Rapid7_Nexpose_SecEvent template

Template	Comments
{ "version": "2.0", "name": "Rapid7 Nexpose Scan assets by security event", "comment": "", "type": "REST_EVENT", "event_type": ["RPZ", "TUNNEL"], "action_type": "Rapid7 Nexpose Scan assets by security event", "content_type": "text/xml", "vendor_identifier": "Rapid7", "quoting": "XMLA",	“version” must be set to “2.0” This template can be used with RPZ and TUNNEL events/notifications. XMLA quoting is used by default.

"steps":
[
 	
{
"name": "checkIPEAs",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{"left": "${E::ip.extattrs{R7_ScanOnEvent}}", "op": "==", "right": ""}
],
"next": "checkNetEAs"
}
},
	if R7_ScanOnEvent is not defined on the object level (if it is a lease or unmanaged IP) go to checkNetEAs step

{
"name": "checkIPScanOnEvent",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::ip.extattrs{R7_Site}}", "op": "==", "right": ""},
{"left": "${E::ip.extattrs{R7_ScanOnEvent}}", "op": "==", "right": "false"}
],
"stop": true
}
},
	Stop if R7_Site is not set or R7_ScanOnEvent set to “false”

{ "name": "setLIPVars", "operation": "NOP", "body_list": ["${XC:COPY:{L:source_ip}:{E:source_ip}}", "${XC:ASSIGN:{L:EASource}:{S:IP}}", "${XC:COPY:{L:Hostname}:{E:ip.names[0]}}", "${XC:ASSIGN:{L:SaveEA}:{S:false}}", "${XC:COPY:{L:Site}:{E:ip.extattrs{R7_Site}}}"] },	Set the local variables: **source_ip** - Source IP which triggered the event **EASource** - internal variable, defines object type **Hostname** - hostname of the host which triggered the event **SaveEA** - internal variable,

	defines if the extensible attributes can be updated **Site** - Site name in Rapid7 Nexpose

{
"name": "setIPSiteID",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::ip.extattrs{R7_SiteID}}", "op": "==", "right": ""}
],
"eval": "${XC:ASSIGN:{L:SiteID}:{I:0}}",
"else_eval": "${XC:COPY:{L:SiteID}:{E:ip.extattrs{R7_SiteID}}}"
}
},
{
"name": "setIPLastScan",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::ip.extattrs{R7_LastScan}}", "op": "==", "right": ""}
],
"eval": "${XC:ASSIGN:{L:LastScan}:{S:}}",
"else_eval": "${XC:COPY:{L:LastScan}:{E:ip.extattrs{R7_LastScan}}}"
}
},
{
"name": "setIPScanTemplate",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::ip.extattrs{R7_ScanTemplate}}", "op": "==", "right": ""}
],
"eval": "${XC:ASSIGN:{L:ScanTemplate}:{S:default}}",
"else_eval": "${XC:COPY:{L:ScanTemplate}:{E:ip.extattrs{R7_ScanTemplate}}}"
}
},
{
"name": "setIPAddByHostname",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::ip.extattrs{R7_AddByHostname}}", "op": "==", "right": ""}
],
"eval": "${XC:ASSIGN:{L:AddByHostname}:{S:false}}",
"else_eval": "${XC:COPY:{L:AddByHostname}:{E:ip.extattrs{R7_AddByHostname}}}"
	Set local variables based on EAs values:
	SiteID - Rapid7 internal Site ID
	LastScan - defines when the asset was scanned last time
	ScanTemplate - defines a scan template, if EA was not defined, default parameters are used for the scan
	AddByHostname - defines if a host should be scanned by a hostname

```
      }
    },
```

```
    {
      "name": "checkNetView",
      "operation": "CONDITION",
      "condition": {
        "condition_type": "OR",
        "statements": [
            {"left": "${E::network.network_view}", "op": "==", "right": ""}
        ],
        "next": "assignScanVars",
        "else_eval": "${XC:COPY:{L:network_view}:{E:network.network_view}}"
      }
    },
```

