```
{
    "version": "4.0",
    "name": "Aruba ClearPass Security",
    "comment": "",
    "type": "REST_EVENT",
    "event_type": [
        "RPZ",
        "TUNNEL",
        "ADP"
    ],
    "content_type": "application/json",
    "vendor_identifier": "Aruba ClearPass",
    "headers": {
        "Accept": "*/*"
    },
    "instance_variables": [
        {
            "name": "ThreatSeverity",
            "type": "STRING",
            "value": "Low"
        }
    ],
    "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:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "check if IPv4 or IPv6 for assigning variables",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:source_ip}",
                        "op": "=~",
                        "right": ":"
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:address}:{E:source_ip}}$
{XC:ASSIGN:{L:addr}:{S:ipv4addr}}${XC:ASSIGN:{L:fixed}:
{S:fixedaddress}}",
                "else_eval": "${XC:COPY:{L:address}:{E:source_ip}}$
{XC:ASSIGN:{L:addr}:{S:ipv6addr}}${XC:ASSIGN:{L:fixed}:
{S:ipv6fixedaddress}}"
            }
        },
        {
            "name": "assignTimeValue",
            "operation": "NOP",
            "body_list": [
```

```
                "${XC:COPY:{L:ArubaAddDate}:{UT:TIME}}$
{XC:FORMAT:TRUNCATE:{L:ArubaAddDate}:{10t}}"
            ]
        },
        {
            "name": "check for IPv6",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E::source_ip}",
                        "op": "=~",
                        "right": ":"
                    }
                ],
                "condition_type": "AND",
                "next": "Get IPv6Fixed _ref"
            }
        },
        {
            "name": "Get IPv4Fixed _ref",
            "operation": "GET",
            "transport": {
                "path": "fixedaddress?ipv4addr=${E:U:source_ip}
&network_view=default&_return_fields=extattrs"
            },
            "wapi": "v2.7"
        },
        {
            "operation": "CONDITION",
            "name": "wapi_response_getIPv4Fix_ref",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{_ref}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "next": "Get_Objref"
            }
        },
        {
            "name": "Get HostIPv4 _ref",
            "operation": "GET",
            "transport": {
                "path": "record:host?ipv4addr=${E:U:source_ip}
&network_view=default&_return_fields=extattrs"
            },
            "wapi": "v2.7"
        },
        {
            "operation": "CONDITION",
```

```
            "name": "wapi_response_getIPv4Host_ref",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{_ref}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "next": "Get_Objref",
                "else_stop": true
            }
        },
        {
            "name": "Get IPv6Fixed _ref",
            "operation": "GET",
            "transport": {
                "path": "ipv6fixedaddress?ipv6addr=${E:U:source_ip}
&network_view=default&_return_fields=extattrs"
            },
            "wapi": "v2.7"
        },
        {
            "operation": "CONDITION",
            "name": "wapi_response_getIPv6Fix_ref",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{_ref}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "next": "Get_Objref"
            }
        },
        {
            "name": "Get HostIPv6 _ref",
            "operation": "GET",
            "transport": {
                "path": "record:host?ipv6addr=${E:U:source_ip}
&network_view=default&_return_fields=extattrs"
            },
            "wapi": "v2.7"
        },
        {
            "operation": "CONDITION",
            "name": "wapi_response_getIPv6Host_ref",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
```

```
                        "left": "${P:A:PARSE[0]{_ref}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "next": "Get_Objref",
                "else_stop": true
            }
        },
        {
            "name": "Get_Objref",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{_ref}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:Obj_ref}:{P:PARSE[0]{_ref}}}"
            }
        },
        {
            "name": "Stop if no Obj_ref",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${L:A:Obj_ref}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "stop if no extattrs",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{extattrs}
{Aruba_Secure}{value}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "stop": true
            }
```

```
        },
        {
            "name": "Set Old_Time",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{extattrs}
{Aruba_LastSecurityEvent}{value}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:ArubaAddDateRecorded}:
{S:}}",
                "else_eval": "${XC:COPY:{L:ArubaAddDateRecorded}:
{P:PARSE[0]{extattrs}{Aruba_LastSecurityEvent}{value}}}$
{XC:FORMAT:TRUNCATE:{L:ArubaAddDateRecorded}:{10t}}"
            }
        },
        {
            "name": "check if secure external attribute set",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{extattrs}
{Aruba_Secure}{value}}",
                        "op": "==",
                        "right": "false"
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "Get Lease",
