TenableScan.json template

Template	Comments
{ "name": "Tenable Scan", "vendor_identifier": "Tenable", "comment": "Tenable scan assets by a security event", "version": "3.0", "type": "REST_EVENT", "event_type": ["RPZ","TUNNEL"], "content_type": "application/json", "headers": {"X-Requested-With": "XMLHttpRequest", "X-SecurityCenter": "${S:A:SESSID}"},	“version” must be set to “3.0” (NIOS 8.2 supports version "3.0") This template can be used with RPZ and TUNNEL events/notifications. X-SecurityCenter parameter send in HTTP headers to authenticate the session
"steps": [{ "name": "DebugOnStart", "operation": "NOP", "body": "${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": "checkIPEAs",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${E:A:ip.extattrs{TNBL_ScanOnEvnt}}", "op": "==", "right": ""}],
"next": "checkNetEAs"}
},{
"name": "checkIPScanOnEvent",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left": "${E:A:ip.extattrs{TNBL_ScanOnEvnt}}", "op": "==", "right": "false"}],
"stop": true}
},
	If TNBL_ScanOnEvnt is not set on an object (IP) level or there is no such object go to "checkNetEAs" step.
	
	If a scan is not requested for the object, stop the template

{
"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}}"]
},{
"name": "blocked_domain",
"operation": "CONDITION",
"condition": {
	Set the local variables to use the common name for both events type

"condition_type": "AND", "statements": [{"left": "${E:A:event_type}", "op": "==", "right": "RPZ"}],
"eval": "${XC:COPY:{L:BlockedDomain}:{E:query_name}}",
"else_eval": "${XC:COPY:{L:BlockedDomain}:{E:domain_name}}"}
},{
"name": "setIPLastScan",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left": "${E:A:ip.extattrs{TNBL_ScanTime}}", "op": "==", "right": ""}],
"eval": "${XC:ASSIGN:{L:LastScan}:{S:}}",
"else_eval": "${XC:COPY:{L:LastScan}:{E:ip.extattrs{TNBL_ScanTime}}}"}
},	

{
"name": "setIPScanTemplate",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left": "${E:A:ip.extattrs{TNBL_ScanTemplate}}", "op": "==", "right": ""}],
"stop": true,
"else_eval": "${XC:COPY:{L:ScanTemplate}:{E:ip.extattrs{TNBL_ScanTemplate}}}"}
},
	If TNBL_ScanTemplate is not set, stop the template

{
"name": "setIPScanTemplateID",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left": "${E:A:ip.extattrs{TNBL_ScanTemplate}}", "op": "==", "right": ""}],
"eval": "${XC:ASSIGN:{L:ScanTemplateID}:{S:false}}",
"else_eval": "${XC:COPY:{L:ScanTemplateID}:{E:ip.extattrs{TNBL_ScanTemplateID}}}"}
},{
"name": "setIPAddByHostname",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left": "${E:A:ip.extattrs{TNBL_AddByHostname}}", "op": "==", "right": ""}],
"eval": "${XC:ASSIGN:{L:AddByHostname}:{S:false}}",
"else_eval": "${XC:COPY:{L:AddByHostname}:{E:ip.extattrs{TNBL_AddByHostname}}}"}
},
	If TNBL_ScanTemplateID, TNBL_AddByHostname are set the template will use it for scanning

{
"name": "checkNetView",
"operation": "CONDITION",
"condition": {
	If network view was not set go to "assignScanVars" step

"condition_type": "OR","statements": [{"left": "${E:A:network.network_view}", "op": "==", "right": ""}], "next": "assignScanVars", "else_eval": "${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": "v2.7" },{ "operation": "CONDITION", "name": "wapi_response_getIPv4Fix_ref", "condition": { "condition_type": "AND","statements": [{"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}], "next": "Get_Objref"} },{ "name": "Get HostIPv4 _ref", "operation": "GET", "transport": {"path": "record:host?ipv4addr=${L:U:Source_ip}&network_view=${L:U:network_view}"}, "wapi": "v2.7" },{ "operation": "CONDITION", "name": "wapi_response_getIPv4Host_ref", "condition": { "condition_type": "AND","statements": [{"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}], "next": "Get_Objref"} },{ "name": "Get IPv6Fixed _ref", "operation": "GET", "transport": {"path": "ipv6fixedaddress?ipv4addr=${L:U:Source_ip}&network_view=${L:U:network_view}"}, "wapi": "v2.7" },{ "operation": "CONDITION", "name": "wapi_response_getIPv6Fix_ref", "condition": { "condition_type": "AND","statements": [{"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}], "next": "Get_Objref"} },{ "name": "Get HostIPv6 _ref", "operation": "GET", "transport": {"path": "record:host?ipv6addr=${L:U:Source_ip}&network_view=${L:U:network_view}"},	To update TNBL_ScanTime the template tries to determine _ref property of the object.

