API Guide

Contents

REST Client Authentication Mechanism

REST server supports three modes of authentication, as follows:

- Authentication through an AD server
- Authentication through LDAP server
- Authentication using credentials configured on the Ordr SCE

The AD/LDAP authentication servers' configuration page is located at **System** > **Service Integration** > **Internal Services** > **External Authentication** in Ordr SCE user interface.

The REST server authentication mode configuration page is located at **System** > **Service Integration** > **Internal Services** > **SCE API** in Ordr SCE user interface.

Note:

The response to a request may not contain all the data, because the REST server supports paginated response. In paginated response, the response includes metadata that contains the next-link the REST client must submit in order to fetch successive data for the request it had originally initiated.

Ordr SCE REST APIs

HTTP Method	Uniform Resource Identifier (URI)	API Description	Output Format
GET	`/Rest/Devices`	Fetches information of all devices in the system.	JSON
GET	`/Rest/Devices/version`	Retrieves the current version of the Devices API supported by Ordr SCE.	Plain text
	`/Rest/Devices?limit=<value>`	Controls the count of devices in response. For example: `Devices?limit=10`	
GET	`/Rest/Devices?vulnIds`	Fetches the list of devices having specific vulnerabilities. **Note**: Values can be comma separated list. For example: `/Devices?vulnIds==FDA-173239,CVE-2020-0601`	JSON
	`/Rest/Devices?iot=true`	Fetches all IoT devices.	
	`/Rest/Devices?non_iot=true`	Fetches all Non-IoT devices.	
	`/Rest/Devices?openPorts=true`	Includes open ports in devices information response.	

	`/Rest/Devices?weakPassword=true`	Includes open ports with weak password in devices information response.	
	`/Rest/Devices?tenantGuid=<tenantGuid>`	Uses APIs for a particular tenant.	
	`/Rest//Devices?include=subcategory`	Includes optional information in response. **Note**: Values can be comma separated list.	
GET	`/Rest/Devices?mac=<mac-address>`	Fetches the device information for the given MAC address. The software's information (third-party apps, OS patches, and anti-virus products) is displayed in response if the device is Windows and WinRM enabled.	JSON
GET	`/Rest/Devices?ip=<ip-address>`	Fetches the device information for the given IP address. The software's information (third-party apps, OS patches, and anti-virus products) is displayed in response if the device is Windows and WinRM enabled.	JSON
GET	`/Rest/Devices?group=<group-name>`	Fetches all the devices belonging to the given group. For example, medical devices group.	JSON
GET	`/Rest/Devices?appName=<application-name>`	Fetches all devices talking to an application.	JSON
GET	`/Rest/Devices?riskState=<state>`	Fetches all the devices whose risk state matches the specified state. Possible values for risk state are Critical, High, Medium, Low, and Normal.	JSON
GET	`/Rest/Devices?connStatus=ONLINE_IN_LAST_24_HRS`	Fetches new devices that showed up in the last 24 hours.	JSON
GET	`/Rest/Devices?connStatus=ONLINE_IN_LAST_WEEK`	Fetches new devices that showed up within last week.	JSON
GET	`/Rest/Devices?connStatus=ONLINE&startTime=1569737345520&endTime=1569823745520`	Fetches new devices that showed up between start time and end time. **Note**: You should specify start time and end time in milliseconds (since Jan 1, 1970, UTC).	JSON
GET	`/Rest/Devices?connStatus=ONLINE`	Fetches devices that are currently online.	JSON
GET	`/Rest/Devices?connStatus=OFFLINE_IN_LAST_24_HRS`	Fetches devices that went offline in the last 24 hours.	JSON
GET	`/Rest/Devices?connStatus=OFFLINE_IN_LAST_WEEK`	Fetches devices that went offline within last week.	JSON
GET	`/Rest/Devices?connStatus=OFFLINE&startTime=`	Fetches devices that went offline between start time and end time.	JSON

