```
{
    "name": "DXL_Lease_Events",
    "version": "3.0",
    "type": "DXL_EVENT",
    "event_type": [
        "LEASE"
    ],
    "vendor_identifier": "McAfee",
    "quoting": "ASIS",
    "instance_variables": [
        {
            "name": "DXL_MessageFormat",
            "type": "STRING"
        }
    ],
    "steps": [
        {
            "name": "STOP if sync not requested",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "${E:A:ip.extattrs{DXL_Sync}}",
                        "op": "==",
                        "right": ""
                    },
                    {
                        "left": "${E:A:ip.extattrs{DXL_Sync}}",
                        "op": "==",
                        "right": "false"
                    }
                ],
                "stop": true
            }
        },
        {
            "name": "init_internal_data",
            "operation": "VARIABLEOP",
            "variable_ops": [
                {
                    "operation": "ASSIGN",
```

```
                "type": "DICTIONARY",
                "destination": "L:internal",
                "keys": [
                    "analyzer_ipv4",
                    "analyzer_ipv6",
                    "source_ipv4",
                    "source_ipv6",
                    "target_ipv4",
                    "target_ipv6",
                    "severity"
                ],
                "values": [
                    "",
                    "",
                    "",
                    "",
                    "",
                    "",
                    "7"
                ]
            }
        ]
    },
    {
        "name": "is_analyzer_source_LEASE_ipv4",
        "operation": "CONDITION",
        "condition": {
            "statements": [
                {
                    "left": "${E::event_type}",
                    "op": "==",
                    "right": "LEASE"
                },
                {
                    "left": "${E::address}",
                    "op": "!~",
                    "right": ":"
                }
            ],
            "condition_type": "AND",
            "eval": "${XC:COPY:{L:internal{analyzer_ipv4}}:
{E:member_ip}}${XC:COPY:{L:internal{source_ipv4}}:
```

```
{E:member_ip}}${XC:ASSIGN:{L:IPv}:{I:4}}",
            "else_eval": "${XC:COPY:{L:internal{analyzer_ipv6}}:
{E:member_ip}}${XC:COPY:{L:internal{source_ipv6}}:
{E:member_ip}}${XC:ASSIGN:{L:IPv}:{I:6}}"
          }
       },
       {
          "name": "set up address",
          "operation": "NOP",
          "body_list": [
             "${XC:COPY:{L:IP}:{E:address}}"
          ]
       },
       {
          "name": "Get User Data",
          "operation": "GET",
          "transport": {
             "path": "networkuser?user_status=ACTIVE&address=${L:A:IP}"
          },
          "wapi": "v2.6"
       },
       {
          "name": "check_user_response",
          "operation": "CONDITION",
          "condition": {
             "condition_type": "AND",
             "statements": [
                {
                   "left": "${P:L:PARSE}",
                   "op": "==",
                   "right": "0"
                }
             ],
             "next": "check_username"
          }
       },
       {
          "name": "Pop User from the list",
          "operation": "VARIABLEOP",
          "variable_ops": [
             {
                "operation": "UNSHIFT",
```

```
                "type": "DICTIONARY",
                "destination": "L:user",
                "source": "P:PARSE"
            }
        ]
    },
    {
        "name": "check_username",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "AND",
            "statements": [
                {
                    "left": "${L::user{name}}",
                    "op": "!=",
                    "right": ""
                }
            ],
            "eval": "${XC:COPY:{L:username}:{L:user{name}}}${XC:COPY:
{L:domainname}:{L:user{domainname}}}",
            "else_eval": "${XC:ASSIGN:{L:username}:{S:.}}${XC:ASSIGN:
{L:domainname}:{S:.}}"
        }
    },
    {
        "name": "check if lease IPv4 event to assign target_ipv4",
        "operation": "CONDITION",
        "condition": {
            "statements": [
                {
                    "left": "${E::event_type}",
