```
{
        "name": "Fortinet Security",
        "comment": "Fortinet DNS Security Events",
        "version": "4.0",
        "type": "REST_EVENT",
        "event_type": [
                "RPZ",
                "TUNNEL",
                "ADP"
        ],
        "action_type": "Fortinet RPZ action to add address to deny_group",
        "content_type": "application/json",
        "vendor_identifier": "Fortinet",
        "instance_variables": [
    {
     "name": "Fortinet_Security_Group",
     "type": "STRING",
     "value": "deny_group"
    }
   ],
        "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:}}"
      },
```

```
        {
                "name":"Check if Sync EA is set on the IP",
                "operation": "CONDITION",
                "condition": {
"statements": [
  {
                                        "left": "${E:A:ip.extattrs{Fortinet_Security_Sync}}",
                                        "op": "!=",
                                        "right": "true"
                                }
],
"condition_type": "AND",
"next": "Check if Sync EA is set on the network"
                }
        },
        {
                "name":"Check if Security Group EA is set on the IP",
                "operation": "CONDITION",
                "condition": {
"statements": [
  {
                                        "left": "${E:A:ip.extattrs{Fortinet_Security_Group}}",
                                        "op": "==",
                                        "right": ""
                                }
],
"condition_type": "AND",
"eval": "${XC:COPY:{L:HostGroup}:{I:Fortinet_Security_Group}}",
```

```
                                "else_eval":
"${XC:COPY:{L:HostGroup}:{E:ip.extattrs{Fortinet_Security_Group}}",
                                "next":"Check if IPv4 or IPv6 for assigning variables for insert"
                        }
                },
                {
                        "name":"Check if Sync EA is set on the network",
                        "operation": "CONDITION",
                        "condition": {
        "statements": [
          {
                                                "left":
"${E:A:network.extattrs{Fortinet_Security_Sync}}",
                                                "op": "!=",
                                                "right": "true"
                                        }
        ],
        "condition_type": "AND",
                                "stop":true
                        }
                },
                {
                        "name":"Check if Security Group EA is set on the network",
                        "operation": "CONDITION",
                        "condition": {
        "statements": [
          {
                                                "left":
"${E:A:network.extattrs{Fortinet_Security_Group}}",
                                                "op": "==",
```

```
                                                "right": ""
                                        }
                ],
                "condition_type": "AND",
                "eval": "${XC:COPY:{L:HostGroup}:{I:Fortinet_Security_Group}}",
                                        "else_eval":
"${XC:COPY:{L:HostGroup}:{E:network.extattrs{Fortinet_Security_Group}}"
                        }
                },
                {
                        "name": "Assign L variables from E namespace",
                        "operation": "NOP",
                        "body_list": [
                                "${XC:COPY:{L:HostIP}:{E:source_ip}}",
                                "${XC:COPY:{L:timestamp}:{E:timestamp}}",
                                "${XC:COPY:{L:network_view}:{E:network.network_view}}"
                        ]
                },
                {
                        "name": "Check if IPv4 or IPv6 for assigning variables for insert",
                        "operation": "CONDITION",
        "condition": {
          "statements": [
            {
                                                "left": "${L:A:HostIP}",
                                                "op": "=~",
                                                "right": ":"
                                        }
                ],
```

```
            "condition_type": "AND",
            "eval":
"${XC:ASSIGN:{L:AddressType}:{S:address6}}${XC:ASSIGN:{L:AddressGroup}:{S:addrgrp6}}${XC:ASSIGN:{L
:AddressField}:{S:ip6}}${XC:ASSIGN:{L:Mask}:{S:128}}",
                                        "else_eval":
"${XC:ASSIGN:{L:AddressType}:{S:address}}${XC:ASSIGN:{L:AddressGroup}:{S:addrgrp}}${XC:ASSIGN:{L:A
