SigmaToRML

|  |  |
| --- | --- |
| logsource:  product: windows  service: application  detection:  selection:  EventID: 524  Provider\_Name: Microsoft-Windows-Backup  condition: selection | // log source filter  logsource matches {product: 'windows', service: 'application'};  // event types  selection matches {provider\_name: 'Microsoft-Windows-Backup'};  // safe event type  safe\_selection not matches selection;  // proprty section  Main = logsource >> Monitor;  Monitor = safe\_selection\*; |
| logsource:  product: windows  service: 'security'  detection:  selection:  EventID:  - 4728 # or where  - 4729 # or where  - 4730  condition: selection | // log source filter  logsource matches {product: 'windows', service: 'security'};  // event types  selection matches {eventid: 4728 | 4729 | 4730};  // negation  no\_selection not matches selection;  // proprty section  Main = logsource >> Monitor;  Monitor = no\_selection\*; |
| logsource:  product: windows  service: security  detection:  selection:  EventID: 4663 # An attempt was made to access an object.  # This field must contain ALL of the following values  Accesses|all:  - 'ReadData'  - 'WriteData'  - 'DELETE'  condition: selection | // log source filter  logsource matches {product: 'windows', service: 'security'};  // event types  selection matches {eventid: 4663, accesses: ‘ReadData’&&‘'WriteData'&&‘'DELETE'};  // safe event type  safe\_selection not matches selection;  // proprty section  Main = logsource >> Monitor;  Monitor = safe\_selection\*; |
| logsource:  product: windows  service: 'security'  detection:  selection\_1:  EventID: 4663  selection\_2:  ObjectName: '\Device\CdRom0\setup.exe'  condition: selection\_1 and selection\_2 | // log source filter  logsource matches {product: 'windows', service: 'security'};  // event types  selection\_1 matches {eventid: 4663 };  selection\_2 matches {objectname: '\\Device\\CdRom0\\setup.exe'};  // safe event type  no\_selection\_1 not matches {eventid: 4663 };  no\_selection\_2 not matches {objectname: '\\Device\\CdRom0\\setup.exe'};  // proprty section  Main = logsource >> Monitor!;  Monitor = (no\_selection\_1 \/ no\_selection\_2)\*; |
| detection:  selection1:  eventSource: sts.amazonaws.com  eventName: AssumeRoleWithSAML  selection2:  eventSource: iam.amazonaws.com  eventName: UpdateSAMLProvider  condition: selection1 or selection2 | // log source filter  logsource matches {product: 'aws', service: 'cloudtrail'};  // event types  selection1 matches {eventsource: 'sts.amazonaws.com', eventname: 'AssumeRoleWithSAML'};  selection2 matches {eventsource: 'iam.amazonaws.com', eventname: 'UpdateSAMLProvider'};  // safe event type  safe\_selection1 not matches selection1;  safe\_selection2 not matches selection2;  // proprty section  Main = logsource >> Monitor!;  Monitor = (safe\_selection1 /\ safe\_selection2)\*; |
| detection:  selection:  Image: "werfault.exe"  filter1:  ParentImage: "svchost.exe"  filter2:  DestinationIp:  - "10.0.0.0"  - "172.16.0.0"  - "192.168.0.0"  filter3:  DestinationHostname:  - "windowsupdate.com"  - "microsoft.com"  condition: selection and not ( filter1 or filter2 or filter3 ) | // log source filter  logsource matches {product: 'windows', service: 'edr'};  // event types  selection matches {image: 'werfault.exe'};  filter1 matches {parentimage: 'svchost.exe'};  filter2 matches {destinationip: '10.0.0.0'|'172.16.0.0'|'192.168.0.0'};  filter3 matches {destinationhostname: 'windowsupdate.com'|'microsoft.com'};  // safe event type  safe\_selection not matches {image: 'werfault.exe'};  // proprty section  Main = logsource >> Monitor!;  Monitor = (safe\_selection \/ Filter)\*;  Filter = (filter1 \/ filter2 \/ filter3); |
| logsource:  product: gcp  service: audit  detection:  selection\_base:  eventService: admin.googleapis.com  eventName:  - ENFORCE\_STRONG\_AUTHENTICATION  - ALLOW\_STRONG\_AUTHENTICATION  selection\_eventValue:  new\_value: "false"  condition: all of selection\* | // log source filter  logsource matches {product: 'gcp', service: 'audit'};  // event types  selection\_base matches {eventservice: 'admin.googleapis.com', eventname:'ENFORCE\_STRONG\_AUTHENTICATION' | 'ALLOW\_STRONG\_AUTHENTICATION'};  selection\_eventValue matches {new\_value: 'false'};  // safe event type  safe\_selection\_base not matches {eventservice: 'admin.googleapis.com', eventname:'ENFORCE\_STRONG\_AUTHENTICATION' | 'ALLOW\_STRONG\_AUTHENTICATION'};  safe\_selection\_eventvalue not matches {new\_value: 'false'};  // proprty section  Main = logsource >> Monitor!;  Monitor = (safe\_selection\_base \/ safe\_selection\_eventvalue)\*; |