                    "op": "==",
                    "right": "LEASE"
                },
                {
                    "left": "${E::range_start_addr}",
                    "op": "!~",
                    "right": ":"
                }
            ],
            "condition_type": "AND",
            "eval": "${XC:COPY:{L:internal{target_ipv4}}:{E:address}}}",
```

```
        "next": "is_severity_7"
      }
    },
    {
      "name": "check if lease IPv6 event to assign 6",
      "operation": "CONDITION",
      "condition": {
        "statements": [
          {
            "left": "${E::event_type}",
            "op": "==",
            "right": "LEASE"
          }
        ],
        "condition_type": "AND",
        "eval": "${XC:COPY:{L:internal{target_ipv6}}:{E:address}}}"
      }
    },
    {
      "name": "is_severity_7",
      "operation": "CONDITION",
      "condition": {
        "statements": [
          {
            "left": "1",
            "op": "==",
            "right": "1"
          }
        ],
        "condition_type": "AND",
        "eval": "${XC:ASSIGN:{L:internal{severity}}:{I:7}}"
      }
    },
    {
      "name": "check if lease event to assign values",
      "operation": "CONDITION",
      "condition": {
        "statements": [
          {
            "left": "${E::event_type}",
            "op": "==",
            "right": "LEASE"
```

```
                }
            ],
            "condition_type": "OR",
            "eval": "${XC:COPY:{L:ruleName}:
{E:member_name}}${XC:FORMAT:TRUNCATE:{L:ruleName}:
{-128f}}${XC:COPY:{L:threatName}:
{E:client_hostname}}${XC:FORMAT:TRUNCATE:{L:threatName}:
{-128f}}${XC:COPY:{L:DetectedUTC}:{E:timestamp}}${XC:ASSIGN:
{L:Obj_ref}:{S:}}${XC:ASSIGN:{L:network_view}:{S:default}}${XC:COPY:
{L:Object_type}:{E:event_type}}${XC:ASSIGN:{L:threatActionTaken}:
{S:Alert}}${XC:ASSIGN:{L:threatHandled}:{I:1}}${XC:COPY:
{L:operation_type}:{E:binding_state}}"
        }
    },
    {
        "name": "check GUID",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "OR",
            "statements": [
                {
                    "left": "${E:A:values{ip.extattrs}{ePO_GUID}{value}}",
                    "op": "==",
                    "right": ""
                }
            ],
            "eval": "${XC:COPY:{L:GUID}:{UT:UUID}}${XC:ASSIGN:
{L:GUIDtype}:{S:generated}}",
            "else_eval": "${XC:COPY:{L:GUID}:{E:values{ip.extattrs}
{ePO_GUID}{value}}}${XC:ASSIGN:{L:GUIDtype}:{S:local}}"
        }
    },
    {
        "name": "jump if have GUID or no WAPI credentials or is delete",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "OR",
            "statements": [
                {
                    "left": "${L:A:GUIDtype}",
                    "op": "==",
                    "right": "local"
```

```
                },
                {
                    "left": "${UT:A:WAPIUSERNAME}",
                    "op": "==",
                    "right": ""
                },
                {
                    "left": "${E:A:operation_type}",
                    "op": "==",
                    "right": "DELETE"
                }
            ],
            "next": "GET Lease data"
        }
    },
    {
        "name": "check if lease event to assign a different E: value to get
object_ref",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "AND",
            "statements": [
                {
                    "left": "${E:A:event_type}",
                    "op": "==",
                    "right": "LEASE"
                }
            ],
            "eval": "${XC:COPY:{L:GetIP}:{E:address}}",
            "next": "Get IPv4Fixed _ref"
        }
    },
    {
        "name": "assign ipv4 or ipv6 ip to use for GET requests",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "AND",
            "statements": [
                {
                    "left": "${E::values{ipv4addr}}",
                    "op": "!=",
                    "right": ""
```

```
                }
            ],
            "eval": "${XC:COPY:{L:GetIP}:{E:values{ipv4addr}}}",
            "else_eval": "${XC:COPY:{L:GetIP}:{E:values{ipv6addr}}"
        }
    },
    {
        "name": "check IPv6",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "OR",
            "statements": [
