```
{
    "vendor_identifier": "Aruba ClearPass",
    "version": "4.0",
    "name": "Aruba ClearPass Assets",
    "content_type": "application/json",
    "type": "REST_EVENT",
    "event_type": [
        "LEASE",
        "FIXED_ADDRESS_IPV4",
        "HOST_ADDRESS_IPV4",
        "FIXED_ADDRESS_IPV6",
        "HOST_ADDRESS_IPV6",
        "DISCOVERY_DATA"
    ],
    "headers": {
        "Accept": "*/*"
    },
    "instance_variables": [
    ],
    "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "skip object modification and deletion",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:operation_type}",
                        "op": "==",
                        "right": "DELETE"
                    }
                ],
                "condition_type": "OR",
                "stop": true
            }
        },
        {
            "name": "check if lease",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "==",
                        "right": "LEASE"
                    }
                ],
                "eval": "${XC:COPY:{L:address}:{E:address}}",
```

```
                "next": "all discovery information",
                "else_next": "assignLVarsNet from E:",
                "condition_type": "OR"
            }
        },
        {
            "name": "assignLVarsNet from E:",
            "operation": "NOP",
            "body_list": [
                "${XC:COPY:{L:timestamp}:{E:timestamp}}",
                "${XC:COPY:{L:network_view}:
{E:values{network_view}}}"
            ]
        },
        {
            "name": "check if IPv4 or IPv6 for assigning variables",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:values{ipv4addr}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:address}:{E:values{ipv4addr}}}
${XC:ASSIGN:{L:addr}:{S:ipv4addr}}${XC:ASSIGN:{L:fixed}:
{S:fixedaddress}}",
                "else_eval": "${XC:COPY:{L:address}:
{E:values{ipv6addr}}}${XC:ASSIGN:{L:addr}:{S:ipv6addr}}${XC:ASSIGN:
{L:fixed}:{S:ipv6fixedaddress}}"
            }
        },
        {
            "name": "Check if location",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${E:A:values{extattrs}
{Aruba_Location}{value}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:name}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:Location}:
{E:values{extattrs}{Aruba_Location}{value}}}"
            }
        },
        {
            "name": "assignTimeValue",
```

```
            "operation": "NOP",
            "body_list": [
                "${XC:COPY:{L:ArubaAddDate}:{UT:TIME}}$
{XC:FORMAT:TRUNCATE:{L:ArubaAddDate}:{16t}}"
            ]
        },
        {
            "name": "Set Old_Time",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${E:A:values{extattrs}
{Aruba_SyncedAt}{value}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:ArubaAddDateRecorded}:
{S:}}",
                "else_eval": "${XC:COPY:{L:ArubaAddDateRecorded}:
{E:values{extattrs}{Aruba_SyncedAt}{value}}}${XC:FORMAT:TRUNCATE:
{L:ArubaAddDateRecorded}:{16t}}"
            }
        },
        {
            "name": "check If Scan Happened today",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${E:A:values{extattrs}{Aruba_Sync}
{value}}",
                        "op": "==",
                        "right": "false"
                    },
                    {
                        "left": "${L:A:ArubaAddDateRecorded}",
                        "op": "==",
                        "right": "${L:A:ArubaAddDate}"
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "all discovery information",
            "operation": "GET",
            "transport": {
                "path": "discovery:device?address=${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": "Debug#40",
            "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 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": "Debug#41",
            "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 for description",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{description}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:description}:{S:None}}",
                "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:ASSIGN:{L:os_version}:{S:Unknown}}",
                "else_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": "${XC:ASSIGN:{L:model}:{S:Unknown}}",
                "else_eval": "${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:ASSIGN:{L:vendor}:{S:Unknown}}",
                "else_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": "${XC:ASSIGN:{L:type}:{S:Unknown}}",
                "else_eval": "${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 lease to jump to lease event",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "==",
                        "right": "LEASE"
                    }
                ],
                "condition_type": "AND",
                "next": "Check if Lease is wanted"
            }
        },
        {
            "name": "stop add if Asset is not wanted",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${E:A:values{extattrs}{Aruba_Sync}
{value}}",
                        "op": "!=",
                        "right": "true"
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "Debug#11",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Debug#12",
            "operation": "NOP",
            "body": "${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 host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "=~",
                        "right": "HOST"
```

```
                    }
                ],
                "condition_type": "AND",
                "next": "check if IPv4 or IPv6 for host"
            }
        },
        {
            "name": "Debug#13",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "check if IPv4 or IPv6",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${L:A:addr}",
                        "op": "==",
                        "right": "ipv6addr"
                    }
                ],
                "condition_type": "AND",
                "next": "Get Fixed IPv6 Mac"
            }
        },
        {
            "name": "Debug#14",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Get Fixed IPv4 Mac",
            "operation": "GET",
            "parse": "JSON",
            "transport": {
                "path": "${E:A:values{_ref}}?
