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": "Debug#0",
"operation": "NOP",
"body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"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",	Stop if R7_Site is not set or R7_ScanOnEvent set to “false”

"condition": {
"condition_type": "OR",
"statements": [
{
"left": "${E::ip.extattrs{R7_Site}}",
"op": "==",
"right": ""
},
{
"left": "${E::ip.extattrs{R7_ScanOnEvent}}",
"op": "==",
"right": "false"
}
],
"stop": true
}
},	

{
"name": "checkIfHOSTname",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{
"left": "${L:A:Hostname}",
"op": "==",
"right": ""
}
],
"next": "setLIPVarsWithOutHostName"
}
},
{
"name": "setLIPVarsWithHostName",
"operation": "NOP",
"body_list": [
"${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

{	Update: This does everything

"name": "Skip next step to avoid errors",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{
"left": "1",
"op": "==",
"right": "1"
}
],
"next": "setIPSiteID"
}
},
{
"name": "setLIPVarsWithOutHostName",
"operation": "NOP",
"body_list": [
"${XC:COPY:{L:source_ip}:{E:source_ip}}",
"${XC:ASSIGN:{L:EASource}:{S:IP}}",
"${XC:ASSIGN:{L:Hostname}:{S:}}",
"${XC:ASSIGN:{L:SaveEA}:{S:false}}",
"${XC:COPY:{L:Site}:{E:ip.extattrs{R7_Site}}}"
]
},
	the above step does except that it assigns the hostname to a blank string because there are no **ip.names** in some cases. This helps avoid errors.

{
"name": "setIPSiteID",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{
"left": "${E::ip.extattrs{R7_SiteID}}",
"op": "==",
"right": ""
}
],
"eval":
"${XC:ASSIGN:{L:SiteID}:{I:0}}${XC:ASSIGN:{L:LastScan}:{S:}}",
"else_eval":
"${XC:COPY:{L:SiteID}:{E:ip.extattrs{R7_SiteID}}}"
}
},
{
	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": "Debug#4",
            "operation": "NOP",
            "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${X
C:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:D
EBUG:{UT:}}${XC:DEBUG:{R:}}"
        },
        {
            "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":
"${XC:COPY:{L:LastScan}:{E:ip.extattrs{R7_LastScan}}}"
            }
        },
        {
            "name": "Debug#5",
            "operation": "NOP",
            "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${X
C:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:D
EBUG:{UT:}}${XC:DEBUG:{R:}}"
        },
        {
            "name": "setIPScanTemplate",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${E::ip.extattrs{R7_ScanTemplate}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval":
"${XC:ASSIGN:{L:ScanTemplate}:{S:default}}",
                "else_eval":
```

"${XC:COPY:{L:ScanTemplate}:{E:ip.extattrs{R7_ScanTemplate}}}"
}
},
{
"name": "Debug#6",
"operation": "NOP",
"body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"name": "setIPAddByHostname",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{
"left":
"${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}}}"
}
},	

{
"name": "checkNetView",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{
"left": "${E::network.network_view}",
"op": "==",
"right": ""
}
],
"next": "assignScanVars",
	check if Network View is not exists go to assignScanVars. if it is exists set **network_view** local variable

"else_eval": "${XC:COPY:{L:network_view}:{E:network.network_view}}" } },	

{
"name": "Get IPv4Fixed _ref",
"operation": "GET",
"transport": {
"path": "fixedaddress?ipv4addr=${L:U:source_ip}&network_view=${L:U:network_view}"
},
"wapi": "v2.6"
},
{
"name": "Debug#9",
"operation": "NOP",
"body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"operation": "CONDITION",
"name": "wapi_response_getIPv4Fix_ref",
"condition": {
"statements": [
{
"left": "${P:A:PARSE[0]{_ref}}",
"op": "!=",
"right": ""
}
],
"condition_type": "AND",
"next": "Get_Objref"
}
},
{
"name": "Debug#10",
"operation": "NOP",
"body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
	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 HostIPv4 _ref",
        "operation": "GET",
        "transport": {
            "path":
"record:host?ipv4addr=${L:U:source_ip}&network_view=${L:
U:network_view}"
        },
        "wapi": "v2.6"
    },
    {
        "name": "Debug#11",
        "operation": "NOP",
        "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${X
C:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:D
EBUG:{UT:}}${XC:DEBUG:{R:}}"
    },
    {
        "operation": "CONDITION",
        "name": "wapi_response_getIPv4Host_ref",
        "condition": {
            "statements": [
                {
                    "left": "${P:A:PARSE[0]{_ref}}",
                    "op": "!=",
                    "right": ""
                }
            ],
            "condition_type": "AND",
            "next": "Get_Objref"
        }
    },
    {
        "name": "Debug#12",
        "operation": "NOP",
        "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${X
C:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:D
EBUG:{UT:}}${XC:DEBUG:{R:}}"
    },
    {
        "name": "Get IPv6Fixed _ref",
        "operation": "GET",
        "transport": {
            "path":
"ipv6fixedaddress?ipv4addr=${L:U:source_ip}&network_vie
```

```
w=${L:U:network_view}"
          },
          "wapi": "v2.6"
       },
       {
          "name": "Debug#13",
          "operation": "NOP",
          "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${X
C:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:D
EBUG:{UT:}}${XC:DEBUG:{R:}}"
       },
       {
          "operation": "CONDITION",
          "name": "wapi_response_getIPv6Fix_ref",
          "condition": {
             "statements": [
                {
                   "left": "${P:A:PARSE[0]{_ref}}",
                   "op": "!=",
                   "right": ""
                }
             ],
             "condition_type": "AND",
             "next": "Get_Objref"
          }
       },
       {
          "name": "Debug#14",
          "operation": "NOP",
          "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${X
C:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:D
EBUG:{UT:}}${XC:DEBUG:{R:}}"
       },
       {
          "name": "Get HostIPv6 _ref",
          "operation": "GET",
          "transport": {
             "path":
"record:host?ipv6addr=${L:U:source_ip}&network_view=${L:
U:network_view}"
          },
          "wapi": "v2.6"
       },
       {
          "name": "Debug#15",
```