            "operation": "GET",
            "transport": {
                "path": "lease?address=${L:A:address}&network_view=$
{E:A:network.network_view}
&_return_fields=hardware,discovered_data.mac_address,discovered_data
.vmhost_mac_address,discovered_data.vport_mac_address"
            },
            "wapi": "v2.7"
        },
            {
            "name": "Debug ADP",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
```

```
        },
        {
            "name": "check if ADP event",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "==",
                        "right": "ADP"
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:ThreatName}:{E:query_fqdn}}$
{XC:COPY:{L:ThreatCategory}:{E:event_type}}${XC:COPY:
{L:Infoblox_Last_Known_IP}:{E:source_ip}}${XC:COPY:
{L:ThreatDetection}:{E:member_ip}}${XC:COPY:{L:RuleId}:{E:rule_sid}}
${XC:COPY:{L:RuleCategory}:{E:rule_category}}${XC:COPY:
{L:ThreatSeverity}:{I:ThreatSeverity}}",
                "next": "Check if lease is present"
            }
        },

        {
            "name": "Check RPZ or Tunnel event to assign variables",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "==",
                        "right": "RPZ"
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:ThreatCategory}:{E:rpz_type}}$
{XC:COPY:{L:Infoblox_Last_Known_IP}:{E:source_ip}}${XC:COPY:
{L:ThreatDetection}:{E:member_ip}}${XC:COPY:{L:ThreatName}:
{E:query_name}}${XC:COPY:{L:ThreatSeverity}:{I:ThreatSeverity}}",
                "else_eval": "${XC:COPY:{L:ThreatCategory}:
{E:event_type}}${XC:COPY:{L:Infoblox_Last_Known_IP}:{E:source_ip}}$
{XC:COPY:{L:ThreatDetection}:{E:member_ip}}${XC:COPY:{L:ThreatName}:
{E:domain_name}}${XC:COPY:{L:ThreatSeverity}:{I:ThreatSeverity}}"
            }
        },
        {
            "name": "Check if lease is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]}",
                        "op": "!=",
                        "right": ""
```

```
                    },
                    {
                        "left": "${E:A:ip.extattrs{Aruba_Secure}
{value}}",
                        "op": "==",
                        "right": "true"
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:client_hostname}:
{E:client_hostname}}",
                "next": "check if mac is present"
            }
        },
        {
            "name": "check if Lease and if so then stop",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "Check if location",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${E:A:values{extattrs}
{Aruba_Location}{value}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:Location}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:Location}:
{E:values{extattrs}{Aruba_Location}{value}}}"
            }
        },
        {
            "name": "check if IPv4 or IPv6 for checking assets on
Infoblox",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
```

```
                    {
                        "left": "${E:A:source_ip}",
                        "op": "=~",
                        "right": ":"
                    }
                ],
                "next": "Get Host IPv6"
            }
        },
        {
            "name": "Get Host IPv4",
            "operation": "GET",
            "transport": {
                "path": "record:host_ipv4addr?ipv4addr=$
{L:A:address}&network_view=${E:A:network.network_view}
&_return_fields=mac,host"
            },
            "wapi": "v2.7"
        },
        {
            "name": "Check if Host IPv4 is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:host}:{P:PARSE[0]{host}}}",
                "next": "check if mac is present",
                "else_eval": "${XC:ASSIGN:{L:host}:{S:Unknown}}"

            }
        },
        {
            "name": "Get fixed IPv4",
            "operation": "GET",
            "transport": {
                "path": "fixedaddress?ipv4addr=${L:A:address}
&network_view=${E:A:network.network_view}
&_return_fields=mac,discovered_data.mac_address,discovered_data.vmho
st_mac_address,discovered_data.vport_mac_address,extattrs"
            },
            "wapi": "v2.7"
        },
        {
            "name": "Check if fixed IPv4 is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
```

```
                        "left": "${P:A:PARSE[0]}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:ASSIGN:{L:host}:{S:Unknown}}",
                "next": "check if mac is present"
            }
        },
        {
            "name": "stop because there is no information IPv4",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "Get Host IPv6",
            "operation": "GET",
            "transport": {
                "path": "record:host?ipv6addr=${L:A:address}
&network_view=${E:A:network.network_view}&_return_fields=mac,host"
            },
            "wapi": "v2.7"
        },
        {
            "name": "Check if Host IPv6 is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:host}:{P:PARSE[0]{host}}}",
                "next": "check if mac is present"
            }
        },
        {
            "name": "Get fixed IPv6",
            "operation": "GET",
            "transport": {
```

```
                "path": "ipv6fixedaddress?ipv6addr=${L:A:address}
&network_view=${E:A:network.network_view}