                {
                    "left": "${L:A:IPv}",
                    "op": "==",
                    "right": "6"
                }
            ],
            "next": "Get IPv6Fixed _ref"
        }
    },
    {
        "name": "Get IPv4Fixed _ref",
        "operation": "GET",
        "transport": {
            "path": "fixedaddress?ipv4addr=${L:U:GetIP}&network_view=
${L:U:network_view}&_return_fields=extattrs"
        },
        "wapi": "v2.7"
    },
    {
        "operation": "CONDITION",
        "name": "wapi_response_getIPv4Fix_ref",
        "condition": {
            "condition_type": "AND",
            "statements": [
                {
                    "left": "${P:A:PARSE[0]{_ref}}",
                    "op": "!=",
                    "right": ""
                }
            ],
```

```
            "next": "Get_Objref"
        }
    },
    {
        "name": "Get HostIPv4 _ref",
        "operation": "GET",
        "transport": {
            "path": "record:host?ipv4addr=${L:U:GetIP}&network_view=
${L:U:network_view}&_return_fields=extattrs"
        },
        "wapi": "v2.7"
    },
    {
        "operation": "CONDITION",
        "name": "wapi_response_getIPv4Host_ref",
        "condition": {
            "condition_type": "AND",
            "statements": [
                {
                    "left": "${P:A:PARSE[0]{_ref}}",
                    "op": "!=",
                    "right": ""
                }
            ],
            "next": "Get_Objref"
        }
    },
    {
        "name": "IPv4 object was not found",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "OR",
            "statements": [
                {
                    "left": "1",
                    "op": "==",
                    "right": "1"
                }
            ],
            "next": "GET Lease data"
        }
    },
```

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

```
                "right": ""
            }
        ],
        "next": "Get_Objref"
    }
},
{
    "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": "jump if no Obj_ref",
    "operation": "CONDITION",
    "condition": {
        "condition_type": "OR",
        "statements": [
            {
                "left": "${L:A:Obj_ref}",
                "op": "==",
                "right": ""
            }
        ],
        "next": "GET Lease data"
    }
},
{
    "name": "Update GUID",
    "operation": "PUT",
    "transport": {
        "path": "${L:A:Obj_ref}"
    },
```

```
            "wapi": "v2.7",
            "wapi_quoting": "JSON",
            "body_list": [
                "{\"extattrs+\":{\"ePO_GUID\": { \"value\": \"${L:A:GUID}\"},
\"DXL_LastEventSentAt\": { \"value\": \"${E:A:timestamp}\"}}}"
            ]
        },
        {
            "name": "GET Lease data",
            "operation": "GET",
            "transport": {
                "path": "lease?address=${E::address}&_return_fields=address,cl
ient_hostname,ipv6_duid,binding_state,cltt,ends,fingerprint,hardware,ipv6
_prefix_bits,is_invalid_mac,network,network_view,never_ends,never_start
s,protocol,served_by,server_host_name,starts,tstp,variable"¢           ÒÊ
            "wapi": "v2.7"
        },
        {
            "name": "Debug#43",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{UT:}}${XC:DEBUG:{R:}}"
        },
        {
            "name": "set Lease vars",
            "operation": "NOP",
            "body_list": [
                "${XC:COPY:{L:address}:{P:PARSE[0]{address}}}",
                "${XC:COPY:{L:binding_state}:{P:PARSE[0]{binding_state}}}",
                "${XC:COPY:{L:cltt}:{P:PARSE[0]{cltt}}}",
                "${XC:COPY:{L:ends}:{P:PARSE[0]{ends}}}",
                "${XC:COPY:{L:fingerprint}:{P:PARSE[0]{fingerprint}}}",
                "${XC:COPY:{L:hardware}:{P:PARSE[0]{hardware}}}",
                "${XC:COPY:{L:ipv6_prefix_bits}:{P:PARSE[0]
{ipv6_prefix_bits}}}",
                "${XC:COPY:{L:is_invalid_mac}:{P:PARSE[0]
{is_invalid_mac}}}",
                "${XC:COPY:{L:network}:{P:PARSE[0]{network}}}",
                "${XC:COPY:{L:network_view}:{P:PARSE[0]{network_view}}}",
                "${XC:COPY:{L:never_ends}:{P:PARSE[0]{never_ends}}}",
                "${XC:COPY:{L:never_starts}:{P:PARSE[0]{never_starts}}}",
```

```
