

```

{
  "name": "Fortinet Assets",
  "comment": "Fortinet Asset Events",
  "version": "4.0",
  "type": "REST_EVENT",
  "event_type": [
    "LEASE",
    "FIXED_ADDRESS_IPV4",
    "HOST_ADDRESS_IPV4",
    "FIXED_ADDRESS_IPV6",
    "HOST_ADDRESS_IPV6",
    "NETWORK_IPV4",
    "NETWORK_IPV6",
    "RANGE_IPV4",
    "RANGE_IPV6",
    "DISCOVERY_DATA"
  ],
  "action_type": "Fortinet Assets",
  "content_type": "application/json",
  "vendor_identifier": "Fortinet",
  "quoting": "XMLA",
  "steps": [
    {
      "name": "Debug Beginning",
      "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:}}"
    },

```

```

        {
"operation": "CONDITION",
"condition": {
  "condition_type": "OR",
  "statements": [
    {
      "left": "${E:A:event_type}",
      "op": "=",
      "right": "DISCOVERY_DATA"
    }
  ],
"eval": "${XC:COPY:{L:timestamp}:[E:timestamp]}${XC:COPY:{L:HostObjectType}:[E:object_type]}",
"next": "Check if IPv4 or IPv6 for assigning variables for discovery"
}
},
{
  "name": "Assign L variables from E namespace",
  "operation": "NOP",
  "body_list": [
    "${XC:COPY:{L:timestamp}:[E:timestamp]}",
    "${XC:COPY:{L:network_view}:[E:values{network_view}]}",
    "${XC:COPY:{L:HostObjectType}:[E:object_type]}"
  ]
},
{
"name": "Check for modify operation to skip",

```

```
        "comment": "Check for modify operation to skip",
"operation": "CONDITION",
"condition": {
  "statements": [
    {
      "left": "${E:A:operation_type}",
      "op": "=",
      "right": "MODIFY"
    }
  ],
  "condition_type": "OR",
  "stop": true
}
},
```

```
{
  "name": "Check for event type network or range",
  "comment": "Check for network or range",
  "operation": "CONDITION",
  "condition": {
    "condition_type": "OR",
    "statements": [
      {
        "left": "${E::event_type}",
        "op": "=~",
        "right": "NETWORK"
      },
      {
        "left": "${E::event_type}",
```

```

        "op": "=~",
        "right": "RANGE"
    }
],
"eval": "${XC:COPY:{L:HostIP}:{E:values{network}}}${XC:ASSIGN:{L:Mask}:{S:}}",
    "next": "Check for insert operation"
}
},
{
    "name": "Check for event type host address or fixed address",
    "comment": "Check for host address or fixed address",
    "operation": "CONDITION",
    "condition": {
        "condition_type": "OR",
        "statements": [
            {
                "left": "${E::event_type}",
                "op": "=~",
                "right": "FIXED"
            },
            {
                "left": "${E::event_type}",
                "op": "=~",
                "right": "HOST"
            }
        ],
        "next": "Check for IPv4 or IPv6 host or fixed"
    }
}

```

```

    },
    {
"name": "Check for IPv4 or IPv6 host or fixed",
        "operation": "CONDITION",
"condition": {
    "statements": [
        {
            "left": "${E:A:values{ipv4addr}}",
            "op": "!=",
            "right": ""
        }
    ],
    "condition_type": "AND",
    "eval": "${XC:COPY:{L:HostIP}:{E:values{ipv4addr}}}${XC:ASSIGN:{L:Mask}:{S:/32}}",
    "else_eval": "${XC:COPY:{L:HostIP}:{E:values{ipv6addr}}}${XC:ASSIGN:{L:Mask}:{S:/128}}"
        }
    },
    {
"name": "Check for insert operation",
"comment": "Check for insert operation",
"operation": "CONDITION",
"condition": {
        "condition_type": "AND",

"statements": [
        {
            "left": "${E::operation_type}",
            "op": "==",
            "right": "INSERT"
        }
    ]
    }