check if Network View is not exists go to assignScanVars. if it is exists set **network_view** local variable

```
    {
       "name": "Get IPv4Fixed _ref",
       "operation": "GET",
       "transport": {"path":
"fixedaddress?ipv4addr=${L:U:source_ip}&network_view=${L:U:network_view}"
},
       "wapi": "v2.6"
    },

    {
       "operation": "CONDITION",
       "name": "wapi_response_getIPv4Fix_ref",
       "condition": {
          "statements": [
             {"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}
          ],
          "condition_type": "AND",
          "next": "Get_Objref"
       }
    },

    {
       "name": "Get HostIPv4 _ref",
       "operation": "GET",
       "transport": {"path":
"record:host?ipv4addr=${L:U:source_ip}&network_view=${L:U:network_view}"},
       "wapi": "v2.6"
    },

    {
       "operation": "CONDITION",
       "name": "wapi_response_getIPv4Host_ref",
       "condition": {
          "statements": [
             {"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}
          ],
          "condition_type": "AND",
          "next": "Get_Objref"
       }
    },


    {
```

RPZ and TUNNEL events do not contain object reference. The code is trying to find/guess the object reference ID in the IPAM DB.

```
      "name": "Get IPv6Fixed _ref",
      "operation": "GET",
      "transport": {"path":
"ipv6fixedaddress?ipv4addr=${L:U:source_ip}&network_view=${L:U:network_view}"},
      "wapi": "v2.6"
    },

    {
      "operation": "CONDITION",
      "name": "wapi_response_getIPv6Fix_ref",
      "condition": {
        "statements": [
          {"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}
        ],
        "condition_type": "AND",
        "next": "Get_Objref"
      }
    },

    {
      "name": "Get HostIPv6 _ref",
      "operation": "GET",
      "transport": {"path":
"record:host?ipv6addr=${L:U:source_ip}&network_view=${L:U:network_view}"},
      "wapi": "v2.6"
    },

    {
      "operation": "CONDITION",
      "name": "wapi_response_getIPv6Host_ref",
      "condition": {
        "statements": [
          {"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}
        ],
        "condition_type": "AND",
        "next": "Get_Objref"
      }
    },
```

```
    {
      "name": "Get_Objref",
      "operation": "CONDITION",
      "condition": {
        "statements": [
          {"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}
        ],
        "condition_type": "AND",
        "eval":
"${XC:COPY:{L:Obj_ref}:{P:PARSE[0]{_ref}}}${XC:ASSIGN:{L:SaveEA}:{S:true}}"
      }
    },
```

If the previous steps were able to identify an object reference, set **Obj_ref** and **SaveEA** variables in order to be able to update R7_LastScan attribute

```
    {
      "name": "CheckIfHost",
      "operation": "CONDITION",
      "condition": {
        "statements": [
```

If the object is a host set **EASource** variable to HOST.

{"left": "${L::Obj_ref}", "op": "=~", "right": "record:host"}], "condition_type": "AND", "eval": "${XC:ASSIGN:{L:EASource}:{S:HOST}}" } },	

{
"name": "goToSiteIDcheck",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "", "op": "==", "right": ""}
],
"next": "assignScanVars"
}
},
	Go to assignScanVars step (skipping steps if there were no EAs on the object level)

{
"name": "checkNetEAs",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::network.extattrs{R7_ScanOnEvent}}", "op": "==", "right": ""},
{"left": "${E::network.extattrs{R7_ScanOnEvent}}", "op": "==", "right": "false"}
],
"stop": true
}
},
	Stop execution if **R7_ScanOnEvent** does not exists or set to false

{
"name": "setLNetVars",
"operation": "NOP",
"body_list": [
"${XC:COPY:{L:source_ip}:{E:source_ip}}",
"${XC:COPY:{L:Site}:{E:network.extattrs{R7_Site}}}",
"${XC:ASSIGN:{L:LastScan}:{S:}}",
"${XC:ASSIGN:{L:EASource}:{S:Net}}",
"${XC:ASSIGN:{L:SaveEA}:{S:false}}",
"${XC:ASSIGN:{L:Hostname}:{S:}}",
"${XC:ASSIGN:{L:AddByHostname}:{S:false}}"
]
},
 	
{
"name": "setNetSiteID",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::network.extattrs{R7_SiteID}}", "op": "==", "right": ""}
],
"eval": "${XC:ASSIGN:{L:SiteID}:{I:0}}${XC:ASSIGN:{L:LastScan}:{S:}}",
"else_eval": "${XC:COPY:{L:SiteID}:{E:network.extattrs{R7_SiteID}}}"
}
	Set the local variables (for the variable description see **setLIPVars** step)

