Common Event Rule Expression

Engineering Session

NIST

CERE

- Vision for the specification
- General Requirements
- Rule Types
- Data Exchange
- Example Rules
- Flexibility
- Content
- Content Production
- Content Protection

CERE Vision

- Provide vendors and consumers a way to express and share rules for pattern matching, correlation, and filtering of logs
  - Support distributed multi-vendor enterprises
  - Aid in acquisition
  - Simplify sharing detection rules to public
  - Achieve this with minimal impact to vendors and consumers

General Requirements

- Match based on Boolean combinations
  - AND, OR, NOT, XOR
- Temporal constraints
  - Ordering
    - Ordered sequences of events, or sets of events
    - Unordered sets of events
  - Time window
    - Fixed time window
    - Gradient time window

General Requirements

- State
  - Match based on previous events or current state
- Additionally query triggers
  - Ability to gather data from repositories
  - Ability to direct agents to gather additional data

Rules Types

- Filters (Common Event Filtering Expression)
  - Just another rule
  - Priority based filtering – filtering by criticality
  - Compression/Normalization – Combine identical events into a single event
  - Discarding – remove those events that aren't relevant
  - Time out – for time window correlation, remove those things that have aged out of consideration

Rule Types

- Rule based reasoning
  - Single event – a single event matches a criteria and events are processed in the stream on their own
  - Multi-event – a criteria is met when multiple events occur events are still treated independently, but correlated to other streams
  - Fixed threshold – a criteria is met when an event rate threshold is met or exceeded

Rule Types

- Ordered multi-stage chaining – a criteria is met when x condition follows y condition is met within z time period. Order is a factor

Data Exchange

- Modern SIEM products already have a native rules expression and processing capability
  - A rule interchange should not impact how products internally represent or process rules
  - Investigating the W3C Rule Interchange Format (RIF)
    - Designed for the purpose of exchanging rules
    - Reasonable momentum as a standard (accepted as a recommendation by W3C)
    - Is highly expressive and extensible

Data Exchange

- Doesn’t require creating a new expression from scratch

– There are also some drawbacks to RIF
  - Very early in development
  - Not much adoption yet
  - Very complex
  - Very generic

– Mitigations
  - Create a purpose-built dialect for the security event use case
  - Monitor adoption and continue research

Data Exchange

- There are other rule languages (RuleML, Drools)
- It may prove necessary or efficient to construct a new expression
  - would rather adopt a usable existing standard

Example Rules

- Examples from Open Source SIEM tool (OSSIM)

Single Event

```
directive id="3015" name="SQL injection attempt against DST_IP"priority="3">
  <rule type="detector" name="Sql injection attacker request" reliability="3"
  occurrence="1" from="ANY" to="ANY" port_from="ANY"
  port_to="ANY" plugin_id="SNORTRULES"
  plugin_sid="'snort: "ET WEB_SERVER Possible SQL Injection Attempt DELETE FROM"',
  'snort: "ET WEB_SERVER Possible SQL Injection Attempt INSERT INTO"'
  'snort: "ET WEB_SERVER Possible SQL Injection Attempt SELECT FROM"'
  'snort: "ET WEB_SERVER Possible SQL Injection Attempt UNION SELECT"',
  'snort: "ET WEB_SERVER Possible SQL Injection Attempt UPDATE SET"'' protocol="ANY">
    <rules>
    </rules>
  <rule type="detector" name="Sql error server response"
  reliability="+7" time_out="10" occurrence="1" from="1:DST_IP" to="1:SRC_IP" port_from="ANY"
  port_to="ANY" plugin_id="SNORTRULES" plugin_sid="5000006,5000007,5000008"
protocol="ANY"/>
                                </rules>
                    </rule>
```

Example Rules

- Examples from Open Source SIEM tool (OSSIM)

Multi Event

```
<directive id="24000" name="Doly Trojan" priority="5">
        <rule type="detector" name="Intrusion rule matched" reliability="2"
occurrence="1" from="ANY" to="ANY"  port_from="ANY" port_to="ANY"
plugin_id="SNORTRULES" plugin_sid="'BACKDOOR Doly 2.0 access','BACKDOOR
        Doly 1.5  server response'">
                <rules>
                 </rules>
        </rule>
        <rule type="detector" name="Rare but open dest port used"
reliability="+4" occurrence="1" from="1:SRC_IP"  to="1:DST_IP"
port_from="1:SRC_PORT" port_to="1:DST_PORT" plugin_id="SPADE"
plugin_sid="'Spade:
        Rare but open dest port used'">
                <rules>
                </rules>
        </rule>
```

Example Rules

Fixed Threshold

```
<directive id="3011" name="POP3 Bruteforce against SRC_IP" priority="3">
<rule type="detector" name="Bruteforce against " reliability="3"
   occurrence="1" from="ANY" to="ANY" port_from="ANY"
   port_to="ANY" plugin_id="SNORTRULES" plugin_sid="5000004" protocol="ANY">
      <rules>
         <rule type="detector" name="POP3 Bruteforce against SRC_IP"
reliability="+5" time_out="100" occurrence="5"
         from="1:SRC_IP" to="1:DST_IP" port_from="ANY" port_to="ANY"
         plugin_id="SNORTRULES" plugin_sid="1:PLUGIN_SID" sticky="true"
protocol="ANY">
            <rules>
               <rule type="detector" name="POP3 Bruteforce against
SRC_IP" reliability="+7" time_out="300"
occurrence="20" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" port_to="ANY"
plugin_id="SNORTRULES" plugin_sid="1:PLUGIN_SID" sticky="true" protocol="ANY">
                  <rules>
                     <rule type="detector" name="POP3 Bruteforce
against SRC_IP" reliability="+10" time_out="500" occurrence="50" from="1:SRC_IP" to="1:DST_IP"
port_from="ANY" port_to="ANY"
plugin_id="SNORTRULES" plugin_sid="1:PLUGIN_SID" sticky="true" protocol="ANY">
                     </rule>
```

Flexibility

- For a specification to be effective it needs be flexible enough to express all (or almost all) rules for patterns matching, correlation, and filtering
  - Feasibility still being studied
  - Many cases to be considered
  - Will being this generic prove impractical?
  - Need to identify MUST have cases and those that are less critical

Content

- What about the content?
  - Content is always a battle
  - In this case, content should be a distributed effort
    - Rules come from consumers, vendors, and organizations that produce guidance
    - Many organizations have such rules, but have no format in which to express them
    - Many products have “default” rules but no means to express them
    - The good news, compatibility with the specification means as you write a rule, you can share the content

Content Reduction

- What about lossiness (lost in translation)?
  - How do we ensure content reduction does not occur?
  - Who is responsible for ensuring content reduction does not occur?

Content Protection

- What if I DON'T want to share?
  - Content is proprietary
  - Content is classified
  - Content exposes vulnerability
  - Should the specification allow for encrypted content (does this even help)?
  - Variables appear necessary in general, do they help here?
  - What other cases of "protecting" content can we envision?

Summary

- A generic rules expression would assist in standardizing the event management space
- There are many existing efforts, and and vendor implementations
- To minimize impact and maximize information exchange a language suited to expression vs. execution is desirable
- There is still research and experimentation required