	1569737345520&endTime=1569823745520	**Note**: You should specify start time and end time in milliseconds (since Jan 1, 1970, UTC).	
	/Rest/DeviceUtil?mac=<mac address>&startTime=<unix epoch timestamp>&endTime=<unix epoch timestamp>	Fetches utilization information for a given device between start time and end time.	
	/Rest/DeviceUtil?deviceType=<type>&startTime=<unix epoch timestamp>&endTime=<unix epoch timestamp>	Fetches utilization information for all devices of a given type (for example: patient monitor, infusion pump, or ultrasound/mri) between start time and end time.	
	/Rest/DeviceUtil?profileName=<ordr profile name>&startTime=<unix epoch timestamp>&endTime=<unix epoch timestamp>	Fetches utilization information for all devices of a given profile between start time and end time.	
GET	/Rest/Devices?connStatus=OFFLINE	Fetches devices that are currently offline.	JSON
GET	/Rest/Devices?sensorName=test	Fetches devices that are currently under the purview of the test sensor.	JSON
GET	/Rest/Devices?sensorIp=192.168.101.1	Fetches devices that are currently under the purview of the sensor with 192.168.101.1 IP.	JSON
GET	/Rest/Devices?diskEncrypted=<status>	Fetches all devices which has its disk encryption status matches given status. Status can be 'true' or 'false' (currently Windows-only devices using WinRM gathered data. Also return devices only if disk encryption status present in database).	JSON
GET	/Rest/Devices?biosPassword=true	Fetches all devices which has its BIOS password status matches given status. Status can be 'true' or 'false' (currently Windows-only devices using WinRM gathered data. Also return devices only if BIOS password status present in database).	JSON
GET	/Rest/Devices?softwareInstalled=<softwareName>	Fetches all devices which has the specified software installed. This will do a substring comparison of the passed argument. Example software name is 'Mozilla' (currently Windows-only devices using WinRM gathered data. Returns devices only if its software information present in database).	JSON
GET	/Rest/Devices?patchInstalled=<HotfixId>	Fetches all devices which has the specified hotfix installed.	JSON

		This will do full string comparison of the passed argument. Example hotfix ID is KB4534132 (currently Windows-only devices using WinRM gathered data. Returns devices only if patch info present in database).	
GET	`/Rest/Flows/version`	Retrieves the current version of the Flows API supported by Ordr SCE.	Plain text
GET	`/Rest/Flows?srcIp=<ip-address>`	Fetches all the flows whose source is the specified IP address.	JSON
GET	`/Rest/Flows?dstIp=<ip-address>`	Fetches all the flows whose destination is the specified IP address.	JSON
GET	`/Rest/Flows?srcMac=<mac-address>`	Fetches all the flows whose source is the specified device.	JSON
GET	`/Rest/Flows?dstMac=<mac-address>`	Fetches all the flows whose destination is the specified device.	JSON
	`/Rest/Flows?limit=<value>`	Controls the count of flows in response. For example: `/Flows?limit=10`	
GET	`/Rest/Applications/version`	Retrieves the current version of the Applications API supported by Ordr SCE.	Plain text
GET	`/Rest/Applications?mac=<mac-address>`	Fetches all the applications used by the device for the given device MAC.	JSON
GET	`/Rest/Applications?ip=<ip-address>`	Fetches all the applications used by the device for the given device IP.	JSON
GET	`/Rest/SecurityAlarms`	Fetches all the security incidents detected by the system.	JSON
GET	`/Rest/SecurityAlarms?mac=<mac-address>`	Fetches all the security incidents for the device for the given device MAC.	JSON
GET	`/Rest/SecurityAlarms?ip=<ip-address>`	Fetches all the security incidents for the device for the given device IP.	JSON
GET	`/Rest/SecurityAlarms?category=<category-type>`	Fetches all the security incidents for the given alarm category.	JSON
GET	`/Rest/SecurityAlarms/Summary`	Provides a summary of the security alarms by category.	JSON
	`/Rest/SecurityAlarms?limit=<value>`	Controls the count of alarms in response. For example: `/SecurityAlarms?limit=10`	
GET	`/Rest/NetworkDevices`	Fetches all network equipments.	JSON

REST APIs - Sample Queries and Output

Description : Retrieve the current version of the Devices API supported by Ordr SCE.