            "${XC:COPY:{L:protocol}:{P:PARSE[0]{protocol}}}",
            "${XC:COPY:{L:served_by}:{P:PARSE[0]{served_by}}}",
            "${XC:COPY:{L:server_host_name}:{P:PARSE[0]
{server_host_name}}}",
            "${XC:COPY:{L:starts}:{P:PARSE[0]{starts}}}",
            "${XC:COPY:{L:extattrs}:{E:ip.extattrs{ePO_GUID}}}",
            "${XC:COPY:{L:variable}:{P:PARSE[0]{variable}}}"
         ]
      },
      {
         "name": "Check client_hostname",
         "operation": "CONDITION",
         "condition": {
            "condition_type": "AND",
            "statements": [
               {
                  "left": "${P::PARSE[0]{client_hostname}}",
                  "op": "!=",
                  "right": ""
               }
            ],
            "eval": "${XC:COPY:{L:client_hostname}:{P:PARSE[0]
{client_hostname}}}",
            "else_eval": "${XC:ASSIGN:{L:client_hostname}:{S:}}"
         }
      },
      {
         "name": "Check ipv6_duid",
         "operation": "CONDITION",
         "condition": {
            "condition_type": "AND",
            "statements": [
               {
                  "left": "${P::PARSE[0]{ipv6_duid}}",
                  "op": "!=",
                  "right": ""
               }
            ],
            "eval": "${XC:COPY:{L:ipv6_duid}:{P:PARSE[0]{ipv6_duid}}}",
            "else_eval": "${XC:ASSIGN:{L:ipv6_duid}:{S:}}"
         }
      },
```

```
        {
            "name": "check DXL_MessageFormat",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${I::DXL_MessageFormat}",
                        "op": "==",
                        "right": "CEF"
                    }
                ],
                "next": "send_CEF"
            }
        },
        {
            "name": "Debug#44",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{UT:}}${XC:DEBUG:{R:}}"
        },
        {
            "name": "send_OpenDXL",
            "operation": "DXL_SEND_EVENT",
            "body_list": [
                "{",
                "    \"eventMsgType\": \"Infoblox Change Event\",",
                "    \"eventMsgVersion\": \"1.0\",",
                "    \"event\": {",
                "        \"category\": \"${E::event_type}\",",
                "        \"eventDesc\": \"DNS ${E::event_type} ${E::binding_state}
event\",",
                "        \"eventType\": \"${E::binding_state}\",",
                "        \"eventId\": \"204162\",",
                "        \"analyzer\": {",
                "            \"id\": \"S_INFBLX0802\",",
                "            \"version\": \"8.2.1\",",
                "            \"name\": \"NIOS\",",
                "            \"detectionMethod\": \"NIOS\",",
                "            \"hostName\": \"${E::member_name}\",",
                "            \"detectedUTC\": \"${L::DetectedUTC}\",",
```

```
    "            \"ipv4\": \"${L::internal{analyzer_ipv4}}\",",
    "            \"ipv6\": \"${L::internal{analyzer_ipv6}}\"",
    "        },",
    "        \"entity\": {",
    "            \"groupName\": \"\",",
    "            \"osPlatform\": \"\",",
    "            \"osType\": \"\",",
    "            \"type\": \"\",",
    "            \"sessionID\": \"\",",
    "            \"address\": \"${L::address}\",",
    "            \"binding_state\": \"${L::binding_state}\",",
    "            \"cltt\": \"${L::cltt}\",",
    "            \"ends\": \"${L::ends}\",",
    "            \"fingerprint\": \"${L::fingerprint}\",",
    "            \"hardware\": \"${L::hardware}\",",
    "            \"ipv6_prefix_bits\": \"${L::ipv6_prefix_bits}\",",
    "            \"is_invalid_mac\": \"${L::is_invalid_mac}\",",
    "            \"network\": \"${L::network}\",",
    "            \"username\": \"${L::username}\",",
    "            \"domainname\": \"${L::domainname}\",",
    "            \"network_view\": \"${L::network_view}\",",
    "            \"never_ends\": \"${L::never_ends}\",",
    "            \"never_starts\": \"${L::never_starts}\",",
    "            \"protocol\": \"${L::protocol}\",",
    "            \"served_by\": \"${L::served_by}\",",
    "            \"server_host_name\": \"${L::server_host_name}\",",
    "            \"starts\": \"${L::starts}\",",
    "            \"client_hostname\": \"${L::client_hostname}\",",