},
 	
{
"name": "setNetScanTemplate",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${E::network.extattrs{R7_ScanTemplate}}", "op": "==", "right": ""}
],
"eval": "${XC:ASSIGN:{L:ScanTemplate}:{S:default}}",
"else_eval": "${XC:COPY:{L:ScanTemplate}:{E:network.extattrs{R7_ScanTemplate}}}"
}
},	

{ "name": "assignScanVars", "operation": "NOP", "body_list": ["${XC:COPY:{L:ScanDate}:{UT:TIME}}${XC:FORMAT:TRUNCATE:{L:ScanDate}:{10t}}", "${XC:COPY:{L:R7ScanSchTime}:{UT:EPOCH}}${XC:FORMAT:DATE_STRFTIME:{L:R7ScanSchTime}:{%Y%m%dT%H%M59000Z}}"] },	Set local variables: **ScanDate** is used as a value for R7_LastScan attribute **R7ScanSchTime** is used as a scheduled scan time in Rapid7 Nexpose API call

{
"name": "checkIFScannedToday",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{"left": "${L::LastScan}", "op": "==", "right": "${L::ScanDate}"}
],
"stop": true
}
},
	Stop If the asset was scanned today

{
"name": "Check SiteID",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{"left": "${L:A:SiteID}", "op": "!=", "right": "0"}
],
"next": "Create a schedule"
}
},
	If SiteID set jump to “Create a schedule” step

{ "name": "Request R7 sites", "parse": "XMLA", "operation": "POST", "body_list": [	The code (from this step to “Create a schedule”) is executed if R7_SiteID attribute was not set and it tries to

```
        "<?xml version=\"1.0\" encoding=\"UTF-8\"?>",
        "<SiteListingRequest session-id=\"${S::SESSID}\" />"
      ]
    },

    {
      "name": "Check sites request on errors",
      "operation": "CONDITION",
      "condition": {
        "statements": [
          {"left": "${P:A:PARSE[[name]]}", "op": "!=", "right":
"SiteListingResponse"},
          {"left": "${P:A:PARSE{{success}}}", "op": "!=", "right": "1"}
        ],
        "condition_type": "AND",
        "else_eval": "${XC:COPY:{L:site_list}:{P:PARSE}",
        "error": true
      }
    },

    {
      "name": "Check if sites list is empty",
      "operation": "CONDITION",
      "condition": {
        "statements": [
          {"left": "${L:L:site_list}", "op": "==", "right": "0"}
        ],
        "condition_type": "AND",
        "stop": true
      }
    },

    {
       "name": "Pop site from the list",
       "operation": "VARIABLEOP",
       "variable_ops": [
         {
           "operation": "POP",
           "type": "COMPOSITE",
           "destination": "L:a_site",
           "source": "L:site_list"
         }
      ]
    },

    {
      "name": "check_a_site",
      "operation": "CONDITION",
      "condition": {
        "statements": [
         {"left": "${L:A:Site}", "op": "!=", "right": "${L:A:a_site{{name}}}"}
        ],
        "condition_type": "AND",
        "next": "Check if sites list is empty",
        "else_eval": "${XC:COPY:{L:SiteID}:{L:a_site{{id}}}}"
      }
    },

    {
```

determinate **SiteID** base on **Site** name

SiteListingRequest is used to retrieve a list of sites from Rapid 7 Nexpose

In a loop a single value is retrieved from the list and compared with the **Site** attribute.
If the Site was found and **SaveEA** set to true SiteID attribute saved in R7_SiteID attribute and jumps to “Create a schedule”.

Stop if the Site was not found.

