ArcSight SmartConnector Map
Files for fun and profit
Mike Weston, Software Designer VI

#HPProtect
ArcSight Connector Map files for fun and profit

Agenda

• Introduction
• Ranges
• Regular expressions
• Parser-like expressions (New!)
• Real world example
• Q&A
Connector Map Files: Introduction
“With great power comes great responsibility”

Voltaire
ArcSight Connector map files for fun and profit

What are map files?

- Map files operate on events after they are collected and parsed, but before they are sent to the destination, conditionally changing one or more event fields
- They are actual files, located on the connector itself
- There are several parts of the connector code that use map files:
  - “Classic” map files, which operate on events early in the event flow
  - AgentInfoAdder1 map files, which operate on events later in the event flow, and can be made to operate differently when there are multiple destinations and/or multiple connectors running in one container
  - The categorizer modules use map files to do their work
  - Map file “extra processors” can be specified in FlexConnector parsers
ArcSight Connector Map files for fun and profit

Simple example of a map file

- Normal comma-separated value text file:

<table>
<thead>
<tr>
<th>event.destinationPort</th>
<th>set.event.applicationProtocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>ftp</td>
</tr>
<tr>
<td>21</td>
<td>ftp</td>
</tr>
<tr>
<td>80</td>
<td>http</td>
</tr>
<tr>
<td>110</td>
<td>pop3</td>
</tr>
</tbody>
</table>

- Edit with a plain text editor or a spreadsheet program
- The first line defines the event fields that will be looked at ("getters") and those that will potentially be set ("setters")
ArcSight Connector Map files for fun and profit

The “real” format of a map file

- The map file on the previous slide would look like this in a text editor:

```plaintext
event.destinationPort, set.event.applicationProtocol
20, ftp
21, ftp
80, http
110, pop3
```
ArcSight Connector Map files for fun and profit

File naming and locations

• For “classic” map files:
  – Put the files in the user/agent/map directory under the ArcSight home directory
  – Name the files map.0.properties, map.1.properties, etc.
  – New or changed files will be picked up after about 5 minutes, or there’s a command to reload them

• For AgentInfoAdder1 map files:
  – Put the files in the user/agent/aup/acp directory under the ArcSight home directory
  – Or use the user/agent/aup/id/acp directory for destination/connector-specific files
  – Name the files AgentInfoAdder1.map.10.csv, AgentInfoAdder1.map.11.csv, etc.
ArcSight Connector Map files for fun and profit

The “no getter” trick

• Example:

```
set.event.message
Map file was here
```

• By having no “getters,” you can set one or more fields to specific constant values, unconditionally
• Such a map file always has exactly two lines
• Can have more than one column if you want to set more than one field
ArcSight Connector Map files for fun and profit

Without ranges

- Example:

<table>
<thead>
<tr>
<th>event.sourceAddress</th>
<th>set.event.flexString1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.1.0</td>
<td>China</td>
</tr>
<tr>
<td>1.0.1.1</td>
<td>China</td>
</tr>
<tr>
<td>1.0.1.2</td>
<td>China</td>
</tr>
<tr>
<td>1.0.1.3</td>
<td>China</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>1.0.3.255</td>
<td>China</td>
</tr>
</tbody>
</table>
ArcSight Connector Map files for fun and profit

With ranges

• Example:

<table>
<thead>
<tr>
<th>range.event.sourceAddress</th>
<th>set.event.flexString1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.1.0-1.0.3.255</td>
<td>China</td>
</tr>
</tbody>
</table>

• Ranges can also be used on:
  - Number event fields like sourcePort or fileSize
  - IPv6 event fields like deviceCustomIPv6Address1
  - MAC address event fields like destinationMacAddress
Connector Map Files: Regular expressions
ArcSight Connector Map files for fun and profit

With regular expressions

- Example:

<table>
<thead>
<tr>
<th>regex.event.sourceUserName</th>
<th>set.event.flexString1</th>
</tr>
</thead>
<tbody>
<tr>
<td>.<em>?arcsight.com.</em></td>
<td>ArcSight</td>
</tr>
<tr>
<td>.<em>?microsoft.com.</em></td>
<td>Microsoft</td>
</tr>
</tbody>
</table>
Connector Map Files: Parser-like expressions
ArcSight Connector Map files for fun and profit