&_return_fields=discovered_data.mac_address,discovered_data.vmhost_m
ac_address,discovered_data.vport_mac_address,extattrs"
            },
            "wapi": "v2.7"
        },
        {
            "name": "Check if fixed IPv6 is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:ASSIGN:{L:host}:{S:Unknown}}",
                "next": "check if mac is present"
            }
        },
        {
            "name": "stop because there is no information IPv6",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "check if mac is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{mac}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:{P:PARSE[0]{mac}}}",
                "next": "assignMac from P: for host"
            }
        },
```

```
        {
            "name": "check if discovered mac_address is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "$
{P:A:discovered_data.mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.mac_address}}",
                "next": "assignMac from P: for host"
            }
        },
        {
            "name": "check if discovered vmhost_mac_address is
present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "$
{P:A:discovered_data.vmhost_mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.vmhost_mac_address}}",
                "next": "assignMac from P: for host"
            }
        },
        {
            "name": "check if discovered vport_mac_address is
present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "$
{P:A:discovered_data.vport_mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.vport_mac_address}}",
                "next": "assignMac from P: for host"
```

```
            }
        },
        {
            "name": "Stop bacause there is no mac",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "stop": true
            }
        },
        {
            "name": "assignMac from P: for host",
            "operation": "NOP",
            "body_list": [
                "${XC:COPY:{L:Mac1}:{L:mac}}${XC:FORMAT:TRUNCATE:
{L:Mac1}:{2t}}",
                "${XC:COPY:{L:Mac2}:{L:mac}}${XC:FORMAT:TRUNCATE:
{L:Mac2}:{5t}}${XC:FORMAT:TRUNCATE:{L:Mac2}:{-2f}}",
                "${XC:COPY:{L:Mac3}:{L:mac}}${XC:FORMAT:TRUNCATE:
{L:Mac3}:{8t}}${XC:FORMAT:TRUNCATE:{L:Mac3}:{-2f}}",
                "${XC:COPY:{L:Mac4}:{L:mac}}${XC:FORMAT:TRUNCATE:
{L:Mac4}:{11t}}${XC:FORMAT:TRUNCATE:{L:Mac4}:{-2f}}",
                "${XC:COPY:{L:Mac5}:{L:mac}}${XC:FORMAT:TRUNCATE:
{L:Mac5}:{14t}}${XC:FORMAT:TRUNCATE:{L:Mac5}:{-2f}}",
                "${XC:COPY:{L:Mac6}:{L:mac}}${XC:FORMAT:TRUNCATE:
{L:Mac6}:{-2f}}",
                "${XC:COPY:{L:MacFull}:{L:mac}}"
            ]
        },
        {
            "name": "Debug#test1",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Get Check if duplicate endpoint with host",
            "operation": "GET",
            "parse": "JSON",
            "headers": {
                "Authorization": "Bearer ${S:A:SESSID}"
            },
            "transport": {
                "path": "/api/endpoint/mac-address/${L:A:Mac1}$
{L:A:Mac2}${L:A:Mac3}${L:A:Mac4}${L:A:Mac5}${L:A:Mac6}"
            },
```

```
            "result": [{
                "codes": "200,201,202,203,204,404,405",
                "next": "Create Endpoint if one is not present"
            }]
        },
        {
            "name": "Debug#test2",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Create Endpoint if one is not present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac_address}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "condition_type": "OR",
                "else_eval": "${XC:ASSIGN:{L:MacFound}:{S:true}}"
            }
        },
                 {
            "name": "check for Location",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Location}}",
                        "op": "!=",
                        "right": ""
                    },
                      {
                        "left": "${L:A:Location}",
                        "op": "==",
                        "right": "Unknown"
                    }
                ],
                "eval": "${XC:COPY:{L:Location}:
{P:attributes{Location}}}"
            }
        },
        {
            "name": "check if managed by infoblox",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
```

```
                    {
                        "left": "${P:A:attributes{Infoblox
Managed}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:managed}:{S:False}}",
                "else_eval": "${XC:COPY:{L:managed}:
{P:attributes{Infoblox Managed}}}"
            }
        },
        {
            "name": "check for Fingerprint",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox DHCP
Fingerprint}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:fingerpring}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:fingerpring}:
{P:attributes{Infoblox DHCP Fingerprint}}}"
            }
        },
        {
            "name": "check for Device Vendor",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Device Vendor}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:vendor}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:vendor}:
{P:attributes{Device Vendor}}}"
            }
        },
        {
            "name": "check for client_hostname",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
