

# Check Firewall Rules Provisioning telemetry

Last updated by | Charlene Wang | Jan 3, 2023 at 12:30 AM PST

## Contents

- [Customer side](#)
- [CSS side](#)
- [Classification](#)

Created On: Nov 17, 2022

Authored by: luyang1

Reviewed By: zhizhwan

How to check firewall rule provisioning log:

## Customer side

If the change operation was issued via Azure Portal, [Check Firewall changes in Portal Activity Log](#). (May not appear in activity log immediately, around 5mins latency)

## CSS side

XTS(CMS)/ASC/Kusto

XTS(CMS query) (telemetry latency is the shortest):

```
-- Check firewall rule change operations
select * from management_operations where logical_server_name = '{serverName}'
and subscription_id = '{subscriptionId}'
and operation_type like '%FirewallRule%'
order by last_update_time

-- Check VNet firewall rule setting, create time, last update time
select logical_server_name, rule_name, state, vnet_subscription_id, vnet_name, subnet_name, ignore_missing_vne
create_time, last_update_time, last_state_change_time, last_exception, details, error_message, error_code, is_
from vnet_firewall_rules
where logical_server_name = 'noraauessv01'

-- Check outbound firewall rules for a specific server
select logical_server_name, outbound_rule_fqdn, state, subscription_id, workflow_position, create_time, last_u
from logical_server_outbound_firewall_rules
where logical_server_name = 'noraauessv01'
```

ASC:

Server level: Provisioning -> Operations/Server CRUD tab

Kusto:

```
// check ARM layer log
let server = 'norauesv01';
let subID = '8659b2e4-b6b0-4772-86d2-e92896a894a4';
let startTime = datetime('2022-10-27 09:00:35');
let endTime = datetime('2022-10-27 09:50:35');
EventServiceEntries
| where PreciseTimeStamp >= startTime and PreciseTimeStamp <= endTime
| where subscriptionId =~ subID
| where resourceProvider =~ 'Microsoft.Sql'
| where eventCategory != 'Policy'
| where isempty(server) or resourceUri endswith strcat('/',server) or resourceUri has strcat('/',server, '/')
| extend statusMessage = parse_json(tostring(parse_json(properties).statusMessage))
| extend statusMessageStatus = statusMessage.status
| extend error = statusMessage.error
| extend errorMessage = statusMessage.error.message
| extend errorMessageCode = tostring(statusMessage.error.details[0].code)
| extend errorMessageDetails = tostring(statusMessage.error.details[0].message)
| where errorMessageDetails != 'The operation timed out and automatically rolled back. Please retry the operat
| parse resourceUri with '/subscriptions/' subId '/resourcegroups/' RGname 'providers/Microsoft.Sql/' Resourc
| parse resourceUri with '/subscriptions/' subId2 '/resourceGroups/' RGname2 'providers/Microsoft.Sql/' Resou
| parse operationName with 'Microsoft.Sql/' Operation
| extend Message = iif(errorMessageDetails!=', errorMessageDetails, errorMessage)
| extend Resource = iif(isnotempty(Resource), Resource, Resource2)
| project PreciseTimeStamp, correlationId, status, subStatus, Operation, Resource, Message, statusMessageStatu
| order by PreciseTimeStamp desc nulls last
```

```
// check ARM layer log, union three tables: HttpIncomingRequests,HttpOutgoingRequests,EventServiceEntries
let server = '{serverName}';
let db = '{dbName}';
let subID = '{SubscriptionId}';
let crlID = '{correlationId}'
let startTime = ago(1d);
let endTime = now();
union withsource=SourceTable kind=outer HttpIncomingRequests,HttpOutgoingRequests,EventServiceEntries
| where TIMESTAMP >= startTime and TIMESTAMP <= endTime
| where subscriptionId =~ subID
| where properties contains server //and properties contains db
| where correlationId =~ crlID
| where isempty(server) or resourceUri endswith strcat('/',server) or resourceUri has strcat('/',server, '/')
| extend statusMessage = parse_json(tostring(parse_json(properties).statusMessage))
| extend statusMessageStatus = statusMessage.status
| extend error = statusMessage.error
| extend errorMessage = statusMessage.error.message
| extend errorMessageCode = tostring(statusMessage.error.details[0].code)
| extend errorMessageDetails = tostring(statusMessage.error.details[0].message)
| where errorMessageDetails != 'The operation timed out and automatically rolled back. Please retry the operat
| parse resourceUri with '/subscriptions/' subId '/resourcegroups/' RGname 'providers/Microsoft.Sql/' Resourc
| parse resourceUri with '/subscriptions/' subId2 '/resourceGroups/' RGname2 'providers/Microsoft.Sql/' Resou
| extend Message = iif(errorMessageDetails!=', errorMessageDetails, errorMessage)
| extend Resource = iif(isnotempty(Resource), Resource, Resource2)
| project SourceTable, PreciseTimeStamp, resourceProvider, operationName, Resource
, status, subStatus, errorCode, errorMessage, Message, statusMessageStatus, errorMessageCode
, eventTimestamp, correlationId, operationId, resourceUri, targetResourceProvider, targetUri
, properties, authorization, claims, Deployment, principalOid, principalPuid
```

```
// get ARM layer correlationID from the above query, and use it in below queries
Traces
| where TIMESTAMP > {startTime}
| where TIMESTAMP < {endTime}
| where subscriptionId =~ "{subscriptionId}"
| where correlationId =~ "{correlationId}"
| project PreciseTimeStamp, TaskName, correlationId, operationName, message, exception, SourceNamespace, addit
, ActivityId, subscriptionId, tenantId
```

```
Errors
| where TIMESTAMP > {startTime}
```

```

| where TIMESTAMP < {endTime}
| where correlationId =~ "{correlationId}"
| project PreciseTimeStamp, correlationId, operationName, message, exception, additionalProperties, tenantId,

ProviderTraces
| where TIMESTAMP > {startTime}
| where TIMESTAMP < {endTime}
| where subscriptionId =~ "{subscriptionId}"
| where correlationId =~ "{correlationId}"
| project PreciseTimeStamp, correlationId, operationName, message, exception, ActivityId, principalOid, provid

ProviderErrors
| where TIMESTAMP > {startTime}
| where TIMESTAMP < {endTime}
| where subscriptionId =~ "{subscriptionId}"
| where correlationId =~ "{correlationId}"
| project PreciseTimeStamp, correlationId, providerNamespace, resourceType, operationName, message, exception,

// check resource provider layer telemetry
// -- run below Kusto query in corresponding region
// check firewall rule change operations for a specific server
MonManagementResourceProvider
| where http_verb == "PUT"
| where TIMESTAMP > {startTime}
| where TIMESTAMP < {endTime}
| where request_url contains '{serverName}'
| where operation_name contains "MICROSOFT.SQL"
| where operation_name contains "FIREWALLRULES" or operation_name contains "VIRTUALNETWORKRULES" or operation_
| project originalEventTimestamp, logical_server_name, request_id, controller_name, action_name, http_verb, me
exception_http_code, request_url, code_package_version, operation_type, api_version,
response_code, body_size, headers, azure_async_operation_header, client_request_id, exception_type, error_desc
extra_parameters, client_routing_id, caller_address
// Outbound Firewall rule -> operation_name contains "SECURITYALERTPOLICIES"

// use the request id got from the above query
MonManagement
| where request_id =~ "{request_id}"
| project originalEventTimestamp, request_id, event, state, old_state, new_state, action, request_name, operat
, message, fsm_error_message, error_message, is_user_error, last_exception, exception_message

// check VNet firewall rules settings
MonVnetFirewallRules
| where logical_server_name =~ '{serverName}'
| where TIMESTAMP > datetime(2022-10-27 10:00:00)
| where TIMESTAMP < datetime(2022-10-27 12:25:00)
| project PreciseTimeStamp, logical_server_name, rule_name, vnet_subscription_id, vnet_name, subnet_name, vnet
, operation_type, create_time, last_update_time, last_exception, error_code, error_message, details, ignore_mi
| order by PreciseTimeStamp desc

// check VNet firewall rules operations
MonManagementVnetFirewallRule
| where logical_server_name =~ '{serverName}'
| where TIMESTAMP > datetime(2022-10-26 01:50:00)
| where TIMESTAMP < datetime(2022-10-28 05:25:00)
| project originalEventTimestamp, logical_server_name, operation_name, operation_type, rule_name, event, subsc

```



## Classification

Root cause path - /CRUD/Firewall rule/Create failure

**How good have you found this content?**