| logsource:  product: gcp  service: audit  detection:  selection\_base:  eventService: admin.googleapis.com  eventName:  - ENFORCE\_STRONG\_AUTHENTICATION  - ALLOW\_STRONG\_AUTHENTICATION  selection\_eventValue:  new\_value: "false"  condition: 1 of selection\* | // log source filter  logsource matches {product: 'gcp', service: 'audit'};  // event types  selection\_base matches {eventservice: 'admin.googleapis.com', eventname:'ENFORCE\_STRONG\_AUTHENTICATION' | 'ALLOW\_STRONG\_AUTHENTICATION'};  selection\_eventValue matches {new\_value: 'false'};  // negation  no\_selection\_base not matches {eventservice: 'admin.googleapis.com', eventname:'ENFORCE\_STRONG\_AUTHENTICATION' | 'ALLOW\_STRONG\_AUTHENTICATION'};  no\_selection\_eventvalue not matches {new\_value: 'false'};  // proprty section  Main = logsource >> Monitor!;  Monitor = (no\_selection\_base /\ no\_selection\_eventvalue)\*; |
| logsource:  product: windows  service: security  detection:  selection:  EventID: 4738  AttributeLDAPDisplayName: 'Min-Pwd-Length'  check\_value:  AttributeValue|lte: 7  condition: selection and check\_value | // log source filter  logsource matches {product: 'windows', service: 'security'};  // event types  selection matches {eventid: 4378, attributeldapdisplayname:'Min-Pwd-Length'};  check\_value matches {attributevalue: x} with x <= 7;  // negation  no\_selection not matches {eventid: 4378, attributeldapdisplayname:'Min-Pwd-Length'};  safe\_check\_value matches {attributevalue: x} with x > 7;  // proprty section  Main = logsource >> Monitor;  Monitor = (no\_selection \/ safe\_check\_value)\*; |
| logsource:  product: windows  service: security  detection:  selection\_task:  EventID: 4698  selection\_firewall:  EventID: 4946  timeframe: 5m  condition: selection\_task and selection\_firewall | // log source filter  logsource matches {product: 'windows', service: 'security'};  // event types  selection\_task(ts) matches {timestamp: ts, eventid: 4698};  selection\_firewall(ts) matches {timestamp: ts, eventid: 4946};  other\_events(ts) matches {timestamp: ts};  // proprty section  Main = logsource >> Monitor<0, 0, 0>!;  Monitor<start\_ts, s1, s2> =  {  let ts; selection\_task(ts) (  if (start\_ts == 0 || ts - start\_ts > 300000)  Monitor<ts, 1, 0>  else (  if (s2 == 1) empty else Monitor<start\_ts, 1, s2>  )  )  }  \/  {  let ts; selection\_firewall(ts) (  if (start\_ts == 0 || ts - start\_ts > 300000)  Monitor<ts, 0, 1>  else (  if (s1 == 1) empty else Monitor<start\_ts, s1, 1>  )  )  }  \/  {  let ts; other\_events(ts) (  if (start\_ts > 0 && ts - start\_ts > 300000)  Monitor<0, 0, 0>  else (  Monitor<start\_ts, s1, s2>  )  )  }; |
| logsource:  product: windows  service: security  detection:  selection\_priv\_add:  EventID: 4728  selection\_log\_clear:  EventID: 1102  condition: selection\_priv\_add | near selection\_log\_clear | // event types  selection\_priv\_add(ts) matches {timestamp: ts, eventid: 4728};  selection\_log\_clear(ts) matches {timestamp: ts, eventid: 1102};  other\_events(ts) matches {timestamp: ts};  // proprty section  Main = logsource >> Monitor<0, 0, 0>!;  Monitor<start\_ts, s1, s2> =  {  let ts; selection\_priv\_add(ts) (  if (start\_ts == 0 || ts - start\_ts > 10000)  Monitor<ts, 1, 0>  else (  if (s2 == 1) empty else Monitor<start\_ts, 1, s2>  )  )  }  \/  {  let ts; selection\_log\_clear(ts) (  if (start\_ts == 0 || ts - start\_ts > 10000)  Monitor<ts, 0, 1>  else (  if (s1 == 1) empty else Monitor<start\_ts, s1, 1>  )  )  }  \/  {  let ts; other\_events(ts) (  if (start\_ts > 0 && ts - start\_ts > 10000)  Monitor<0, 0, 0>  else (  Monitor<start\_ts, s1, s2>  )  )  }; |
| logsource:  product: windows  service: security  detection:  selection:  EventID: 4663  Accesses: 'DELETE'  timeframe: 1m  condition: selection | count() > 5 | // log source filter  logsource matches {product: 'windows', service: 'security'};  // event types  selection(ts) matches {timestamp: ts, eventid: 4663, accesses: 'DELETE'};  safe\_selection not matches {eventid: 4663, accesses: 'DELETE'};  other\_events(ts) matches {timestamp: ts};  // proprty section  Main = logsource >> Monitor<0, 0>!;  Monitor<start\_ts, count> =  {let ts; selection(ts)  (  if (start\_ts == 0 || ts - start\_ts > 60000) (  Monitor<ts, 1>  )  else if (count+1 > 4) safe\_selection else Monitor<start\_ts,count+1>  )  }  \/  {let ts; other\_events(ts)  (  if (start\_ts > 0 && ts - start\_ts > 60000)  Monitor<0, 0>  else  Monitor<count, start\_ts>  )  }; |
|  |  |