```

```
                        "left": "$
{P:A:attributes{client_hostname}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:host}:
{P:attributes{client_hostname}}}"
            }
        },
        {
            "name": "check for Device Type",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Device Type}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:type}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:type}:
{P:attributes{Device Type}}}"
            }
        },
        {
            "name": "check for OS Version",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{OS Version}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:os_version}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:os_version}:
{P:attributes{OS Version}}}"
            }
        },
        {
            "name": "check for Model",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Model}}",
                        "op": "==",
                        "right": ""
```

```
                    }
                ],
                "eval": "${XC:ASSIGN:{L:model}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:model}:
{P:attributes{Model}}}"
            }
        },

    {
            "name": "check for Rule Category",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox
RuleCategory}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:rule_category}:
{S:Unknown}}",
               "else_eval": "${XC:COPY:{L:rule_category}:
{P:attributes{Infoblox RuleCategory}}}"
            }
        },
           {
            "name": "check for Rule id",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox
RuleId}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                 "eval": "${XC:ASSIGN:{L:rule_id}:{I:0}}",
                "else_eval": "${XC:COPY:{L:rule_id}:
{P:attributes{Infoblox RuleId}}}"
            }
        },
        {
            "name": "Get active sessions",
            "operation": "GET",
            "parse": "JSON",
            "headers": {
                "Authorization": "Bearer ${S:A:SESSID}"
            },
            "transport": {
                "path": "/api/session"
```

```
    }
},
{
    "name": "check if there was no sessions active",
    "operation": "CONDITION",
    "condition": {
        "condition_type": "OR",
        "statements": [
            {
                "left": "${P:A:_embedded{items}}",
                "op": "==",
                "right": "[]"
            }
        ],
        "next": "all discovery information"
    }
},
{
    "name": "Pop item from the list",
    "operation": "VARIABLEOP",
    "variable_ops": [
        {
            "operation": "UNSHIFT",
            "type": "DICTIONARY",
            "destination": "L:items",
            "source": "P:_embedded{items}"
        }
    ]
},
{
    "name": "check if mac is equal",
    "operation": "CONDITION",
    "condition": {
        "condition_type": "OR",
        "statements": [
            {
                "left": "${L:A:MacFull}",
                "op": "==",
                "right": "${L:A:items{mac_address}}"
            }
        ],
        "next": "disconnect the session"
    }
},
{
    "name": "check if there are more items",
    "operation": "CONDITION",
    "condition": {
        "condition_type": "OR",
        "statements": [
            {
                "left": "${P:A:_embedded{items}}",
                "op": "==",
                "right": "[]"
```

```
                    }
                ],
                "next": "check for Location",
                "else_next": "Pop item from the list"
            }
        },
        {
            "name": "disconnect the session",
            "operation": "POST",
            "parse": "JSON",
            "transport": {
                "path": "/api/session/${L:A:items{id}}/disconnect"
            },
            "body_list": [
                "{}"
            ]
        },


          {
            "name": "Debug ruleid1",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },

        {
            "name": "all discovery information",
            "operation": "GET",
            "transport": {
                "path": "discovery:device?address=${L:A:address}
&_return_fields=name,description,os_version,chassis_serial_number,mo
del,ms_ad_user_data,type,vendor,interfaces"
            },
            "wapi": "v2.7"
        },
        {
            "name": "Check if name is unknown",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{name}}",
                        "op": "==",
                        "right": ""
                    },
                    {
                        "left": "${P:A:PARSE[0]{name}}",
                        "op": "==",
                        "right": "unknown"
                    }
                ],
```

```
                "eval": "${XC:ASSIGN:{L:name}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:name}:{P:PARSE[0]
{name}}}"
            }
        },
        {
            "name": "check for description",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{description}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:description}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:description}:{P:PARSE[0]
{description}}}"
            }
        },