Expression “setters” example: Inputs and output

• Here's three input events, and the resulting deviceCustomString1 values we want:

<table>
<thead>
<tr>
<th>deviceCustomNumber1</th>
<th>deviceCustomString1</th>
<th>deviceCustomString3</th>
<th>deviceCustomString1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“Leading and trailing”</td>
<td>“Whatever”</td>
<td>“Leading and trailing”</td>
</tr>
<tr>
<td>10</td>
<td>“Anyone reading this?”</td>
<td>“Overwrite with this”</td>
<td>“Overwrite with this”</td>
</tr>
<tr>
<td>17</td>
<td>“Hello...”</td>
<td>“...there!”</td>
<td>“Hello...”</td>
</tr>
</tbody>
</table>

Map file magic!
Expression “setters” example

- And here’s the map file that can do that:

| event.deviceCustomNumber1 | set.expr(deviceCustomString1|deviceCustomString3).event.deviceCustomString1 |
|--------------------------|--------------------------------------------------|
| 1                        | __stringTrim(deviceCustomString1)                |
| 10                       | deviceCustomString3                              |

- The “getter” column controls which row, if any, is used
- In the header line, the expression “setter” lists what event fields might be used in the expressions in that column, inside the parentheses, and what event field will be set, at the end
- Then one of the actual expressions below that is evaluated and the result put into the event field
- The operations (like __stringTrim above) that can be used can be found in appendix A of the FlexConnector Developer’s Guide
ArcSight Connector Map files for fun and profit

More about the expression “setter” syntax

• The expression “setter” header has several parts:
  – Two constant parts: “set.expr(" and ").event.”
  – Between those is the list of event fields and/or additional data fields that might be used in the expressions, separated by pipes (two pipes separate event fields from additional data)
  – And lastly the one event field that will be set to the result of the expression

• Here's a “no getter” example:

```python
set.expr(deviceCustomNumber1|deviceCustomNumber2||addnumber).event.deviceCustomNumber3
"__sum(deviceCustomNumber1,deviceCustomNumber2,__safeToInt(addnumber))"
```

• Note that the expression had to be in quotes since it contains commas
• In this case the deviceCustomNumber3 event field is set to the sum of the three fields
Connector Map Files: Real world example
ArcSight Connector Map files for fun and profit

Adding country names to events for Logger

- This comes from Aaron Kramer, and was posted on the Protect 724 site last year
- The idea is to augment events with new fields with the name of the source and destination countries, based on the sourceAddress and destinationAddress event fields
- The data divides the IPv4 address space into many ranges, each of which is associated with a particular country
- The map files are large enough (order of magnitude 100K lines) that you may need to increase the connector heap size
## ArcSight Connector Map files for fun and profit

### Adding country names to events

- The resulting map file would look something like this:

<table>
<thead>
<tr>
<th><code>range.event.sourceAddress</code></th>
<th><code>set.additionaldata.SCN</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0.0-1.0.0.255</td>
<td>Australia</td>
</tr>
<tr>
<td>1.0.1.0-1.0.3.255</td>
<td>China</td>
</tr>
<tr>
<td>1.0.4.0-1.0.7.255</td>
<td>Australia</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

- This just uses the range feature on an IPv4 event field
- A second map file with the same data is also needed for the destinationAddress event field
Connector Map Files: Q & A
For more information

Attend these sessions

• TB3044, Using Windows Event Forwarding with the Windows Unified Connector (Thurs 11am)

After the event

• Contact your sales rep

Your feedback is important to us. Please take a few minutes to complete the session survey.
Tonight’s party
@ Newseum

Time
7:00 – 10:00 pm

Shuttles run between hotel’s Porte Cochere (Terrace Level, by registration) and Newseum from 6:30 - 10:00 pm

Questions?
Please visit the Info Desk by registration

Enjoy food, drinks, company, and a private concert by Counting Crows
Please give me your feedback

Session TT3097 Speaker Mike Weston

Please fill out a survey.

Hand it to the door monitor on your way out.

Thank you for providing your feedback, which helps us enhance content for future events.
Thank you