DCAP CENTRAL

jSONAR Simplifying Security

tech note 1802

IMPLEMENTATION TECH NOTE
Integrating Netskope with jSonar's DCAP Central

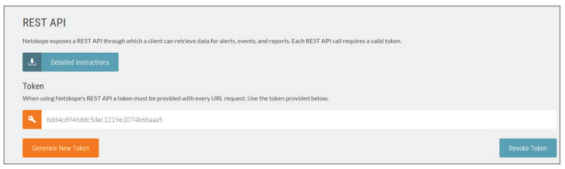
jSonar's DCAP Central security data lake can consume data from RESTful API to analyze the syslog data. This Technote will outline the steps necessary to integrate the two systems.

SYSTEM INTEGRATION

Integration between the systems is done using SonarGateway. Netskope is configured to generate syslog data using a JSON format that is then sent to DCAP Central for ingestion into the database. The documents are put in the netskope collection within the sonargd database as defined.

Steps to integrate the systems:

1. In the Netskope webui, The token can be generated by navigating to Settings > Tools > Rest API..



2. In your Sonar command line interface, Turn on the netskope configuration.
   a. vi /etc/rsyslog.d/sonargateway.conf and uncomment the line $IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/netskope.conf

```
$MaxMessageSize 32768
$IncludeConfig /etc/rsyslog.d/sonar/gateway/include/*.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/cassandra.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/cloudera.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/cloudwatch_aurora.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/cosmosdb.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/eventhub.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/helloworld.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/horton.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/kafka_consumer.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/mapr.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/mssql.conf
$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/okta.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/oracle.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/qradar.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/redjack.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/service_now.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/sonar_audit_log.conf
#$IncludeConfig /etc/rsyslog.d/sonar/gateway/rulesets/windows.conf
~
```

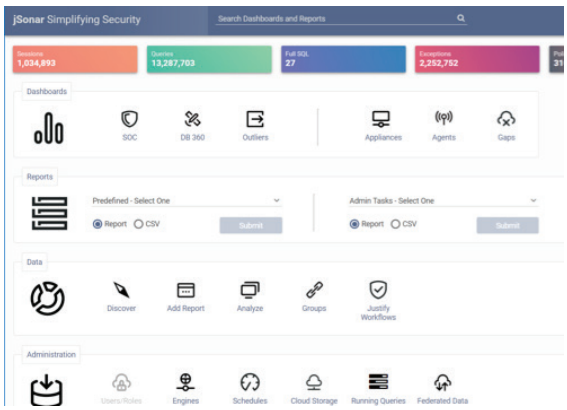
3. Get port number from rulesets configuration file.
   a. vi /etc/rsyslog.d/sonar/gateway/rulesets/netskope.conf

```
input(type="imptcp" port="10564" keepalive="on" ruleset="okta")

template(name="okta_rawmsg" type="list") {
    property(name="rawmsg")
    constant(value="D3L1M1T3R")
}

ruleset(name="okta") {
    action(type="omprog"
           binary="/usr/lib/sonar/gateway/sonargateway --config /etc/sonar/gateway/okta.json --input_delimiter D3L1M1
T3R --prenorm_delimiter PR3NORM --delay_depot_dir /var/lib/sonar/gateway"
           template="okta_rawmsg")
    stop
}
```