Request URI : `https://<Ordr_SCE>/Rest/Devices/version`

Output : `Current supported version of /Rest/Devices API is 1.0.`

Description : Fetch device information for the given MAC address.

Request URI : `https://192.168.104.182/Rest/Devices?mac=00:50:56:6A:42:46`

Output :

```
 {
     "MetaData":{
         "Count":1
     },
     "Devices":[
         {
             "MacAddress":"<mac-address>”,
             "IpAddress":"<IP-address>",
             "Group":"Medical Devices",
             "Profile":"GE-LOGIQ700-Ultrasound",
             "MfgName":"GEMedica",
             "LongMfgName":"G.E. Medical Systems",
             "Vlan":204,
             "ModelNameNo":"LOGIQ 700",
             "RiskState":"NORMAL",
             "DeviceType":"Ultrasound",
             "SerialNo":"<serial-number>",
             "DeviceDescr":"Ultrasound",
             "Subnet":"10.200.204.0/24",
             "SwVersion":"R6.1"
             "softwareInfo": {
                  "ThirdPartyApps": [{
                        "Name": "Mozilla Firefox 65.0.1 (x64 en-US)",
                        "Version": "65.0.1",
                        "Vendor": "Mozilla",
                        "InstallDate": null
                  }],
                  "OsPatches": [{
                        "HotfixId": "KB4534132",
                        "InstalledOn": "1581580800000",
                        "Description": "Update"
                  }],
                  "AvProducts": [{
                        "displayName": "Windows Defender",
                        "ProtectionState": "ACTIVE",
                        "IsUpToDate": "true",
                "UpdateTime": "true",
                "pathToSignedProductExe": "windowsdefender://"
            }]
        }
    }]
}
```

Description : Fetch the device information for the given IP address.

Request URI : `https://<Ordr_SCE>/Rest/Devices?ip=192.168.53.4`

Output : Sample output

Description : Fetch information of all devices in the system.