_return_fields=mac,discovered_data.mac_address,discovered_data.vmhos
t_mac_address,discovered_data.vport_mac_address"
            },
            "wapi": "v2.7",
            "result": [{
                "codes": "200,201,202,203,204",
                "next": "check if mac is present"
            }]
        },
        {
            "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:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Get Fixed IPv6 Mac",
            "operation": "GET",
            "parse": "JSON",
            "transport": {
                "path": "${E:A:values{_ref}}?
_return_fields=discovered_data.mac_address,discovered_data.vmhost_ma
c_address,discovered_data.vport_mac_address"
            },
            "wapi": "v2.7",
            "result": [{
                "codes": "200,201,202,203,204",
                "next": "check if discovered mac_address is present"
            }]
        },
        {
            "name": "Debug#16",
            "operation": "NOP",
            "body": "${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 mac is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:{P:mac}}",
                "next": "assignMac from L: for fixed"
            }
        },
        {
            "name": "Debug#17",
            "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 discovered mac_address is present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
```

```
                        "left": "$
{P:A:discovered_data.mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.mac_address}}",
                "next": "assignMac from L: for fixed"
            }
        },
        {
            "name": "check if discovered vmhost_mac_address is
present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "$
{P:A:discovered_data.vmhost_mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.vmhost_mac_address}}",
                "next": "assignMac from L: for fixed"
            }
        },
        {
            "name": "check if discovered vport_mac_address is
present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "$
{P:A:discovered_data.vport_mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.vport_mac_address}}",
                "next": "assignMac from L: for fixed"
            }
        },
        {
            "name": "Debug#18",
            "operation": "NOP",
            "body": "${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": "Stop if no mac for fixed",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "stop": true
            }
        },
        {
            "name": "Debug#19",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "assignMac from L: for fixed",
            "operation": "NOP",
            "body_list": [
                "${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#20",
            "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 Fixed",
            "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": "Skip if modify event and no mac address
with Fixed"
            }]
        },
        {
            "name": "Debug#21",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Skip if modify event and no mac address with
Fixed",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac_address}",
                        "op": "!=",
                        "right": ""
                    },
                    {
                        "left": "${E:A:operation_type}",
                        "op": "==",
                        "right": "MODIFY"
                    }
                ],
                "condition_type": "AND",
                "next": "check for Location"
            }
        },
        {
            "name": "Stop everthing if mac isn't present with
Fixed",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
```

```
                "condition_type": "OR",
                "stop": true
            }
        },
        {
            "name": "Debug#22",
            "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": "Add an endpoint from a Fixed",
            "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 ${UT:A:TIME}
\",",
                    "\"attributes\":{",
                        "\"Device Type\":\"${L:A:type}\",",
                        "\"Device Vendor\":\"${L:A:vendor}\",",
                        "\"Location\":\"${L:A:Location}\",",
                        "\"Model\":\"${L:A:model}\",",
                        "\"Infoblox DHCP Fingerprint\":
\"Unknown\",",
                        "\"Infoblox Managed\":\"True\",",
                        "\"Infoblox Last Known IP\":\"${L:A:address}
\",",
                        "\"OS Version\":\"${L:A:os_version}\"",
                    "}",
                "}"
            ]
        },
        {
            "name": "Testing fixed",
            "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}\",",
                "\"device\":{",
                    "\"family\":\"${L:A:vendor}\",",
                    "\"category\":\"${L:A:type}\",",
                    "\"name\":\"${L:A:name}\"",
                "}",
                "}"
            ]
        },
        {
            "name": "Update extattrs for update fixed ip",
            "operation": "PUT",
            "transport": {
                "path": "${E:A:values{_ref}}"
            },
            "wapi": "v2.7",
            "wapi_quoting": "JSON",
            "body_list": [
                "{\"extattrs+\":{\"Aruba_SyncedAt\": { \"value\":
\"${L:A:timestamp}\"}}}"
            ]
        },
        {
            "name": "end of adding a Fixed",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "next": "Stop everthing"
            }
        },
        {
            "name": "check if IPv4 or IPv6 for host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${L:A:addr}",
                        "op": "==",
                        "right": "ipv6addr"
                    }
                ],
                "condition_type": "AND",
                "next": "Get Host_IPv6 information"
            }
```

```
        },
        {
            "name": "Debug#5",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Get Host_IPv4 information",
            "operation": "GET",
            "parse": "JSON",
            "transport": {
                "path": "${E:A:values{_ref}}?