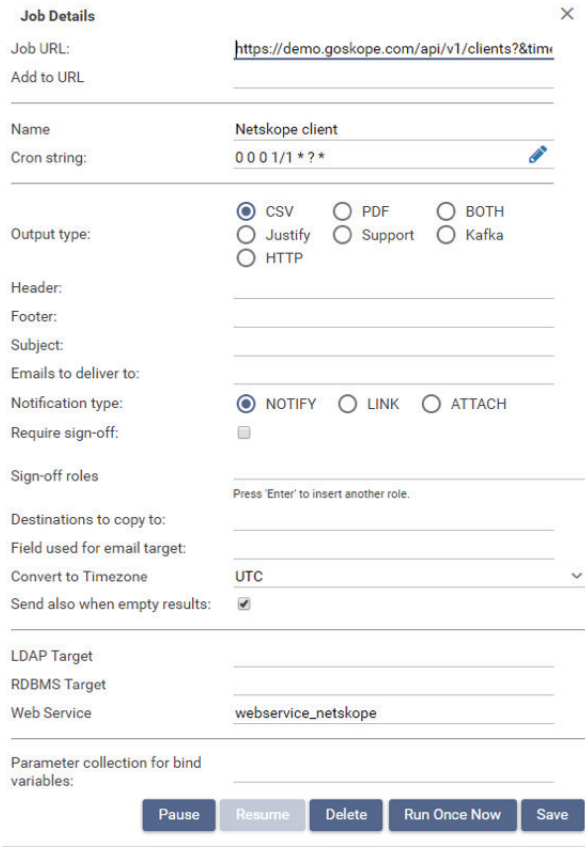
4. Configure the dispatcher web service section for Okta.
   a. vi /opt/sonarfinder/sonarFinder/dispatcher.conf
   b. add lines for Netskope configuration
      i. [webservice_netskope]
      ii. download_type = JSON
      iii. authurl = &token=<insert generated token>
      iv. SonarGateway_Address =<IP of Sonar machine>
      v. SonarGateway_Port =<port from netskope.conf>
      vi. limit = <limit of records to pull each run default=10>
      vii. initial_skip = <start skip for new jobs default=0>
      viii. paging_end_string = {"status":"success","msg":"","data":[]}
         1. Vendor specific default is for Netskope only
      ix. result_array_field_name=data
         1. Vendor specific default is for Netskope only
      x. result_error_string={"status":"error","errorCode":
         1. Vendor specific default is for Netskope only

5. Restart sonardispatcher and rsyslog
   a. systemctl restart sonardispatcher rsyslog

6. In the Central UI, schedule the Okta pull
   a. Click schedules



b. Click anywhere that is not an existing job to open the scheduler
c. Configure for netskope pull
   i. Enter the job URL:
      1. https://demo.goskope.com/api/v1/log_type?&timeperiod=time_period
   ii. Netskope has many different logs and parameters this is configured through the above URL. Please seek their documentation for guidance.
   iii. Enter the Name: netskope
   iv. Enter desired cron string
   v. Enter Web Service: webservice_netskope
   vi. Click save



7. To test that the pull is working
   a. In putty window into sonar environment
      i. tail -f /opt/sonarfinder/sonarFinder/dispatcher.log
   b. In the DCAP Central UI, open the netskope job in schedules that was just created
      i. Click Run Once Now at the bottom of the job

c. Check the tail of the dispatcher.log. The process should run with no errors, example below

```
2018-02-05 18:37:17Z [PID-6407] [INFO] [5a78a4496b48d9103613d264] Beginning to process job with name okta.
2018-02-05 18:37:17Z [PID-6407] [INFO] [5a78a4496b48d9103613d264] Start to Download from Web Service
2018-02-05 18:37:17Z [PID-6407] [INFO] [5a78a4496b48d9103613d264] WebService - Downloading JSON...
2018-02-05 18:37:17Z [PID-6407] [WARNING] authURL is not found
2018-02-05 18:37:17Z [PID-6407] [INFO] WebService - Reserved pattern detected in URL, replace with values
2018-02-05 18:37:17Z [PID-6407] [WARNING] Last call date not found, use the date from 180 days ago
2018-02-05 18:37:17Z [PID-6407] [INFO] Replace $LAST_CALL_DATE in URL
2018-02-05 18:37:17Z [PID-6407] [INFO] Web Service - Use apikey to download file
2018-02-05 18:37:17Z [PID-6407] [WARNING] API paging pattern not found in web service section, use 'X-Rate-Limit'
2018-02-05 18:37:17Z [PID-6407] [WARNING] 'warning_percentage' not set or not a integer, use 100 percent
2018-02-05 18:37:17Z [PID-6407] [INFO] Using api_paging_pattern: X-Rate-Limit
2018-02-05 18:37:17Z [PID-6407] [INFO]  ## Start processing API Web Service ##
2018-02-05 18:37:17Z [PID-6407] [INFO] Web Service URL: https://dev-269228.oktapreview.com/api/v1/logs?since=2017-08-09T19:37:17Z&jobName=okta
2018-02-05 18:37:17Z [PID-6407] [INFO] Start to get API Service results
2018-02-05 18:37:17Z [PID-6407] [INFO] Downloading data - 0 requests sent, still getting data..
2018-02-05 18:37:31Z [PID-6407] [INFO] Downloading data - 20 requests sent, still getting data..
2018-02-05 18:37:44Z [PID-6407] [INFO] Downloading data - 40 requests sent, still getting data..
2018-02-05 18:37:57Z [PID-6407] [INFO] Reach the warning limit:1
2018-02-05 18:37:57Z [PID-6407] [INFO] Reached API web wervice rate limit/threshold - Waiting for next reset :20 s
2018-02-05 18:38:23Z [PID-6407] [INFO] Downloading data - 60 requests sent, still getting data..
2018-02-05 18:38:37Z [PID-6407] [INFO] Downloading data - 80 requests sent, still getting data..
2018-02-05 18:38:50Z [PID-6407] [INFO] Downloading data - 100 requests sent, still getting data..
2018-02-05 18:39:01Z [PID-6407] [INFO] Reach the warning limit:1
2018-02-05 18:39:01Z [PID-6407] [INFO] Reached API web wervice rate limit/threshold - Waiting for next reset :21 s
2018-02-05 18:39:29Z [PID-6407] [INFO] Downloading data - 120 requests sent, still getting data..
2018-02-05 18:39:41Z [PID-6407] [INFO] Downloading data - 140 requests sent, still getting data..
2018-02-05 18:39:45Z [PID-6407] [INFO] No next page, Pagination is finished
2018-02-05 18:39:46Z [PID-6407] [INFO] [5a78a4496b48d9103613d264] Sending to SonarGateway...
2018-02-05 18:39:50Z [PID-6407] [INFO] [5a78a4496b48d9103613d264] Job completed
```