Request URI : `https://<Ordr_SCE>/Rest/Devices`

Output :

```
{
  "MetaData": {
    "Count": 100,
    "next": "/Rest/Devices?clientMacToken=-7079632916954617042"
  },
  "Devices": [
    {
      "MacAddress": "<mac>",
      "IpAddress": "<ip>",
      "Group": "Network Devices",
      "Profile": "Cisco-WS-C3560X-24T-Catalyst Switch",
      "MfgName": "Cisco",
      "LongMfgName": "Cisco Systems, Inc",
      "Vlan": 2,
      "ModelNameNo": "WS-C3560X-24T",
      "RiskState": "NORMAL",
      "DeviceType": "Catalyst Switch",
      "SerialNo": "<serial>",
      "DeviceDescr": "Catalyst Switch",
      "SwVersion": "12.2(55)SE10",
      "OsVersion": "C3560E Software",
      "OsType": "Cisco IOS",
      "endpointType": "NONIOT_ENDPOINT",
      "knownVulnRiskState": "NORMAL",
      "noOfPorts": 43,
      "ports": [
        {
          "name": "GigabitEthernet0/12",
          "hardware": "Gigabit Ethernet",
          "type": "TRUNK",
          "vlan": "1",
          "remoteNwEquipIp": "10.200.201.8",
          "remoteNwEquipMac": "<mac>",
          "remoteNwEquipPort": "FastEthernet1/0/24",
          "remoteNwEquipName": "cisco_mgmt.not",
          "remoteNwEquipManufacturer": "Cisco Systems, Inc",
          "remoteNwEquipModelNo": "WS-C3750-24PS-S",
          "remoteNwEquipSwVersion": "12.2(55)SE7"
        }
      ],
      "alarmCount": 0,
      "riskScore": 0,
      "firstSeen": "2020-02-19 05:54:40 GMT",
      "lastSeen": "2020-03-15 06:03:10 GMT",
      "classificationState": "Classified",
      "sensorName": "reports-dpvm-ss48",
      "sensorIp": "172.18.10.16",
      "connStatus": "ONLINE"
    },
    {
      "MacAddress": "<mac>",
      "Group": "Medical Devices",
      "Profile": "Philips-Patient Monitoring",
      "MfgName": "PhilipsP",
      "LongMfgName": "Philips Patient Monitoring",
      "Vlan": 777,
      "ModelNameNo": "",
      "RiskState": "NORMAL",
      "DeviceType": "Patient Monitoring",
      "DeviceDescr": "Patient Monitoring",
      "OsType": "Linux Embedded RTOS",
      "endpointType": "IOT_ENDPOINT",
      "knownVulnRiskState": "NORMAL",
      "accessType": "WIRED",
      "nwEquipInterface": "52",
      "nwEquipHostname": "Aruba-2930F-48G-4SFPP",
      "nwEquipScrapeIp": "10.200.201.39",
      "alarmCount": 0,
      "riskScore": 0,
      "firstSeen": "2020-02-28 23:02:55 GMT",
      "lastSeen": "2020-02-28 23:02:55 GMT",
      "classificationState": "Classified",
      "sensorName": "reports-dpvm-ss48",
      "sensorIp": "172.18.10.16",
      "connStatus": "OFFLINE"
    },
    <<<<<SNIP>>>>>,
    {
      "MacAddress": "<mac>",
      "Group": "Mobile Phones and Tablets",
      "Profile": "Samsung-Galaxy Note9-Phone",
      "MfgName": "Samsung",
      "LongMfgName": "Samsung",
      "Vlan": 2,
      "ModelNameNo": "Galaxy Note9",
      "RiskState": "NORMAL",
      "DeviceType": "Phone",
      "DeviceDescr": "Phone",
      "OsType": "Android",
      "fqdn": "Galaxy-Note9.hq.ordr.net",
      "dhcpHostname": "Galaxy-Note9",
      "endpointType": "NONIOT_ENDPOINT",
      "knownVulnRiskState": "NORMAL",
      "alarmCount": 0,
      "riskScore": 0,
      "firstSeen": "2020-02-20 17:38:16 GMT",
      "lastSeen": "2020-03-06 01:59:33 GMT",
      "classificationState": "Classified",
      "sensorName": "dc13-dpvm-ss72",
      "sensorIp": "192.168.104.86",
      "connStatus": "OFFLINE"
    }
  ]
}
```

Description : Fetch the list of devices having specific vulnerabilities.
Request URI : `https://<SCE-IP>/Rest/Devices?vulnIds=FDA-173239,CVE-2020-0601`
Output : Sample output

Description : Fetch information of all devices talking to an application.
Request URI : `https://<Ordr_SCE>/Rest/Devices?appName=udp.bacnet`
Output : Sample output

Description : Fetch devices that are currently online.
Request URI : `https://<sce ip>/Rest/Devices?connStatus=ONLINE`
Output : Sample output

Description : Fetch utilization information for a given device between start time and end time.
Request URI : `https://<sce ip>/Rest/DeviceUtil?mac=<mac address>&startTime=<unix epoch timestamp>&endTime=<unix epoch timestamp>`
Output :

```
{
"MetaData":{"Count":1},
"DeviceUtilRecords":
[
{
"MacAddress":"48:0F:CF:48:40:BD",
"MfgName":"GE MEDICAL SYSTEMS",
"DeviceType":"MRI",
"ModelNameNo":"Signa Pioneer",
"UtilPercent":16
}
]
}
```

Description : Fetch utilization information for all devices of a given type (for example: patient monitor, infusion pump, or ultrasound/mri) between start time and end time.