_return_fields=mac,discovered_data.mac_address,discovered_data.vmhos
t_mac_address,discovered_data.vport_mac_address"
            },
            "wapi": "v2.7",
            "result": [{
                "codes": "200,201,202,203,204",
                "next": "check if mac is present for host"
            }]
        },
        {
            "name": "Debug#6",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Get Host_IPv6 information",
            "operation": "GET",
            "parse": "JSON",
            "transport": {
                "path": "${E:A:values{_ref}}?
_return_fields=discovered_data.mac_address,discovered_data.vmhost_ma
c_address,discovered_data.vport_mac_address"
            },
            "wapi": "v2.7",
            "result": [{
                "codes": "200,201,202,203,204",
                "next": "check if discovered mac_address is present
for host"
            }]
        },
        {
            "name": "Debug#7.1",
            "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 mac is present for host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:{P:mac}}",
                "next": "assignMac from P: for host"
            }
        },
        {
            "name": "Debug#7.2",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "check if discovered mac_address is present for
host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "$
{P:A:discovered_data.mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.mac_address}}",
                "next": "assignMac from P: for host"
            }
        },
        {
            "name": "Debug#7.3",
            "operation": "NOP",
            "body": "${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 discovered vmhost_mac_address is
present for host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
```

```
                    {
                        "left": "$
{P:A:discovered_data.vmhost_mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.vmhost_mac_address}}",
                "next": "assignMac from P: for host"
            }
        },
        {
            "name": "Debug#7.4",
            "operation": "NOP",
            "body": "${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 discovered vport_mac_address is
present for host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "$
{P:A:discovered_data.vport_mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:mac}:
{P:discovered_data.vport_mac_address}}",
                "next": "assignMac from P: for host"
            }
        },
        {
            "name": "Debug#7.5",
            "operation": "NOP",
            "body": "${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": "Stop if no mac for fixed for host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
```

```
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "stop": true
            }
        },
        {
            "name": "Debug#7.6",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "assignMac from P: for host",
            "operation": "NOP",
            "body_list": [
                "${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#8",
            "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": "Skip if modify event and no mac address"
```

```
            }]
        },
        {
            "name": "Skip if modify event and no mac address",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac_address}",
                        "op": "!=",
                        "right": ""
                    },
                    {
                        "left": "${E:A:operation_type}",
                        "op": "==",
                        "right": "MODIFY"
                    }
                ],
                "condition_type": "AND",
                "next": "check for Location"
            }
        },
        {
            "name": "Debug#9",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Stop everthing if mac isn't present with host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "OR",
                "stop": true
            }
        },
        {
            "name": "Debug#10",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Add an endpoint from a host",
            "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 ${UT:A:TIME}
\",",
                "\"attributes\":{",
                    "\"client_hostname\":\"${E:A:values{host}}\",",
                    "\"Device Type\":\"${L:A:type}\",",
                    "\"Device Vendor\":\"${L:A:vendor}\",",
                    "\"Location\":\"${L:A:Location}\",",
                    "\"Model\":\"${L:A:model}\",",
                    "\"Infoblox DHCP Fingerprint\":\"Unknown\",",
                    "\"Infoblox Managed\":\"True\",",
                    "\"Infoblox Last Known IP\":\"${L:A:address}
\",",
                    "\"OS Version\":\"${L:A:os_version}\"",
                "}",
                "}"
            ]
        },
        {
            "name": "Testing host",
            "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\": \"${E:A:values{host}}\",",
                "\"device\":{",
                    "\"family\":\"${L:A:vendor}\",",
                    "\"category\":\"${L:A:type}\",",
                    "\"name\":\"${L:A:name}\"",
                "}",
                "}"
            ]
        },
        {
```

```