```

```

    ],
    "next": "Check if IPv4 or IPv6 for assigning variables for insert"
  }
},
  {
    "name": "Check for delete operation",
    "comment": "Check for delete operation",
    "operation": "CONDITION",
    "condition": {
      "condition_type": "AND",
      "statements": [
        {
          "left": "${E::operation_type}",
          "op": "==",
          "right": "DELETE"
        }
      ],
      "next": "Check if IPv4 or IPv6 for assigning variables for delete"
    }
  },
  {
    "name": "Check if IPv4 or IPv6 for assigning variables for delete",
    "operation": "CONDITION",
    "condition": {
      "statements": [
        {
          "left": "${L:A:HostIP}",
          "op": "=~",
          "right": ":"
        }
      ]
    }
  }
}

```

```

    }
  ],
  "condition_type": "AND",
  "eval":
"$XC:ASSIGN:{L:AddressType};{S:address6}}${XC:ASSIGN:{L:AddressGroup};{S:addrgrp6}}",
    "else_eval":
"$XC:ASSIGN:{L:AddressType};{S:address}}${XC:ASSIGN:{L:AddressGroup};{S:addrgrp}}"
  }
},
{
  "name": "Check if Asset sync is set for delete",
  "operation": "CONDITION",
  "condition": {
    "condition_type": "OR",
    "statements": [
      {
        "left": "${E:A:values{extattrs}{Fortinet_Asset_Sync}{value}}",
        "op": "!=",
        "right": "true"
      }
    ],
    "stop": true
  }
},
{
  "name": "Check if address group set for delete",
  "comment": "Check if address group the address needs to be deleted from is set",
  "operation": "CONDITION",
  "condition": {

```

```

    "condition_type" : "AND",
    "statements" : [
        {
            "left":
"$E:A:values{extattrs}{Fortinet_Asset_Group}{value}",
            "op": "==",
            "right": ""
        }
    ],
    "stop": true
}
},
{
    "name": "Assign address group to delete from",
        "comment": "Get the address group the address needs to be deleted from",
    "operation": "NOP",
    "body_list": [
        "$XC:COPY:{L:HostGroup}:{E:values{extattrs}{Fortinet_Asset_Group}{value}}"
    ]
},
{
    "name": "Debug before Delete",
    "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": "Host delete address from group",
    "comment": "Delete the IP address from the address group allow_group",

```



```

    "parse": "JSON",
    "operation": "POST",
    "no_connection_debug": false,
    "transport": {"path":
"api/v2/cmdb/firewall/${L:A:AddressGroup}/${L:A:HostGroup}/member/Infoblox_${L:A:HostObjectType}
}_${L:U:HostIP}?vdom=root"},
    "headers":{
        "Content-Type":"application/json",
        "Authorization": "Bearer ${S:A:Token}",
        "X-HTTP-Method-Override": "DELETE"
    }
},
{
    "name": "Host delete address",
    "comment": "Delete IP address from the list of address on the firewall",
    "parse": "JSON",
    "operation": "POST",
    "no_connection_debug": false,
    "transport": {"path":
"api/v2/cmdb/firewall/${L:A:AddressType}/Infoblox_${L:A:HostObjectType}_${L:U:HostIP}?vdom=root"}
,
    "headers":{
        "Content-Type":"application/json",
        "Authorization": "Bearer ${S:A:Token}",
        "X-HTTP-Method-Override": "DELETE"
    }
},
    {
        "name": "Host delete check",
        "comment": "Check deletion",

```

```

        "operation": "CONDITION",
        "condition": {
            "statements": [
                {
                    "left": "success",
                    "op": "==",
                    "right": "${P:A:status}"
                },
                "condition_type": "AND",
                "next": "FinExit"
            ]
        },
    },
{
    "name": "Check for lease",
    "comment": "Check for lease event",
    "operation": "CONDITION",
    "condition": {
        "condition_type": "AND",
        "statements": [
            {
                "left": "${E::event_type}",
                "op": "==",
                "right": "LEASE"
            }
        ],
        "next": "Check if Sync required for Lease"
    }
},