Request URI : `https://<sce ip>/Rest/DeviceUtil?deviceType=<type>&startTime=<unix epoch timestamp>&endTime=<unix epoch timestamp>`

Output :

```
{
      "MetaData":{"Count":4},
      "DeviceUtilRecords":
      [
            {
                  "MacAddress":"00:80:17:3D:D6:40",
                  "MfgName":"Hitachi Medical Corporation",
                  "DeviceType":"MRI",
                  "ModelNameNo":"Oasis",
                  "UtilPercent":40
            },
            {
                  "MacAddress":"00:1B:21:02:1C:46",
                  "MfgName":"GE MEDICAL SYSTEMS",
                  "DeviceType":"MRI",
                  "ModelNameNo":"Signa HDxt",
                  "UtilPercent":44
            },
            {
                  "MacAddress":"C8:D3:FF:BA:7D:70",
                  "MfgName":"Philips Medical Systems",
                  "DeviceType":"MRI",
                  "ModelNameNo":"Achieva",
                  "UtilPercent":36
            },
            {
                  "MacAddress":"00:0E:0C:F5:FB:1C",
                  "MfgName":"GE MEDICAL SYSTEMS",
                  "DeviceType":"MRI",
                  "ModelNameNo":"Signa HDxt",
                  "UtilPercent":18
            }
      ]
}
```

Description : Fetch utilization information for all devices of a given profile between start time and end time.

Request URI : `https://<sce ip>/Rest/DeviceUtil?profileName=<ordr profile name>&startTime=<unix epoch timestamp>&endTime=<unix epoch timestamp>`

Output :

```
{
      "MetaData":{"Count":2},
      "DeviceUtilRecords":
      [
            {
                  "MacAddress":"48:0F:CF:48:40:BD",
                  "MfgName":"GE MEDICAL SYSTEMS",
                  "DeviceType":"MRI",
                  "ModelNameNo":"Signa Pioneer",
                  "UtilPercent":16
            },
            {
                  "MacAddress":"30:9C:23:41:9E:FE",
                  "MfgName":"GE MEDICAL SYSTEMS",
                  "DeviceType":"MRI",
                  "ModelNameNo":"Signa Pioneer",
                  "UtilPercent":0
            }
      ]
}
```

Description : Fetch devices that are currently offline.

Request URI : `https://<sce ip>/Rest/Devices?connStatus=OFFLINE`

Output : Sample output

Description : Fetch new devices that showed up between start time and end time.

Request URI : `https://<sce ip>/Rest/Devices?connStatus=ONLINE&startTime=1569737345520&endTime=1569823745520`

Output : Sample output

Description : Fetch new devices that showed up in the last 24 hours.
Request URI : `https://<sce ip>/Rest/Devices?connStatus=ONLINE_IN_LAST_24_HRS`
Output : Sample output

Description : Fetch new devices that showed up within last week.
Request URI : `https://<sce ip>/Rest/Devices?connStatus=ONLINE_IN_LAST_WEEK`
Output : Sample output

Description : Fetch devices that went offline in the last 24 hours.
Request URI : `https://<sce ip>/Rest/Devices?connStatus=OFFLINE_IN_LAST_24_HRS`
Output : Sample output

Description : Fetch devices that went offline within last week.
Request URI : `https://<sce ip>/Rest/Devices?connStatus=OFFLINE_IN_LAST_WEEK`
Output : Sample output

Description : Fetch all the devices whose risk state matches the specified state. Possible values for risk state are Critical, High, Medium, Low, and Normal.
Request URI : `https://<Ordr_SCE>/Rest/Devices?riskState=MEDIUM`
Output : Sample output

Description : Fetch devices that are currently under the purview of the test sensor.
Request URI : `https://<sce ip>/Rest/Devices?sensorName=test`
Output : Sample output

Description : Fetch devices that are currently under the purview of the sensor with 192.168.101.1 IP.
Request URI : `https://<sce ip>/Rest/Devices?sensorIp=192.168.101.1`
Output : Sample output

Description : Fetch all devices which has its disk encryption status matches given status (true or false).