            "name": "check if IPv4 or IPv6 to get Host for update",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${L:A:addr}",
                        "op": "==",
                        "right": "ipv6addr"
                    }
                ],
                "condition_type": "AND",
                "next": "Get HostIPv6 _ref"
            }
        },
        {
            "name": "Get HostIPv4 _ref",
            "operation": "GET",
            "transport": {
                "path": "record:host?ipv4addr=${L:U:address}
&network_view=${L:U:network_view}&_return_fields=extattrs"
            },
            "wapi": "v2.7",
            "result": [{
                "codes": "200,201,202,203,204",
                "next": "Update extattrs for update Host"
            }]
        },
        {
            "name": "Get HostIPv6 _ref",
            "operation": "GET",
            "transport": {
                "path": "record:host?ipv6addr=${L:U:address}
&network_view=${L:U:network_view}&_return_fields=extattrs"
            },
            "wapi": "v2.7"
        },
        {
            "name": "Update extattrs for update Host",
            "operation": "PUT",
            "transport": {
                "path": "${P:A:PARSE[0]{_ref}}"
            },
            "wapi": "v2.7",
            "wapi_quoting": "JSON",
            "body_list": [
                "{\"extattrs+\":{\"Aruba_SyncedAt\": { \"value\":
\"${L:A:timestamp}\"}}}"
            ]
        },
        {
            "name": "end of adding a host",
            "operation": "CONDITION",
            "condition": {
                "statements": [
```

```
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "next": "Stop everthing"
            }
        },









        {
            "name": "Check if Lease is wanted",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:ip.extattrs{Aruba_Sync}}",
                        "op": "==",
                        "right": "true"
                    }
                ],
                "condition_type": "OR",
                "else_stop": true
            }
        },
        {
            "name": "Check if location for lease",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "$
{E:A:ip.extattrs{Aruba_Location}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:Location}:{S:Unknown}}",
```

```
                "else_eval": "${XC:COPY:{L:Location}:
{E:ip.extattrs{Aruba_Location}}}"
            }
        },
        {
            "name": "assignMac from E: for lease",
            "operation": "NOP",
            "body_list": [
                "${XC:COPY:{L:Mac1}:{E:hardware}}$
{XC:FORMAT:TRUNCATE:{L:Mac1}:{2t}}",
                "${XC:COPY:{L:Mac2}:{E:hardware}}$
{XC:FORMAT:TRUNCATE:{L:Mac2}:{5t}}${XC:FORMAT:TRUNCATE:{L:Mac2}:
{-2f}}",
                "${XC:COPY:{L:Mac3}:{E:hardware}}$
{XC:FORMAT:TRUNCATE:{L:Mac3}:{8t}}${XC:FORMAT:TRUNCATE:{L:Mac3}:
{-2f}}",
                "${XC:COPY:{L:Mac4}:{E:hardware}}$
{XC:FORMAT:TRUNCATE:{L:Mac4}:{11t}}${XC:FORMAT:TRUNCATE:{L:Mac4}:
{-2f}}",
                "${XC:COPY:{L:Mac5}:{E:hardware}}$
{XC:FORMAT:TRUNCATE:{L:Mac5}:{14t}}${XC:FORMAT:TRUNCATE:{L:Mac5}:
{-2f}}",
                "${XC:COPY:{L:Mac6}:{E:hardware}}$
{XC:FORMAT:TRUNCATE:{L:Mac6}:{-2f}}",
                "${XC:COPY:{L:MacFull}:{E:hardware}}"
            ]
        },
        {
            "name": "Debug#1",
            "operation": "NOP",
            "body": "${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",
            "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": "Stop everthing if mac isn't present"
            }]
        },
        {
            "name": "Debug#1b",
            "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": "Stop everthing if mac isn't present",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${P:A:mac_address}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "condition_type": "AND",
                "stop": true
            }
        },
        {
            "name": "Get Lease information",
            "operation": "GET",
            "parse": "JSON",
            "transport": {
                "path": "lease?address=${E:A:address}&network_view=$
{E:A:network_view}&_return_fields=fingerprint"
            },
            "wapi": "v2.7",
            "result": [{
                "codes": "200,201,202,203,204",
                "next": "Check if fingerprint is unknown"
            }]
        },
        {
            "name": "Debug#3",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Check if fingerprint is unknown",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:PARSE[0]{fingerprint}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:fingerprint}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:fingerprint}:{P:PARSE[0]
{fingerprint}}}"
```

```
            }
        },

        {
            "name": "Add an endpoint",
            "operation": "POST",
            "parse": "JSON",
            "headers": {
                "Authorization": "Bearer ${S:A:SESSID}"
            },
            "transport": {
                "path": "/api/endpoint"
            },
            "body_list": [
                "{",
                    "\"mac_address\":\"${E:A:hardware}\",",
                    "\"status\":\"Known\",",
                    "\"description\":\"Added via API at ${UT:A:TIME}
\",",
                    "\"attributes\":{",
                        "\"client_hostname\":\"$
{E:A:client_hostname}\",",