"wapi": "v2.7"
},{
"operation": "CONDITION",
"name": "wapi_response_getIPv6Host_ref",
"condition": {
"condition_type": "AND","statements": [{"left": "${P:A:PARSE[0]{_ref}}",
"op": "!=", "right": ""}],
"next": "Get_Objref"}
},{
"name": "Get_Objref",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${P:A:PARSE[0]{_ref}}",
"op": "!=", "right": ""}],
"eval":
"${XC:COPY:{L:Obj_ref}:{P:PARSE[0]{_ref}}}${XC:ASSIGN:{L:SaveEA}:
{S:true}}"}
},{
"name": "CheckIfHost",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${L:A:Obj_ref}", "op":
"=~", "right": "record:host"}],
"eval": "${XC:ASSIGN:{L:EASource}:{S:HOST}}"}
},	

{
"name": "goToAssignScanVars",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left": "", "op": "==", "right": ""}],
"next": "assignScanVars"}
},
	Go to "assignScanVars" step

{
"name": "checkNetEAs",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [
{"left": "${E:A:network.extattrs{TNBL_ScanOnEvnt}}", "op": "==", "right":
""},
{"left": "${E:A:network.extattrs{TNBL_ScanOnEvnt}}", "op": "==", "right":
"false"}],
"stop": true}
},
	if TNBL_ScanOnEvnt is not set to true on the network level, stop the template

{ "name": "setLNetVars", "operation": "NOP", "body_list": ["${XC:COPY:{L:Source_ip}:{E:source_ip}}", "${XC:ASSIGN:{L:LastScan}:{S:}}",	Set local variables

"${XC:ASSIGN:{L:EASource}:{S:Net}}", "${XC:ASSIGN:{L:SaveEA}:{S:false}}", "${XC:ASSIGN:{L:Hostname}:{S:}}", "${XC:ASSIGN:{L:AddByHostname}:{S:false}}"] },	

{
"name": "setNetScanTemplate",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left":
"${E:A:network.extattrs{TNBL_ScanTemplate}}", "op": "==", "right": ""}],
"stop": true,
"else_eval":
"${XC:COPY:{L:ScanTemplate}:{E:network.extattrs{TNBL_ScanTemplate}}}"}
},{
"name": "setNetScanTemplateID",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left":
"${E:A:network.extattrs{TNBL_ScanTemplateID}}", "op": "==", "right":
""}],
"eval": "${XC:ASSIGN:{L:ScanTemplateID}:{S:false}}",
"else_eval":
"${XC:COPY:{L:ScanTemplateID}:{E:network.extattrs{TNBL_ScanTemplateID}}}"}
},
	If Active Scan (template) is not set, stop the template
	
	Assign Active Scan ID to the local variable if exists

{
"name": "Get Network _ref",
"operation": "GET",
"transport": {"path":
"network?network=${E:U:network.network}&network_view=${E:U:network.network_view}"},
"wapi": "v2.7"
},{
"operation": "CONDITION",
"name": "wapi_response_network_ref",
"condition": {
"condition_type": "AND","statements": [{"left": "${P:A:PARSE[0]{_ref}}",
"op": "!=", "right": ""}],
"next": "Get_NetObjref"}
},{
"name": "Get IPv6Network _ref",
"operation": "GET",
"transport": {"path":
"ipv6network?network=${E:U:network.network}&network_view=${E:U:network.network_view}"},
"wapi": "v2.7"
},{
"name": "Get_NetObjref",
	Get _ref parameter for a network

"operation": "CONDITION", "condition": { "condition_type": "AND","statements": [{"left": "${P:A:PARSE[0]{_ref}}", "op": "!=", "right": ""}], "eval": "${XC:COPY:{L:Obj_ref}:{P:PARSE[0]{_ref}}}${XC:ASSIGN:{L:SaveEA}:{S:true}}"} },	
{ "name": "assignScanVars", "operation": "NOP", "body_list": ["${XC:COPY:{L:ScanDate}:{UT:TIME}}${XC:FORMAT:TRUNCATE:{L:ScanDate}:{10t}}", "${XC:COPY:{L:ScanSchTime}:{UT:EPOCH}}${XC:FORMAT:DATE_STRFTIME:{L:ScanSchTime}:{%Y%m%dT%H%M59000Z}}"] },	Assign scan time

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

{
"name": "scanByHostname",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [
{"left": "${L:A:AddByHostname}", "op": "==", "right": "true"},
{"left": "${L:A:Hostname}", "op": "!=", "right": ""},
{"left": "${L:A:EASource}", "op": "==", "right": "HOST"}
],
"eval": "${XC:COPY:{L:ScanObject}:{L:Hostname}}",
"else_eval": "${XC:COPY:{L:ScanObject}:{L:Source_ip}}"}
},
	For hosts with a hostname and if they were added by a hostname, hostname is used as a scan target. Other objects are scanned by IP.