Request URI : `https://<Ordr_SCE>/Rest/Devices?diskEncrypted=<status>`

Output : Sample output

Description : Fetch all devices which has its BIOS password status matches given status (true or false).

Request URI : `https://<Ordr_SCE>/Rest/Devices?biosPassword=true`

Output : Sample output

Description : Fetch all devices which has the specified software installed.

Request URI : `https://<Ordr_SCE>/Rest/Devices?softwareInstalled=<softwareName>`

Output : Sample output

Description : Fetch all devices which has the specified hotfix installed.

Request URI : `https://<Ordr_SCE>/Rest/Devices?patchInstalled=<HotfixId>`

Output : Sample output

Description : Fetch all the devices belonging to the given group. For example, medical devices group.

Request URI : `https://<Ordr_SCE>/Rest/Devices?group=Industrial Devices`

Output : Sample output

Description : Retrieve the current version of the Flows API supported by Ordr SCE.

Request URI : `https://<Ordr_SCE>/Rest/Flows/version?srcIP`

Output : `Current supported version of /Rest/Flows API is 1.0`

Description : Fetch all the flows whose source is the specified IP Address.

Request URI : `https://<Ordr_SCE>/Rest/Flows?srcIp=10.22.22.176`

Output :

```
{
    "MetaData":{
        "Count":2
    },
    "Flows":[
        {
            "vectorGuid":"a-13913-1700256025",
            "behaviorState":"NORMAL",
            "srcIp":"10.200.204.9",
            "dstIp":"192.168.101.145",
            "srcPort":49866,
            "dstPort":104,
            "ipProto":6,
            "rxBytes":0,
            "txBytes":60,
            "rxPkts":0,
            "txPkts":1,
            "lastSeenTimestamp":1537417950408,
            "external": false,
            "appName": "https",
            "remoteProfile": "Local-IP-Profile",
            "alarms": [{
               "alarmHash": "<alarmHash>",
               "category": "<category>",
               "categoryType": "<categoryType>"
            }]
        },
        {
            "vectorGuid":"a-13917-1700256025",
            "behaviorState":"NORMAL",
            "srcIp":"10.200.204.9",
            "dstIp":"192.168.101.241",
            "srcPort":60496,
            "dstPort":53,
            "ipProto":17,
            "rxBytes":73,
            "txBytes":57,
            "rxPkts":1,
            "txPkts":1,
            "lastSeenTimestamp":1537418130407
        }
    ]
}
```

Description : Fetch all the flows whose destination is the specified IP Address.
Request URI : `https://<Ordr_SCE>/Rest/Flows?dstIp=10.200.204.1`
Output : Sample output

Description : Fetch all the flows whose source is the specified device.
Request URI : `https://192.168.104.182/Rest/Flows?srcMac=00:50:56:07:DB:E4`
Output : Sample output

Description : Fetch all the flows whose destination is the specified device.
Request URI : `https://<Ordr_SCE>/Rest/Flows?dstMac=52:54:00:01:79:B6`
Output : Sample output

Description : Retrieve the current version of the Applications API supported by Ordr SCE.
Request URI : `https://<Ordr_SCE>/Rest/Applications/version`
Output : `Current supported version of /Rest/Applications API is 1.0`