                        "\"Device Type\":\"${L:A:type}\",",
                        "\"Device Vendor\":\"${L:A:vendor}\",",
                        "\"Location\":\"${L:A:Location}\",",
                        "\"Model\":\"${L:A:model}\",",
                        "\"Infoblox DHCP Fingerprint\":\"$
{L:A:fingerprint}\",",
                        "\"Infoblox Managed\":\"True\",",
                        "\"Infoblox Last Known IP\":\"${E:A:address}
\",",
                        "\"OS Version\":\"${L:A:os_version}\"",
                    "}",
                    "}"
            ]
        },
        {
            "name": "Testing lease",
            "operation": "POST",
            "parse": "JSON",
            "headers": {
                "Content-Type": "application/json",
                "User-Agent": "Infoblox Security Integration",
                "Accept": "*/*"
            },
            "transport": {
                "path": "/async_netd/deviceprofiler/endpoints"
            },
            "body_list": [
                "{",
                "\"mac\":\"${L:A:MacFull}\",",
                "\"ip\": \"${E:A:address}\",",
                "\"device\":{",
                    "\"family\":\"${L:A:vendor}\",",
```

```
                    "\"category\":\"${L:A:type}\",",
                    "\"name\":\"${L:A:name}\"",
                "}",
                "}"
            ]
        },
        {
            "name": "Debug4",
            "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": "Stop everthing",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "stop": 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": "${XC:COPY:{L:Location}:
{P:attributes{Location}}}"
            }
        },
        {
            "name": "check for Fingerprint",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${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": "${XC:COPY:{L:host}:
{P:attributes{client_hostname}}}"
            }
        },
        {
            "name": "check for client_hostname if host event",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "=~",
                        "right": "HOST"
                    }
                ],
                "eval": "${XC:COPY:{L:host}:{E:values{host}}}"
            }
        },
        {
            "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 Threat Category",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
```

```
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox Threat
Category}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:ThreatCategory}:
{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:ThreatCategory}:
{P:attributes{Infoblox Threat Category}}}"
            }
        },
        {
            "name": "check for Threat Detection Device IP",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox Threat
Detection Device IP}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:ThreatDetection}:
{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:ThreatDetection}:
{P:attributes{Infoblox Threat Detection Device IP}}}"
            }
        },
        {
            "name": "check for Threat Name",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox Threat
Name}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:ThreatName}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:ThreatName}:
{P:attributes{Infoblox Threat Name}}}"
            }
        },
        {
            "name": "check for Threat Severity",
            "operation": "CONDITION",
```

```
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox Threat
Severity}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:ThreatSeverity}:
{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:ThreatSeverity}:
{P:attributes{Infoblox Threat Severity}}}"
            }
        },
        {
            "name": "check for Threat Status",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox Threat
Status}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:ThreatStatus}:{S:Other}}",
                "else_eval": "${XC:COPY:{L:ThreatStatus}:
{P:attributes{Infoblox Threat Status}}}"
            }
        },
        {
            "name": "Check if name is unknown for modify events",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${P:A:attributes{name}}",
                        "op": "==",
                        "right": ""
                    },
                    {
                        "left": "${P:A:attributes{name}}",
                        "op": "==",
                        "right": "unknown"
                    }
                ],
                "eval": "${XC:ASSIGN:{L:name}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:name}:
{P:attributes{name}}}"
```

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

```
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "$
{P:A:attributes{active_users_count}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:active_users_count}:
{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:active_users_count}:
{P:attributes{ms_ad_user_data}{active_users_count}}}"
            }
        },
        {
            "name": "check for vendor for modify events",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{vendor}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:vendor}:
{P:attributes{vendor}}}"
            }
        },
        {
            "name": "Debug Ruleid",