```

```
{
  "name": "Check if Sync required for Lease",
  "comment": "Check if event sync is required",
  "operation": "CONDITION",
  "condition": {
    "condition_type": "AND",
    "statements": [
      {
        "left": "${E:A:ip.extattrs{Fortinet_Asset_Sync}}",
        "op": "!=",
        "right": "true"
      }
    ],
    "stop": true
  }
},
```

```
{
  "name": "Check if IPv4 or IPv6 for assigning variables for lease",
  "operation": "CONDITION",
  "condition": {
    "statements": [
      {
        "left": "${E:A:values{ipv4addr}}",
        "op": "!=",
        "right": ""
      }
    ],
  },
```

```

        "condition_type": "AND",
        "eval":
"$XC:COPY:{L:HostIP}:{E:values{ipv4addr}}}$XC:COPY:{L:HostObjectType}:{E:object_type}}$XC:ASSIGN:
{L:HostIPtype}:{S:ipv4addr}}$XC:ASSIGN:{L:AddressType}:{S:address}}$XC:ASSIGN:{L:AddressGroup}:{S:
addrgrp}}$XC:ASSIGN:{L:AddressField}:{S:subnet}}$XC:ASSIGN:{L:Mask}:{S:/32}}",
        "else_eval":
"$XC:COPY:{L:HostIP}:{E:values{ipv6addr}}}$XC:COPY:{L:HostObjectType}:{E:object_type}}$XC:ASSIGN:
{L:HostIPtype}:{S:ipv6addr}}$XC:ASSIGN:{L:AddressType}:{S:address6}}$XC:ASSIGN:{L:AddressGroup}:{S:
addrgrp6}}$XC:ASSIGN:{L:AddressField}:{S:ip6}}$XC:ASSIGN:{L:Mask}:{S:/128}}"
    }
},
{
    "name": "Check Lease State",
    "comment": "Check binding state of the lease",
    "operation": "CONDITION",
    "condition": {
        "condition_type" : "AND",
        "statements" : [
            {"left": "${E::binding_state}",
            "op": "!=",
            "right": "ACTIVE"
            }
        ],
        "next": "Debug before Delete",
        "else_next": "Debug before Add"
    }
},
{
    "name": "Check if IPv4 or IPv6 for assigning variables for discovery",
    "operation": "CONDITION",

```

```

"condition": {
  "statements": [
    {
      "left": "${E:A:values{ip_address}}",
      "op": "=~",
      "right": ":"
    }
  ],
  "condition_type": "AND",
  "eval":
"${XC:COPY:{L:HostIP}:{E:values{ip_address}}}${XC:COPY:{L:HostObjectType}:{E:object_type}}${XC:ASSIGN:{L:AddressType}:{S:address6}}${XC:ASSIGN:{L:AddressGroup}:{S:addrgrp6}}${XC:ASSIGN:{L:AddressField}:{S:ip6}}${XC:ASSIGN:{L:Mask}:{S:/128}}${XC:ASSIGN:{L:HostGroup}:{S:allow_group}}",
  "else_eval":
"${XC:COPY:{L:HostIP}:{E:values{ip_address}}}${XC:COPY:{L:HostObjectType}:{E:object_type}}${XC:ASSIGN:{L:AddressType}:{S:address}}${XC:ASSIGN:{L:AddressGroup}:{S:addrgrp}}${XC:ASSIGN:{L:AddressField}:{S:subnet}}${XC:ASSIGN:{L:Mask}:{S:/32}}${XC:ASSIGN:{L:HostGroup}:{S:allow_group}}",
  "next": "Debug before Add",
  "else_next": "Debug before Add"
}
},
{
  "name": "Check if IPv4 or IPv6 for assigning variables for insert",
  "operation": "CONDITION",
  "condition": {
    "statements": [
      {
        "left": "${L:A:HostIP}",
        "op": "=~",
        "right": ":"
      }
    ]
  }
}

