FS_SecEvents template

Template	Comments
{ "version": "2.0", "name": "ForeScout SecEvent Mgmt", "comment": "SecEvent Management", "type": "REST_EVENT", "event_type": ["RPZ","TUNNEL"], "transport": {"path": "/fsapi/niCore/Hosts"}, "action_type": "Assets Management", "content_type": "application/xml", "vendor_identifier": "ForeScout", "quoting": "XML",	“version” must be set to “2.0” (NIOS 8.1 supports version "2.0") This template can be used with RPZ and TUNNEL events/notifications. The API calls will use "/fsapi/niCore/Hosts" as a default path XML quoting is used by default.
"steps": [{ "name": "DebugOnStart", "operation": "NOP", "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:{UT:}}" },	Steps block Debug output all variables in H, E, I, L, S, O, UT name spaces
{ "name": "assignRemediateTime", "operation": "NOP", "body_list": ["${XC:COPY:{L:ScanDate}:{UT:TIME}}${XC:FORMAT:TRUNCATE:{L:ScanDate}:{10t}}"] },	Assign a local variable ScanDate which will be used to populate FS_SyncedAt extensible attribute

{
"name": "check_EA_on_IP",
"operation": "CONDITION",
"condition": {
"condition_type": "AND",
"statements": [{"left": "${E:A:ip.extattrs{FS_RemediateOnEvent}}", "op": "==", "right": "true"},
{"left": "${E:A:ip.extattrs{FS_RemediatedAt}}", "op": "!=", "right": "${L:A:ScanDate}"}],
"next": "Remediate_IT"}
},
	Check if FS_RemediateOnEvent extensible attribute is set to true on the object level it was not checked/scanned today. If yes jump to "Remediate_IT"

{
"name": "check_EA_on_Net",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [
	Check if FS_RemediateOnEvent extensible attribute is not set to true on the network level or the object was checked/scanned today. If yes jump stop the template execution

{"left": "${E::network.extattrs{FS_RemediateOnEvent}}", "op": "!=", "right": "true"}, {"left": "${E:A:ip.extattrs{FS_RemediatedAt}}", "op": "==", "right": "${L:A:ScanDate}"}], "stop": true } },	

{
"name": "Remediate_IT",
"operation": "POST",
"body_list": [
"<?xml version=\"1.0\" encoding=\"UTF-8\"?>",
"<FSAPI TYPE=\"request\" API_VERSION=\"1.0\">",
"<TRANSACTION TYPE=\"update\">",
"<OPTIONS CREATE_NEW_HOST=\"true\"/>",
"<HOST_KEY NAME=\"ip\" VALUE=\"${E::source_ip}\"/>",
"<PROPERTIES>",
"<PROPERTY NAME=\"IB_Scan\"><VALUE>Remediate</VALUE></PROPERTY>",
"</PROPERTIES>",
"</TRANSACTION>",
"</FSAPI>"
],
"parse": "XMLA"
},{
"name": "check action",
"operation": "CONDITION",
"condition": {
"statements": [{ "left": "${P:A:PARSE{FSAPI}{STATUS}{CODE}}", "op": "!=", "right": "FSAPI_OK"}],"condition_type": "OR",
"error": true}
},
	Update IB_Scan property on ForeScout side and check the response code. IB_Scan property should trigger a policy on ForeScout

{
"name": "checkNetView",
"operation": "CONDITION",
"condition": {
"condition_type": "OR",
"statements": [{"left": "${E::network.network_view}", "op": "==", "right": ""}],
"eval": "${XC:ASSIGN:{L:network_view}:{S:default}}",
"else_eval": "${XC:COPY:{L:network_view}:{E:network.network_view}}"}
},{
"name": "Get IPv4Fixed _ref",
"operation": "GET",
"transport": {"path": "fixedaddress?ipv4addr=${E:U:source_ip}&network_view=${L:U:network_view}"},
"wapi": "v2.6"
},{
	These steps looking for _ref attribute for the IP. If there is no such object stop the semplate

"operation": "CONDITION",
"name": "wapi_response_getIPv4Fix_ref",
"condition": {
"statements": [{"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}],
"condition_type": "AND",
"next": "Get_Objref"
}
},{
"name": "Get HostIPv4 _ref",
"operation": "GET",
"transport": {"path":
"record:host?ipv4addr=${E: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"}
},{
"name": "Check_if_Save",
"operation": "CONDITION",
"condition": {
"statements": [{"left": "1", "op": "==", "right": "1"}],
"condition_type": "AND",
"stop": true}
},{
"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}}}"}
},	

{
"name": "Update Remediate Time",
"operation": "PUT",
"transport": {"path": "${L:A:Obj_ref}"},
"wapi": "v2.6",
"wapi_quoting": "JSON",
"body_list": [
"{", "\"extattrs+\":{\"FS_RemediatedAt\": { \"value\":
\"${L:A:ScanDate}\"}}","}"]}]
}
	Update FS_RemediatedAt
	extensible attribute