            "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:}}${XC:DEBUG:{UT:}}"
        },
        {
            "name": "Check if Ruleid is unknown",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox
RuleId}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:RuleId}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:RuleId}:
```

```
{P:attributes{Infoblox RuleId}}}"
            }
        },
        {
            "name": "Check if RuleCategory is unknown",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{Infoblox
RuleCategory}}",
                        "op": "==",
                        "right": ""
                    }
                ],
                "eval": "${XC:ASSIGN:{L:RuleCategory}:{S:Unknown}}",
                "else_eval": "${XC:COPY:{L:RuleCategory}:
{P:attributes{Infoblox RuleCategory}}}"
            }
        },
         {
            "name": "Debug RuleCategory",
            "operation": "NOP",
            "body": "${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 for type for modify events",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${P:A:attributes{type}}",
                        "op": "!=",
                        "right": ""
                    }
                ],
                "eval": "${XC:COPY:{L:type}:{P:attributes{type}}}"
            }
        },
```

```
        {
            "name": "Get all discovery information for modify
events",
            "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": "Modify 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\":\"${L:A:address}
\",",
                    "\"OS Version\":\"${L:A:os_version}\",",
                    "\"Infoblox Managed\":\"True\",",
                    "\"Infoblox DHCP Fingerprint\":\"$
{L:A:fingerpring}\",",
                    "\"Infoblox Threat Category\":\"$
{L:A:ThreatCategory}\",",
                    "\"Infoblox Threat Detection Device IP\":\"$
{L:A:ThreatDetection}\",",
                    "\"Infoblox Threat Name\":\"${L:A:ThreatName}
\",",
                    "\"Infoblox Threat Severity\":\"$
{L:A:ThreatSeverity}\",",
                    "\"Infoblox Threat Status\":\"$
{L:A:ThreatStatus}\",",
                    "\"Infoblox RuleId\":\"${L:A:RuleId}\",",
```

```
                    "\"Infoblox RuleCategory\":\"$
{L:A:RuleCategory}\"",

                "}",
                "}"
            ]
        },
        {
            "name": " modify",
            "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": "Stop if lease event",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
                        "op": "==",
                        "right": "LEASE"
                    }
                ],
                "condition_type": "OR",
                "stop": true
            }
        },
        {
            "name": "skip if fixed event to update information",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${E:A:event_type}",
```

```
                    "op": "=~",
                    "right": "FIXED"
                }
            ],
            "condition_type": "OR",
            "next": "Update extattrs for update fixed ip"
        }
    },
    {
        "name": "check if IPv4 or IPv6 to get Host for update
for modify events",
        "operation": "CONDITION",
        "condition": {
            "statements": [
                {
                    "left": "${L:A:addr}",
                    "op": "==",
                    "right": "ipv6addr"
                }
            ],
            "condition_type": "AND",
            "next": "Get HostIPv6 _ref"
        }
    },
    {
        "name": "Get HostIPv4 _ref for modify events",
        "operation": "GET",
        "transport": {
            "path": "record:host?ipv4addr=${L:U:address}
&network_view=${L:U:network_view}&_return_fields=extattrs"
        },
        "wapi": "v2.7",
        "result": [{
            "codes": "200,201,202,203,204",
            "next": "Update extattrs for update Host"
        }]
    },
    {
        "name": "Get HostIPv6 _ref for modify events",
        "operation": "GET",
        "transport": {
            "path": "record:host?ipv6addr=${L:U:address}
&network_view=${L:U:network_view}&_return_fields=extattrs"
        },
        "wapi": "v2.7"
    },
    {
        "name": "Update extattrs for update Host for modify
events",
        "operation": "PUT",
        "transport": {
            "path": "${P:A:PARSE[0]{_ref}}"
        },
        "wapi": "v2.7",
```

```
            "wapi_quoting": "JSON",
            "body_list": [
                "{\"extattrs+\":{\"Aruba_SyncedAt\": { \"value\":
\"${L:A:timestamp}\"}}}"
            ]
        },
        {
            "name": "Stop host update for modify events",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "stop": true
            }
        },
        {
            "name": "Update extattrs for update fixed ip for modify
events",
            "operation": "PUT",
            "transport": {
                "path": "${E:A:values{_ref}}"
            },
            "wapi": "v2.7",
            "wapi_quoting": "JSON",
            "body_list": [
                "{\"extattrs+\":{\"Aruba_SyncedAt\": { \"value\":
\"${L:A:timestamp}\"}}}"
            ]
        },
        {
            "name": "Stop fixedIP for modify events",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "condition_type": "AND",
                "stop": true
            }
        }



    ]
```

```
}
```