```

```

    ],
    "condition_type": "AND",
    "else_eval":
    "${XC:ASSIGN:{L:HostIPtype}:{S:ipv4addr}}${XC:ASSIGN:{L:AddressType}:{S:address}}${XC:ASSIGN:{L:AddressGroup}:{S:addrgrp}}${XC:ASSIGN:{L:AddressField}:{S:subnet}}${XC:ASSIGN:{L:AddressField}:{S:subnet}}",
    "eval":
    "${XC:ASSIGN:{L:HostIPtype}:{S:ipv6addr}}${XC:ASSIGN:{L:AddressType}:{S:address6}}${XC:ASSIGN:{L:AddressGroup}:{S:addrgrp6}}${XC:ASSIGN:{L:AddressField}:{S:ip6}}"
    }
    },
    {
    "name": "Check if Asset sync is set for insert",
    "operation": "CONDITION",
    "condition": {
    "condition_type": "OR",
    "statements": [
    {
    "left": "${E:A:values{extattrs}{Fortinet_Asset_Sync}{value}}",
    "op": "!=",
    "right": "true"
    }
    ],
    },
    "stop": true
    }
    },
    {
    "name": "Check if address group set for insert",
    "comment": "Check if address group the address needs to be added to is set",
    "operation": "CONDITION",

```

```

"condition": {
  "condition_type": "AND",
  "statements": [
    {
      "left":
"$E:A:values{extattrs}{Fortinet_Asset_Group}{value}",
      "op": "==",
      "right": ""
    }
  ],
  "stop": true
},

{
  "name": "Assign address group to add to",
  "comment": "Get the address group the address needs to added to",
  "operation": "NOP",
  "body_list": [
    "$XC:COPY:{L:HostGroup}:{E:values{extattrs}{Fortinet_Asset_Group}{value}}"
  ]
},

{
  "name": "Debug before Add",
  "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 range",
            "operation": "CONDITION",
            "condition": {
                "statements": [
                    {
                        "left": "${L:A:HostObjectType}",
                        "op": "=~",
                        "right": "Range"
                    }
                ],
                "condition_type": "AND",
                "eval":
                "${XC:ASSIGN:{L:FortigateType}:{S:iprange}}${XC:COPY:{L:StartIP}:{E:values{start_addr}}}${XC:COPY:{L:EndIP}:{E:values{end_addr}}}",
                "next": "Host add range",
                "else_next": "Host add address"
            }
        },
    {
        "name": "Host add address",
        "comment": "Add address to the list of address on the firewall",
        "parse": "JSON",
        "operation": "POST",
        "no_connection_debug": false,
        "transport": {"path": "api/v2/cmdb/firewall/${L:A:AddressType}?vdom=root"},
        "headers": {
            "Content-Type": "application/json",
            "Authorization": "Bearer ${S:A:Token}"
        }
    }

```



```

},
"body_list": [
  "{",
  "\"name\": \"Infoblox_${L:A:HostObjectType}_${L:A:HostIP}\",",
  "\"${L:A:AddressField}\": \"${L:A:HostIP}${L:A:Mask}\",",
  "\"comment\": \"Added from Infoblox at ${L:A:timestamp}\",",
  "}"
]
},
{
  "name": "Check if range to skip next step",
  "operation": "CONDITION",
"condition": {
  "statements": [
    {
      "left": "${L:A:HostObjectType}",
      "op": "=~",
      "right": "Range"
    }
  ],
  "condition_type": "AND",
  "next": "Host add range",
  "else_next": "Host add address to group"
}
},
{
"name": "Host add range",
"comment": "Add range to the list of address on the firewall",