        {
            "name": "check for os_version",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{os_version}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:os_version}:{P:PARSE[0]
{os_version}}}"
            }
        },
        {
            "name": "check for model",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{model}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:model}:{P:PARSE[0]{model}}}"
            }
        },
        {
```

```
            "name": "check for active_users_count",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]
{active_users_count}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:active_users_count}:
{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:active_users_count}:
{P:PARSE[0]{ms_ad_user_data}{active_users_count}}}"
            }
        },
        {
            "name": "check for vendor",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{vendor}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:vendor}:{P:PARSE[0]{vendor}}}"
            }
        },
        {
            "name": "check for type",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{type}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:type}:{P:PARSE[0]{type}}}"
            }
        },
        {
            "name": "check for chassis_serial_number",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
```

```
                    {
                        "left": "${P:A:PARSE[0]
{chassis_serial_number}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:chassis_serial_number}:
{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:chassis_serial_number}:
{P:PARSE[0]{chassis_serial_number}}}"
            }
        },
        {
            "name": "check if mac was found on aruba",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${L:A:MacFound}",
                        "op": "==",
                        "right": "true"
                    }
                ],
                "condition_type": "AND",
                "next": "Add endpoint event check"
            }
        },
         {
            "name": "check if new ADP event",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "==",
                        "right": "ADP"
                    }
                ],
                "condition_type": "AND",
                "next": "Add a new ADP endpoint"
            }
        },
                    {
            "name": "Add a new endpoint",
            "operation": "POST",
            "parse": "JSON",
            "headers": {
                "Authorization": "Bearer ${S:A:SESSID}"
            },
            "transport": {
                "path": "/api/endpoint"
            },
            "body_list": [
```

```
                "{",
                "\"mac_address\":\"${L:A:MacFull}\",",
                "\"status\":\"Known\",",
                "\"description\":\"Added via API at ${UT:A:TIME}
\",",
                "\"attributes\":{",
                    "\"client_hostname\":\"${L:A:host}\",",
                    "\"Device Type\":\"${L:A:type}\",",
                    "\"Device Vendor\":\"${L:A:vendor}\",",
                    "\"Location\":\"${L:A:Location}\",",
                    "\"Model\":\"${L:A:model}\",",
                    "\"Infoblox DHCP Fingerprint\":\"$
{L:A:fingerpring}\",",
                    "\"Infoblox Last Known IP\":\"$
{L:A:Infoblox_Last_Known_IP}\",",
                    "\"OS Version\":\"${L:A:os_version}\",",
                    "\"Infoblox Managed\":\"${L:A:managed}\",",
                    "\"Infoblox Threat Category\":\"$
{L:A:ThreatCategory}\",",
                    "\"Infoblox Threat Detection Device IP\":\"$
{L:A:ThreatDetection}\",",
                    "\"Infoblox Threat Name\":\"${L:A:ThreatName}
\",",
                    "\"Infoblox Threat Severity\":\"$
{I:A:ThreatSeverity}\",",
                    "\"Infoblox Threat Status\":\"Unresolved\"",
                "}",
                "}"
            ]
        },
               {
            "name": "Jump to non ADP assign profiler values ",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "next": "assign profiler values"
            }
        },
        {
            "name": "Add a new ADP endpoint",
            "operation": "POST",
            "parse": "JSON",
            "headers": {
                "Authorization": "Bearer ${S:A:SESSID}"
            },
            "transport": {
                "path": "/api/endpoint"
```

```
            },
            "body_list": [
                "{",
                "\"mac_address\":\"${L:A:MacFull}\",",
                "\"status\":\"Known\",",
                "\"description\":\"Added via API at ${UT:A:TIME}
\",",
                "\"attributes\":{",
                    "\"client_hostname\":\"${L:A:host}\",",
                    "\"Device Type\":\"${L:A:type}\",",
                    "\"Device Vendor\":\"${L:A:vendor}\",",
                    "\"Location\":\"${L:A:Location}\",",
                    "\"Model\":\"${L:A:model}\",",
                    "\"Infoblox DHCP Fingerprint\":\"$
{L:A:fingerpring}\",",
                    "\"Infoblox Last Known IP\":\"$
{L:A:Infoblox_Last_Known_IP}\",",
                    "\"OS Version\":\"${L:A:os_version}\",",
                    "\"Infoblox Managed\":\"${L:A:managed}\",",