{
"name": "Get a UserID",
"operation": "GET",
"parse": "JSON",
"transport": {"path": "/currentUser"}
},{
"name": "Check a user",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${P:A:error_code}", "op":
"!=", "right": "0"}],
"error": true,
	Request Tenable User ID

"else_eval": "${XC:COPY:{L:TNBL_UserId}:{P:response{id}}}"} },	

{
"name": "checkIfExistsScanTemplateID",
"operation": "CONDITION",
"condition": {
"condition_type": "OR","statements": [{"left": "${L:A:ScanTemplateID}",
"op": "!=", "right": "false"}],
"next": "Copy a scan template"}
},
	If TNBL_ScanTemplateID is defined, go to "Copy a scan template" step

{
"name": "Request all scans",
"parse": "JSON",
"operation": "GET",
"transport": {"path": "/scan"}
},{
"name": "Check all scans request on errors",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${P:A:error_code}", "op":
"!=", "right": "0"}],
"else_eval": "${XC:COPY:{L:object_list}:{P:response{manageable}}",
"error": true}
},
	Request list of active scans and copy manageable scans to a local variable

{
"name": "Check if list is empty",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${L:L:object_list}", "op":
"==", "right": "0"}],
"stop": true}
},{
"name": "Pop object from the list",
"operation": "VARIABLEOP",
"variable_ops": [
{
"operation": "POP",
"type": "DICTIONARY",
"destination": "L:an_object",
"source": "L:object_list"}]
},{
"name": "check an object",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${L:A:ScanTemplate}",
"op": "!=", "right": "${L:A:an_object{name}}"}],
"next": "Check if list is empty",
"else_eval": "${XC:COPY:{L:ScanTemplateID}:{L:an_object{id}}}"}
},
	A loop which identifies active scan ID by a name.

{
"name": "checkSaveScanID",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${L:A:SaveEA}", "op":
"!=", "right": "true"}],
"next": "Copy a scan template"}
},{
"name": "Update ScanID",
"operation": "PUT",
"transport": {"path": "${L:A:Obj_ref}"},
"wapi": "v2.7",
"wapi_quoting": "JSON",
"body_list": [
"{\"extattrs+\":{\"TNBL_ScanTemplateID\": { \"value\":
\"${L:A:ScanTemplateID}\"}}}"]
},
	Save active scan ID to TNBL_ScanTemplateID extensible attribute

{
"name": "Copy a scan template",
"operation": "POST",
"parse": "JSON",
"transport": {"path": "/scan/${L:A:ScanTemplateID}/copy"},
"body_list": [
"{\"targetUser\":{\"id\":\"${L:A:TNBL_UserId}\"},\"name\":\"${L:A:ScanObj
ect} scan requested by IB Outbound API ${E:A:event_type} event at
${L:A:ScanSchTime}. Blocked domain: ${L:A:BlockedDomain}\"}"]
},{
"name": "Check Copy",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${P:A:error_code}", "op":
"!=", "right": "0"}],
"error": true}
},
	Copy an active scan (which is used as a template) and populate name field with information why the scan was requested

{
"name": "Run a scan",
"operation": "PATCH",
"parse": "JSON",
"transport": {"path": "/scan/${P:A:response{scan}{id}}"},
"body_list": [
"{\"ipList\":\"${L:A:ScanObject}\",\"schedule\":{\"repeatRule\":
\"FREQ=NOW;INTERVAL=1\", \"type\": \"now\"}}"]
},{
"name": "Check Run a scan",
"operation": "CONDITION",
"condition": {
"condition_type": "AND","statements": [{"left": "${P:A:error_code}", "op":
"!=", "right": "0"}],
"error": true}
},
	Execute a scan (copy) on Tenable

{
"name": "checkSaveLastScan",
"operation": "CONDITION",
"condition": {
"condition_type": "OR", "statements": [
{"left": "${L:A:SaveEA}", "op": "!=", "right": "true"},
{"left": "${L:A:EASource}", "op": "==", "right": "Net"}
],
"next": "Fin"}
},{
"name": "Update_LastScan",
"operation": "PUT",
"transport": {"path": "${L:A:Obj_ref}"},
"wapi": "v2.7",
"wapi_quoting": "JSON",
"body_list": [
"{\"extattrs+\":{\"TNBL_ScanTime\": { \"value\": \"${L:U:ScanDate}\"}}}"]
},{
"name": "Fin",
"operation": "NOP",
"body": "${XC:DEBUG:{L:}}${XC:DEBUG:{E:}}${XC:DEBUG:{P:}}"
}]
}
	Update TNBL_ScanTime extensible attributes for an asset. TNBL_ScanTime is not updated on network and range level