```

```
"parse": "JSON",
"operation": "POST",
"no_connection_debug": false,
"transport": {"path": "api/v2/cmdb/firewall/${L:A:AddressType}?vdom=root"},
"headers":{
  "Content-Type":"application/json",
  "Authorization": "Bearer ${S:A:Token}"
},
"body_list": [
  "{",
    "\"name\": \"Infoblox_${L:A:HostObjectType}_${L:A:HostIP}\",",
    "\"type\": \"${L:A:FortigateType}\",",
    "\"start-ip\": \"${L:A:StartIP}\",",
    "\"end-ip\": \"${L:A:EndIP}\",",
    "\"comment\": \"Added from Infoblox at ${L:A:timestamp}\",",
  "}"
]
},
{
  "name": "Host add address to group",
  "comment": "Add IP address to the group specified",
  "parse": "JSON",
  "operation": "POST",
  "no_connection_debug": false,
  "transport": {"path":
"api/v2/cmdb/firewall/${L:A:AddressGroup}/${L:A:HostGroup}/member?vdom=root"},
  "headers":{
    "Content-Type":"application/json",
    "Authorization": "Bearer ${S:A:Token}"
```

```

},
"body_list": [
  "{",
  "\"name\": \"Infoblox_${L:A:HostObjectType}_${L:A:HostIP}\"",
  "}"
]
},
{
  "name": "Host add check",
  "comment": "Check insertion",
  "operation": "CONDITION",
  "condition": {
    "statements": [
      {"left": "success",
       "op": "==",
       "right": "${P:A:status}"}
    ],
    "condition_type": "AND",
    "else_next": "FinExit",
    "next": "Check if discovery to stop"
  }
},
{
  "name": "Check if discovery to stop",
  "operation": "CONDITION",
  "condition": {
    "condition_type": "OR",
    "statements": [
      {

```

```
"left": "${E:A:event_type}",
"op": "==",
"right": "DISCOVERY_DATA"
}
],
"next": "FinExit",
"else_next": "Check if fixed address or network or range"
}
},
```

```
{
  "name": "Check if fixed address or network or range",
  "comment": "Check event type",
  "operation": "CONDITION",
  "condition": {
    "condition_type": "OR",
    "statements": [
      {
        "left": "${L:A:HostObjectType}",
        "op": "=~",
        "right": "Fixed"
      },
      {
        "left": "${L:A:HostObjectType}",
        "op": "=~",
        "right": "Range"
      },
      {
        "left": "${L:A:HostObjectType}",
```

```

        "op": "=~",
        "right": "Network"
    }
],
    "next": "Get reference from E namespace"
}
},
{
    "name": "Get Host record ref",
    "operation": "GET",
    "transport": {
        "path":
"record:host?${L:A:HostIPtype}=${L:U:HostIP}&network_view=${L:U:network_view}&_return_fields=ext
attrs"
    },
    "wapi": "v2.7",
    "result": [{
        "codes": "200,201,202,203,204",
        "next": "Get object reference"
    }]
},
{
    "name": "Get object reference",
    "operation": "CONDITION",
    "condition": {
        "condition_type": "AND",
        "statements": [
            {
                "left": "${P:A:PARSE[0]_{_ref}}",

```

```

        "op": "!=",
        "right": ""
    }
],
"eval": "${XC:COPY:{L:Obj_ref}:{P:PARSE[0]{_ref}}}",
"next":"Debug before Extensible Attributes update"
}
},
{
"name": "Get reference from E namespace",
"comment":"Get object reference for network/range/fixed address from E
namespace",
"operation": "NOP",
"body_list": [
"${XC:COPY:{L:Obj_ref}:{E:values[_ref}}}"
]
},
{
"name": "Debug before Extensible Attributes update",
"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:}}${XC:DEBUG:{UT:}}"
},
{
"name": "Update extattrs timestamp",
"operation": "PUT",
"transport": {
"path": "${L:A:Obj_ref}"
},

```

```
"wapi": "v2.7",
"wapi_quoting": "JSON",
"body_list": [
  {"\\"extattrs+\":{\\"Fortinet_Asset_SyncedAt\": {\\"value\": \\"${L:A:timestamp}\\"}}}"
]
},
{
  "name": "FinExit",
  "comment": "Stop execution of the template.",
  "operation": "CONDITION",
  "condition": {
    "condition_type": "AND",
    "statements": [
      {"left": "1",
       "op": "==",
       "right": "1"}
    ],
    "stop": true
  }
}
]
```