                    "\"Infoblox Threat Category\":\"$
{L:A:ThreatCategory}\",",
                    "\"Infoblox Threat Detection Device IP\":\"$
{L:A:ThreatDetection}\",",
                    "\"Infoblox Threat Name\":\"${L:A:ThreatName}
\",",
                    "\"Infoblox Threat Severity\":\"$
{I:A:ThreatSeverity}\",",
                    "\"Infoblox RuleId\":\"${L:A:RuleId}\",",
                    "\"Infoblox RuleCategory\":\"${L:A:RuleCategory}
\",",
                    "\"Infoblox Threat Status\":\"Unresolved\"",
                "}",
                "}"
            ]
        },
        {
            "name": "assign profiler values",
            "operation": "POST",
            "parse": "JSON",
            "headers": {
                "Content-Type": "application/json",
                "User-Agent": "Infoblox Security Integration",
                "Accept": "*/*"
            },
            "transport": {
                "path": "/async_netd/deviceprofiler/endpoints"
            },
            "body_list": [
                "{",
                "\"mac\":\"${L:A:MacFull}\",",
                "\"ip\": \"${L:A:address}\",",
                "\"hostname\": \"${L:A:host}\",",
                "\"device\":{",
                    "\"family\":\"${L:A:vendor}\",",
```

```
                    "\"category\":\"${L:A:type}\",",
                    "\"name\":\"${L:A:name}\"",
                "}",
                "}"
            ]
        },
        {
            "name": "jump to update infoblox record",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "next": "Update timestamp"
            }
        },

         {
            "name": "Add endpoint event check",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "==",
                        "right": "ADP"
                    }
                ],
                "condition_type": "AND",
                "next": "Add an endpoint"
            }
        },
         {
            "name": "Debug ruleid2",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
         {
            "name": "check if ruleid found",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${L:A:rule_id}",
                        "op": "!=",
                        "right":""
                    }
```

```
                ],
                  "condition_type": "AND",
                 "eval": "${XC:COPY:{L:RuleId}:{L:rule_id}}",
                 "else_eval": "${XC:ASSIGN:{L:RuleId}:{I:99999}}"
             }
            },
            {
            "name": "check if rule_category found",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${L:A:rule_category}",
                        "op": "!=",
                        "right":""
                    }
                ],
                  "condition_type": "AND",
                 "eval": "${XC:COPY:{L:RuleCategory}:
{L:rule_category}}",
                 "else_eval": "${XC:ASSIGN:{L:RuleCategory}:
{S:unknown}}"
             }
            },

        {
            "name": "Add an endpoint",
            "operation": "PUT",
            "parse": "JSON",
            "headers": {
                "Authorization": "Bearer ${S:A:SESSID}"
            },
            "transport": {
                "path": "/api/endpoint/mac-address/${L:A:Mac1}$
{L:A:Mac2}${L:A:Mac3}${L:A:Mac4}${L:A:Mac5}${L:A:Mac6}"
            },
            "body_list": [
                "{",
                "\"mac_address\":\"${L:A:MacFull}\",",
                "\"status\":\"Known\",",
                "\"description\":\"Added via API at ${UT:A:TIME}
\",",
                "\"attributes\":{",
                    "\"client_hostname\":\"${L:A:host}\",",
                    "\"Device Type\":\"${L:A:type}\",",
                    "\"Device Vendor\":\"${L:A:vendor}\",",
                    "\"Location\":\"${L:A:Location}\",",
                    "\"Model\":\"${L:A:model}\",",
                    "\"Infoblox Last Known IP\":\"$
{L:A:Infoblox_Last_Known_IP}\",",
                    "\"OS Version\":\"${L:A:os_version}\",",
                    "\"Infoblox Managed\":\"${L:A:managed}\",",
                    "\"Infoblox DHCP Fingerprint\":\"$
{L:A:fingerpring}\",",
```

```
                    "\"Infoblox Threat Category\":\"$
{L:A:ThreatCategory}\",",
                    "\"Infoblox Threat Detection Device IP\":\"$
{L:A:ThreatDetection}\",",
                    "\"Infoblox Threat Name\":\"${L:A:ThreatName}
\",",
                    "\"Infoblox Threat Severity\":\"$
{I:A:ThreatSeverity}\",",
                    "\"Infoblox RuleId\":\"${L:A:RuleId}\",",
                    "\"Infoblox RuleCategory\":\"${L:A:RuleCategory}
\",",
                    "\"Infoblox Threat Status\":\"Unresolved\"",
                "}",
                "}"
            ]
        },
        {
            "name": "Update timestamp",
            "operation": "PUT",
            "transport": {
                "path": "${L:A:Obj_ref}"
            },
            "wapi": "v2.7",
            "wapi_quoting": "JSON",
            "body_list": [
                "{\"extattrs+\":{\"Aruba_LastSecurityEvent\":
{ \"value\": \"${UT:A:TIME}\"}}}"
            ]
        },
        {
            "name": "Stop everthing",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "stop": true
            }
        }
    ]
}
```