    "            \"ipv6_duid\": \"${L::ipv6_duid}\",",
    "            \"variable\": \"${L::variable}\"",
    "        },",
    "            \"extattr\":{",
    "                \"ePO_GUID\": \"${L::extattrs}\"",
    "            },",
    "        \"source\": {",
    "            \"ipv4\": \"${L::internal{source_ipv4}}\",",
    "            \"ipv6\": \"${L::internal{source_ipv6}}\",",
    "            \"port\": 00000",
    "        }",
    "    }",
    "}"
],
```

```
            "dxl_topic": "/open/DDI/v1/${E::event_type}/infoblox"
        },
        {
            "name": "Debug#50",
            "operation": "NOP",
            "body": "${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:
{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}${XC:DEBUG:
{UT:}}${XC:DEBUG:{R:}}"
        },
        {
            "name": "goFin",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "next": "Fin"
            }
        },
        {
            "name": "send_CEF",
            "operation": "DXL_SEND_EVENT",
            "body_list": [
                "{\"DXLCommonEvent\":{",
                "\"category\": \"${E::event_type}\",",
                "\"eventDesc\": \"DNS ${E::event_type} ${E::binding_state}
event\",",
                "\"eventType\": \"${E::binding_state}\",",
                "\"eventId\": \"204162\",",
                "\"AgentGUID\": \"${L::GUID}\",",
                "\"Analyzer\":
\"${L::internal{analyzer_ipv4}}${L::internal{analyzer_ipv6}}\",",
                "\"AnalyzerDATVersion\": \"\",",
                "\"AnalyzerDetectionMethod\": \"${L::protocol}
${E::event_type}\",",
                "\"AnalyzerHostName\": \"${E::member_name}\",",
                "\"AnalyzerIPV4\": \"${L::internal{analyzer_ipv4}}\",",
```

```
            "\"AnalyzerIPV6\": \"${L::internal{analyzer_ipv6}}\",",
            "\"AnalyzerMAC\": \"\",",
            "\"AnalyzerName\": \"NIOS\",",
            "\"AnalyzerVersion\": \"8.2.1\",",
            "\"DetectedUTC\": \"${L::DetectedUTC}\",",
            "\"ServerID\":
\"${L::internal{analyzer_ipv4}}${L::internal{analyzer_ipv6}}\",",
            "\"SourceIPV4\": \"${L::internal{source_ipv4}}\",",
            "\"SourceIPV6\": \"${L::internal{source_ipv6}}\",",
            "\"SourcePort\": \"00000\",",
            "\"TargetHostName\": \"${E::member_name}\",",
            "\"TargetIPV4\": \"${L::internal{analyzer_ipv4}}\",",
            "\"TargetIPV6\": \"${L::internal{analyzer_ipv6}}\",",
            "\"TargetPort\": \"53\",",
            "\"TargetProtocol\": \"dns\",",
            "\"address\": \"${L::address}\",",
            "\"binding_state\": \"${L::binding_state}\",",
            "\"cltt\": \"${L::cltt}\",",
            "\"username\": \"${L::username}\",",
            "\"domainname\": \"${L::domainname}\",",
            "\"ends\": \"${L::ends}\",",
            "\"fingerprint\": \"${L::fingerprint}\",",
            "\"hardware\": \"${L::hardware}\",",
            "\"ipv6_prefix_bits\": \"${L::ipv6_prefix_bits}\",",
            "\"is_invalid_mac\": \"${L::is_invalid_mac}\",",
            "\"network\": \"${L::network}\",",
            "\"network_view\": \"${L::network_view}\",",
            "\"never_ends\": \"${L::never_ends}\",",
            "\"never_starts\": \"${L::never_starts}\",",
            "\"ePO_GUID\": \"${L::extattrs}\"",
            "\"protocol\": \"${L::protocol}\",",
            "\"served_by\": \"${L::served_by}\",",
            "\"server_host_name\": \"${L::server_host_name}\",",
            "\"starts\": \"${L::starts}\",",
            "\"client_hostname\": \"${L::client_hostname}\",",
            "\"ipv6_duid\": \"${L::ipv6_duid}\",",
            "\"variable\": \"${L::variable}\"",
            "}}"
        ],
        "dxl_topic": "/infoblox/outbound/${E::event_type}"
    },
    {
```

```
            "name": "goFin#2",
            "operation": "CONDITION",
            "condition": {
                "condition_type": "OR",
                "statements": [
                    {
                        "left": "1",
                        "op": "==",
                        "right": "1"
                    }
                ],
                "next": "Fin"
            }
        },
        {
            "name": "Fin",
            "operation": "NOP",
            "body": ""
        }
    ]
}
```