Description : Fetch all the applications used by the device for the given device MAC.
Request URI : `https://<Ordr_SCE>/Rest/Applications?mac=28:63:36:A6:F9:01`
Output :

```
{
  "MetaData": {
    "Count": 4
  },
  "Applications": [
    {
      "protocol": "tcp",
      "appName": "ssh",
      "peers": [
        "00:0C:29:3F:A3:20"
      ]
    },
    {
      "protocol": "udp",
      "appName": "mdns",
      "peers": [
        "00:0C:29:02:28:1D",
        "00:0C:29:1C:7B:68",
        "00:0C:29:D7:6D:AB"
      ]
    },
    {
      "protocol": "udp",
      "appName": "SNMP",
      "peers": [
        "00:0C:29:02:28:1D",
        "00:0C:29:1C:7B:68",
        "00:0C:29:D7:6D:AB"
      ]
    },
    {
      "protocol": "udp",
      "appName": "ssdp",
      "peers": [
        "9C:93:4E:3C:D7:75",
        "AC:CC:8E:2B:A5:E4"
      ]
    }
  ]
}
```

Description : Fetch all the applications used by the device for the given device IP.

Request URI : `https://<Ordr_SCE>/Rest/Applications?ip=10.200.205.16`

Output : Sample output

Description : Fetch all the security incidents detected by the system.

Request URI : `https://<Ordr_SCE>/Rest/SecurityAlarms`

Output :

```
{
  "MetaData": {
    "Count": 100,
    "next": "/Rest/SecurityAlarms?clientMacToken=-
2454454372417895144&alarmHashToken=f743d6c2626385ef950754df93824e7df5e10d3e"
  },
  "SecurityAlarms": [
    {
      "alarmHash": "1128e80da32f658cb889938e3e2b5618dd85e568",
      "category": "KNOWN_VULN",
      "categoryType": "KNOWN_VULN",
      "severityLevel": "NORMAL",
      "riskScore": 0,
      "metaData": "",
      "peerId": "NA",
      "recentTimestamp": 1579691452742,
      "incidentType": "FDA-156164:1.5T Signa HDx, 3.0T Signa HDx, 1.5T Signa
HDxt, 3.0T Signa HDxt, Sig",
      "deviceMac": "00:50:56:D7:97:DB",
      "sensorName": "reports-dpvm-ss48",
      "sensorIp": "172.18.10.16"
    },
    {
      "alarmHash": "a5992a08bb38bc91c8df29107449c24143e93fe6",
      "category": "BAD_URL",
      "categoryType": "URL Malware",
      "severityLevel": "MEDIUM",
      "riskScore": 6,
      "metaData": "http://www.disneylanddaze.com/",
      "peerId": "http://www.disneylanddaze.com/",
      "recentTimestamp": 1579960155038,
      "incidentType": "Malware Site Access",
      "deviceMac": "00:0C:29:89:70:7E",
      "rawVectorGuid": "a-209104--132490175",
      "sensorName": "reports-dpvm-ss48",
      "sensorIp": "172.18.10.16",
      "locationInfo": {
        "country": "United States",
        "countryCode": "US",
        "city": "Bluffdale"
      }
    },
<<<<<SNIP>>>>>
    {
      "alarmHash": "f86269de68350b1f74cbc8b5df61ea27101fcf35",
      "category": "BAD_IP",
      "categoryType": "Suspicious Traffic",
      "severityLevel": "MEDIUM",
      "riskScore": 6,
      "metaData": "23.129.64.159",
      "peerId": "23.129.64.159",
      "recentTimestamp": 1583330351185,
      "incidentType": "packets to blacklisted destination",
      "deviceMac": "52:54:00:89:82:C5",
      "sensorName": "reports-dpvm-ss48",
      "sensorIp": "172.18.10.16"
    }
  ]
}
```

Description : Fetch all the security incidents for the device for the given device MAC.

Request URI : `https://192.168.104.182/Rest/SecurityAlarms?mac=<mac>`

Output :

```
{
  "MetaData": {
    "Count": 1
  },
  "SecurityAlarms": [
    {
      "alarmHash": "a5992a08bb38bc91c8df29107449c24143e93fe6",
      "category": "BAD_URL",
      "categoryType": "URL Malware",
      "severityLevel": "MEDIUM",
      "riskScore": 6,
      "metaData": "http://www.disneylanddaze.com/",
      "peerId": "http://www.disneylanddaze.com/",
      "recentTimestamp": 1579960155038,
      "rawVectorGuid": "a-209104--132490175",
      "locationInfo": {
        "country": "United States",
        "countryCode": "US",
        "city": "Bluffdale"
      }
    }
  ]
}
```

Description : Fetch all the security incidents for the device for the given device IP.