"name": "checkSaveSiteID",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{"left": "${L::SaveEA}", "op": "!=", "right": "true"}
],
"next": "Create a schedule"
}
},
 	
{
"name": "Update SiteID",
"operation": "PUT",
"transport": {"path": "${L:A:Obj_ref}"},
"wapi": "v2.6",
"wapi_quoting": "JSON",
"body_list": [
"{",
"\"extattrs+\":{\"R7_SiteID\": { \"value\": \"${L:A:SiteID}\"}}",
"}"
]
 	
},	

{ "name": "Create a schedule", "operation": "SERIALIZE", "serializations": [{"destination": "L:R7ScanSch","content": "<Schedules><AdHocSchedule start=\"${L:A:R7ScanSchTime}\" template=\"${L:A:ScanTemplate}\" /></Schedules>"}, {"destination": "L:R7ScanByHost","content": "<Hosts><host>${L:A:Hostname}</host></Hosts>"}, {"destination": "L:R7ScanByIP","content": "<Hosts><range from=\"${L:A:source_ip}\"/></Hosts>"}] },	XML templates are created for an API request: **R7ScanSch** - contains a schedule with a scan template **R7ScanByHost** - contains a target hostname to scan **R7ScanByIP** - contains a target IP-address to scan

{
"name": "scanByHostname",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{"left": "${L::AddByHostname}", "op": "==", "right": "true"},
{"left": "${L::Hostname}", "op": "!=", "right": ""},
{"left": "${L::EASource}", "op": "==", "right": "HOST"}
],
"eval": "${XC:COPY:{L:R7ScanHostsRanges}:{L:R7ScanByHost}}",
"else_eval": "${XC:COPY:{L:R7ScanHostsRanges}:{L:R7ScanByIP}}"
}
},
	if an event was triggered by a host which was added to Rapid7 Nexpose by a hostname and a hostname is exists use **R7ScanByHost** as a scan target, otherwise use **R7ScanByIP**

{
"name": "skipSchedule",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
	“default” is a fake scan template name. If a “default” scan was requested we do not add a schedule section into

<table>
<tr>
<td>

```
      "statements": [
        {"left": "${L::ScanTemplate}", "op": "==", "right": "default"},
        {"left": "${L::ScanTemplate}", "op": "==", "right": ""}
      ],
      "eval": "${XC:ASSIGN:{L:R7ScanSch}:{S:}}"
    }
  },
```

</td>
<td>the API request. Default parameters defined for a Site in Rapid7 Nexpose will be used</td>
</tr>
<tr>
<td>

```
  {
    "name": "RequestAssetScan",
    "parse": "XMLA",
    "operation": "POST",
    "body_list": [
      "<?xml version=\"1.0\" encoding=\"UTF-8\"?>",
      "<SiteDevicesScanRequest session-id=\"${S::SESSID}\"
site-id=\"${L:A:SiteID}\">",
      "${L:A:R7ScanHostsRanges}",
      "${L:A:R7ScanSch}",
      "</SiteDevicesScanRequest>"
    ]
  },
  {
    "name": "scan_site(errorcheck)",
    "operation": "CONDITION",
    "condition": {
      "statements": [
        {"left": "SiteDevicesScanResponse", "op": "!=", "right":
"${P:A:PARSE[[name]]}"},
        {"left": "${P:A:PARSE{{success}}}", "op": "!=", "right": "1"}
      ],
      "condition_type": "OR",
      "error": true
    }
  },
```

</td>
<td>Send SiteDevicesScanRequest API request to Rapid7 Nexpose
If the request was not executed successfully, raise an error and stop execution</td>
</tr>
<tr>
<td>

```
  {
    "name": "checkSaveLastScan",
    "operation": "CONDITION",
    "condition": {
      "condition_type": "OR",
      "statements": [
        {"left": "${L::SaveEA}", "op": "!=", "right": "true"},
        {"left": "${L::EASource}", "op": "==", "right": "Net"}
      ],
      "next": "Fin"
    }
  },

  {
     "name": "Update R7_LastScan",
     "operation": "PUT",
     "transport": {"path": "${L:A:Obj_ref}"},
     "wapi": "v2.6",
     "wapi_quoting": "JSON",
      "body_list": [
         "{",
         "\"extattrs+\":{\"R7_LastScan\": { \"value\": \"${L:U:ScanDate}\"}}",
         "}"
      ]
```

</td>
<td>If <b>SaveEA</b> set to true and <b>EASource</b> is set to IP or HOST, update <b>R7_LastScan</b> extensible attribute.</td>
</tr>
</table>

},	

{
"name": "Fin",
"operation": "NOP",
"body": "${XC:DEBUG:{L:}}${XC:DEBUG:{E:}}${XC:DEBUG:{P:}}"
}
]
}
	If log level set to DEBUG, print all variables in the debug log.