May 11 2020 14:39:23.965 +0100
```

 The logging device of this type of messages is the corresponding ARMR mod.

4.3 Invalid ARMR Mod Messages

Just before ARMR mods are linked and loaded they are parsed and verified for syntax errors. A log message of this type is created in the case of an unexpected syntax. The event type used in this case is `Syntax Error` and the location of the error in the policy file is also shown.

```
<9>1 2020-07-07T22:18:33.383+01:00 I-dev05 java 1439521 - - CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|
Syntax Error|Very-High|rt=Jul 07 2020 22:18:33.377 +0100 dvchost=I-dev05 procid=1439521 msg=at line 15, col
1: extraneous input '' expecting {'endapp', IDENTIFIER} reason=Error loading ARMR App at line 14 :
app("CVE-2013-1537"):
```

 These types of message are always logged with a `Very-High` severity as no ARMR mods can load before the syntax is corrected. The logging device of this type of messages is the Waratek agent.

4.4 ARMR Events on Demand

Additionally, log events can be created by using the `ArmrEvent` API. These calls are made from the code block section of the patch rule. The following is an ARMR mod with a patch rule illustrating the usage:

```
app("CVE-2013-1537"):
  requires(version: ARMR/2.2)

  patch("CVE-2013-1537_01"):
    function("sun/rmi/server/MarshalInputStream.<clinit>()V")
    write("sun/rmi/server/MarshalInputStream.useCodebaseOnlyProperty")
    code(language : java):
      public void patch(JavaFrame frame) {
        ArmrEvent event = ArmrEvent.load("Rule Notice", "Low");
        event.addExtension("msg", "logging from rule");
        event.commit();
      }
    endcode
  endpatch
```

The corresponding log entry:

```
<14>1 2020-07-07T22:19:08.528+01:00 I-dev05 java 1439667 - - CEF:0|ARMR:CVE-2013-1537|CVE-2013-1537|2.2|
CVE-2013-1537_01|Rule Notice|Low|rt=Jul 07 2020 22:19:08.527 +0100 dvchost=I-dev05 procid=1439667
msg=logging from rule
```

 The logging device of this type of messages is the corresponding ARMR mod.

4.5 Informational Messages

The agent would log events alerting the user while executing invalid or incorrect configurations. Generally, these log entries are logged using the `Reload Rules` event type.

When the ARMR policy file configured through the `com.waratek.rules.local` system property is removed from the filesystem and the auto-reload functionality is enabled:


```
<14>1 2020-07-07T22:42:07.528+01:00 I-dev05 java 1442522 - - CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|
Reload Rules|Low|rt=Jul 07 2020 22:42:07.527 +0100 dvchost=I-dev05 procid=1442522 msg=Waratek rules file '/
tmp/test.armr' defined in 'com.waratek.rules.local' does not exist or is inaccessible. No security rules
were loaded!
```

When using the `com.waratek.rules.dir` system property and the specified ARMR mods directory does not exist:

```
<10>1 2020-07-07T22:42:40.455+01:00 I-dev05 java 1442715 - - CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|
Reload Rules|High|rt=Jul 07 2020 22:42:40.449 +0100 dvchost=I-dev05 procid=1442715 msg=Configured rules
directory "/tmp/test.armr" does not exist
```

When using the `com.waratek.rules.dir` system property and extraneous files (`.armr` extension) are detected inside the specified ARMR mods directory:

```
<10>1 2020-07-07T22:42:40.455+01:00 I-dev05 java 1442715 - - CEF:0|ARMR:Waratek|Secure Agent|19.0.1|Engine|
Reload Rules|High|rt=Jul 07 2020 22:42:40.449 +0100 dvchost=I-dev05 procid=1442715 msg=Configured rules
directory contains unexpected file: /tmp/jvc.rules
```

 The logging device of this type of messages is the Waratek agent.