Request URI : `https://192.168.104.182/Rest/SecurityAlarms?mac=<mac>`

Output : Sample output

Description : Fetch all the security incidents for the given alarm category.

Request URI : `https://<Ordr_SCE>/Rest/SecurityAlarms?category=BAD_URL`

Output :

```
{
 "MetaData": {
    "Count": 1
  },
  "SecurityAlarms": [
    {
      "alarmHash": "a5992a08bb38bc91c8df29107449c24143e93fe6",
      "category": "BAD_URL",
      "categoryType": "URL Malware",
      "severityLevel": "MEDIUM",
      "riskScore": 6,
      "metaData": "http://www.disneylanddaze.com/",
      "peerId": "http://www.disneylanddaze.com/",
      "recentTimestamp": 1579960155038,
      "incidentType": "Malware Site Access",
      "deviceMac": "00:0C:29:89:70:7E",
      "rawVectorGuid": "a-209104--132490175",
      "sensorName": "reports-dpvm-ss48",
      "sensorIp": "172.18.10.16",
      "locationInfo": {
        "country": "United States",
        "countryCode": "US",
        "city": "Bluffdale"
      }
    }
  ]
}
```

Description : Provide a summary of the security alarms by category.

Request URI : `https://<Ordr_SCE>/Rest/SecurityAlarms/Summary`

Output :

```
{
"summary": [
{
"category": "BAD_URL",
"summary": [
{
"categoryType": "URL Malware",
"severityLevel": "MEDIUM",
"riskScore": 6,
"deviceCount": 2
}
]
},
{
"category": "KNOWN_VULN",
"summary": [
{
"categoryType": "KNOWN_VULN",
"severityLevel": "NORMAL",
"riskScore": 0,
"deviceCount": 11
}
]
},
{
"category": "URL_GENERIC",
"summary": [
{
"categoryType": "URL Generic alarm",
"severityLevel": "MEDIUM",
"riskScore": 6,
"deviceCount": 1
}
]
},
{
"category": "DEVICE_SIGNATURE_VIOLATION",
"summary": [
{
"categoryType": "Baseline Flow Violation",
"severityLevel": "MEDIUM",
"riskScore": 6,
"deviceCount": 7
}
]
},
{
"category": "PHISHING",
"summary": [
{
"categoryType": "URL Phishing",
"severityLevel": "MEDIUM",
"riskScore": 6,
"deviceCount": 3
}
]
}
]
}
```

Description : Fetch all network equipments.

Request URI : `https://<Ordr_SCE>/Rest/NetworkDevices`

Output :

```
{
    "networkDeviceInfo": {
        "totalAccessSwitches": 22,
        "totalAccessPorts": 564,
        "totalDot1xDisabledPorts": 564,
        "totalMabDisabledPorts": 564,
        "totalDot1xAndMabDisabledPorts": 564,
        "networkDevices": [
            {
                "name": "HP-3800-48G-PoEP-2SFPP",
                "scrapeIp": "10.100.16.10",
                "accessPorts": 23,
                "dot1xDisabledPorts": 23,
                "mabDisabledPorts": 23,
                "dot1xAndMabDisabledPorts": 23
            },

            {
                "name": "uplink_to_controller",
                "scrapeIp": "10.100.13.3",
                "accessPorts": 23,
                "dot1xDisabledPorts": 23,
                "mabDisabledPorts": 23,
                "dot1xAndMabDisabledPorts": 23
            }
        ]
    }
}
```

ōrdr

take control.

info@ordr.net
www.ordr.net

2445 Augustine Drive Suite 601
Santa Clara, CA 95054