"operation": "NOP",
"body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"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 **ifR7_ScanOnEvent** does notexists 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": "Debug#21",
"operation": "NOP",
"body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"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}}}"
}
},
{
"name": "Debug#22",
"operation": "NOP",
	Set the local variables (for the variable description see setLIPVars step)

"body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"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",
	Stop If the asset was scanned today

"statements": [
{
"left": "${L::LastScan}",
"op": "==",
"right": "${L::ScanDate}"
}
],
"stop": true
}
},	

{
"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": ["<?xml version=\"1.0\" encoding=\"UTF-8\"?>", "<SiteListingRequest session-id=\"${S::SESSID}\" />"] }, { "name": "Debug#27", "operation": "NOP", "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}" }, {	The code (from this step to “Create a schedule”) is executed if R7_SiteID attribute was not set and it tries to 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

"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": "${XC:COPY:{L:site_list}:{P:PARSE}",
"error": true
}
},
{
"name": "Debug#28",
"operation": "NOP",
"body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"name": "Check if sites list is empty",
"operation": "CONDITION",
"condition": {
"statements": [
{
"left": "${L:L:site_list}",
"op": "==",
"right": "0"
}
],
"condition_type": "AND",
"stop": true
}
},
{
"name": "Debug#29",
"operation": "NOP",
"body":
	attribute and jumps to “Create a schedule”.
	
	Stop if the Site was not found.

```
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
      },
      {
         "name": "Pop site from the list",
         "operation": "VARIABLEOP",
         "variable_ops": [
            {
               "operation": "POP",
               "type": "COMPOSITE",
               "destination": "L:a_site",
               "source": "L:site_list"
            }
         ]
      },
      {
         "name": "Debug#30",
         "operation": "NOP",
         "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
      },
      {
         "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":
"${XC:COPY:{L:SiteID}:{L:a_site{{id}}}}"
         }
      },
      {
         "name": "Debug#31",
         "operation": "NOP",
         "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:D
```

EBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"name": "checkSaveSiteID",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [
{
"left": "${L::SaveEA}",
"op": "!=",
"right": "true"
}
],
"next": "Create a schedule"
}
},
{
"name": "Debug#32",
"operation": "NOP",
"body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"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": [{	XML templates are created for an API request: **R7ScanSch** - contains a schedule with a scan template

"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>"
}
]
},
	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",
"statements": [
{
"left": "${L::ScanTemplate}",
"op": "==",
"right": "default"
},
{
"left": "${L::ScanTemplate}",
"op": "==",
"right": ""
}
],
"eval": "${XC:ASSIGN:{L:R7ScanSch}:{S:}}"
}
},
	“default” is a fake scan template name. If a “default” scan was requested we do not add a schedule section into the API request. Default parameters defined for a Site in Rapid7 Nexpose will be used

{
"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": "Debug#37",
"operation": "NOP",
"body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"name": "scan_site(errorcheck)",
"operation": "CONDITION",
"condition": {
	Send SiteDevicesScanRequest API request to Rapid7 Nexpose
	If the request was not executed successfully, raise an error and stop execution

"statements": [
{
"left": "SiteDevicesScanResponse",
"op": "!=",
"right": "${P:A:PARSE[[name]]}"
},
{
"left": "${P:A:PARSE{{success}}}",
"op": "!=",
"right": "1"
}
],
"condition_type": "OR",
"error": true
}
},	

{
"name": "checkSaveLastScan",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
{
"left": "${L::SaveEA}",
"op": "!=",
"right": "true"
},
{
"left": "${L::EASource}",
"op": "==",
"right": "Net"
}
],
"next": "Fin"
}
},
{
"name": "Debug#39",
"operation": "NOP",
"body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}${XC:DEBUG:{R:}}"
},
{
"name": "Update R7_LastScan",
	If **SaveEA** set to true and **EASource** is set to IP or HOST, update **R7_LastScan** extensible attribute.

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

{
"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.