ddressField}:{S:subnet}}${XC:ASSIGN:{L:Mask}:{S:32}}"
                                    }
                    },
                    {
        "name": "Check if RPZ",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "AND",
            "statements": [
                {
                    "left": "${E:A:event_type}",
                    "op": "==",
                    "right": "RPZ"
                }
            ],
            "eval": "${XC:COPY:{L:BlockedDomain}:{E:query_name}}",
            "next": "Debug before Add"
        }
    },
    {
        "name": "Check if Tunnel or ADP",
        "operation": "CONDITION",
        "condition": {
            "condition_type": "AND",
```

```
        "statements": [
          {
            "left": "${E:A:event_type}",
            "op": "==",
            "right": "ADP"
          }
        ],
        "eval": "${XC:COPY:{L:BlockedDomain}:{E:rule_name}}",
        "else_eval": "${XC:COPY:{L:BlockedDomain}:{E:domain_name}}"
      }
    },
                {
      "name": "Debug before Add",
      "operation": "NOP",
      "body":
"${XC:DEBUG:{H:}}${XC:DEBUG:{E:}}${XC:DEBUG:{I:}}${XC:DEBUG:{L:}}${XC:DEBUG:{S:}}${XC:DEBUG:{P:}}
${XC:DEBUG:{R:}}${XC:DEBUG:{RH:}}${XC:DEBUG:{UT:}}"
    },
                {
                        "name": "Host add address",
      "comment": "Add address to the list of address on the firewall",
                        "parse": "JSON",
                        "operation": "POST",
                        "no_connection_debug": false,
                        "transport": {"path": "api/v2/cmdb/firewall/${L:A:AddressType}?vdom=root"},
      "headers":{
        "Content-Type":"application/json",
        "Authorization": "Bearer ${S:A:Token}"
      },
```

```
        "body_list": [
            "{",
            "\"name\":\"Infoblox_${E:A:event_type}_${L:A:HostIP}\",",
                                                "\"${L:A:AddressField}\":\"${L:A:HostIP}/${L:A:Mask}\",",
                                                "\"comment\":\"Added from Infoblox at ${L:A:timestamp} for
querying ${L:A:BlockedDomain}\",",
            "}"
          ]
                                },


                        {
                                "name": "Host add address to group",
        "comment": "Add IP address to deny_group",
                                "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 ${S:A:Token}"
        },
        "body_list": [
          "{",
          "\"name\":\"Infoblox_${E:A:event_type}_${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 for IPv6"
        }
},

{
        "name": "check for IPv6",
        "operation": "CONDITION",
        "condition": {
                "statements": [
                        {
                                "left": "${L:A:HostIP}",
                                "op": "=~",
                                "right": ":"
                        }
                ],
        "condition_type": "AND",
        "next": "wapi_response_checkIPv6Fix_ref"
        }
```

```
        },
        {
            "operation": "CONDITION",
            "name": "wapi_response_checkIPv4Fix_ref",
            "condition": {
                "condition_type": "AND",
                "statements": [
                    {
                        "left": "${L:A:Obj_ref}",
                        "op": "!=",
                        "right": ""
                    }
                ],
            "next": "Debug before EA update"
            }
        },
        {
            "name": "Get HostIPv4 _ref",
            "operation": "GET",
            "transport": {
                "path":
"record:host?ipv4addr=${L:A:HostIP}&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",
                "else_stop": true
        }
},

{
        "operation": "CONDITION",
        "name": "wapi_response_checkIPv6Fix_ref",
        "condition": {
                "condition_type": "AND",
                "statements": [
                        {
                                "left": "${L:A:Obj_ref}",
                                "op": "!=",
                                "right": ""
                        }
                ],
                "next": "Debug before EA update"
        }
},
{
```

```
                "name": "Get HostIPv6 _ref",
                "operation": "GET",
                "transport": {
                        "path":
"record:host?ipv6addr=${L:A:HostIP}&network_view=${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": "Debug before EA 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_Security_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
                }
            }
        ]
}
```