8. Now you should see netskope data in DCAP Central in the netskope collection in the sonargd database. Example of user login document:


```
{
  "_id": {
    "$oid": "5a78a4f8fad58d191d2429ca"
  },
  "target": "null",
  "legacyEventType": "core.user_auth.login_success",
  "outcome": {
    "result": "SUCCESS",
    "reason": "null"
  },
  "debugContext": {
    "debugData": {
      "requestUri": "/user/welcome/login/internal"
    }
  },
  "version": 0,
  "securityContext": {
    "isProxy": "null",
    "isp": "null",
    "asOrg": "null",
    "domain": "null",
    "asNumber": "null"
  },
  "transaction": {
    "detail": {},
    "id": "WiljWyh6JhSN394s1MMoSgAADO0",
    "type": "WEB"
  },
  "displayMessage": "User login to Okta",
  "eventType": "user.session.start",
  "request": {
    "ipChain": [
      {
        "source": "null",
        "version": "V4",
        "ip": "xx.xxx.xxx.xxx",
        "geographicalContext": {
          "postalCode": 20149,
```

```
      "geolocation": {
       "lon": -77.4728,
       "lat": 39.0481
      },
      "state": "Virginia",
      "city": "Ashburn",
      "country": "United States"
     }
    },
    {
     "source": "null",
     "version": "V4",
     "ip": "xx.xxx.xxx.xxx",
     "geographicalContext": {
      "postalCode": 6880,
      "city": "Westport",
      "geolocation": {
       "lon": -73.3428,
       "lat": 41.1449
      },
      "state": "Connecticut",
      "country": "United States"
     }
    }
   ]
  },
  "client": {
   "id": "null",
   "ipAddress": "xx.xxx.xxx.xxx",
   "geographicalContext": {
    "postalCode": 6880,
    "geolocation": {
     "lon": -73.3428,
     "lat": 41.1449
    },
    "city": "Westport",
    "state": "Connecticut",
    "country": "United States"
   },
   "device": "Computer",
   "zone": "null",
   "userAgent": {
    "os": "Windows 10",
    "browser": "FIREFOX",
    "rawUserAgent": "Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0"
   }
  },
  "severity": "INFO",
  "uuid": "86916c1a-e724-4713-8bf4-db547bff9e3c",
  "published": {
   "$date": "2017-12-07T15:50:51.679Z"
  },
  "authenticationContext": {
   "authenticationProvider": "null",
   "interface": "null",
   "credentialProvider": "null",
   "issuer": "null",
   "externalSessionId": "102m9PWoS4nRxmuAhKjWXa_
hg",
   "credentialType": "null",
   "authenticationStep": 0
  },
  "actor": {
   "detailEntry": "null",
   "type": "User",
   "displayName": "Jason Belanger",
   "alternateId": "xxxx@gmail.com",
   "id": "00ud6hns8kO999zuo0h7"
  }
 },
```


tech note1802