How to Integrate a CloudGen Firewall with SCADAfence

https://campus.barracuda.com/doc/94541033/

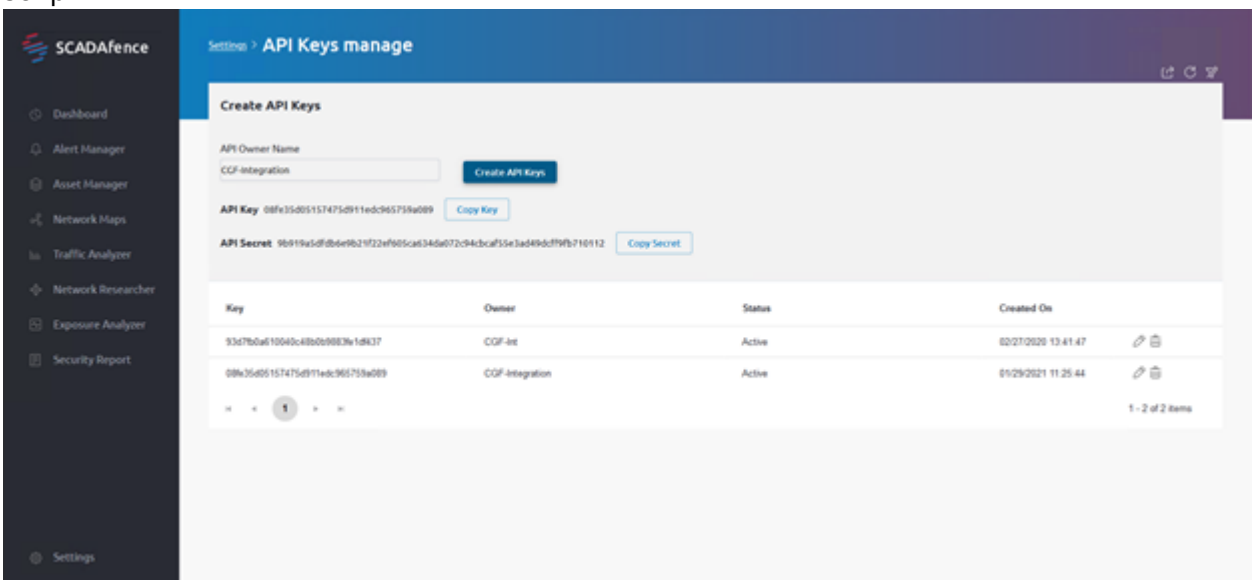
SCADAfence Platform is a non-intrusive continuous network monitoring solution that was purposely built to provide the required visibility and security for organizations adopting advanced Industrial IoT and OT technologies. The integration with the Barracuda CloudGen Firewall extends the capability of a passive approach to anomaly detection to include an automated active response to block suspicious traffic. The API-based approach ensures flexibility on the actions performed.

Step 1: Configure REST API Access to Your Firewall or Control Center

For more information on how to configure REST API access, see REST API.

Step 2: Configure the API Access on the SCADAfence Instance

1. Log into your SCADAfence instance.
2. Navigate to **Settings > Manage External API Keys**.
3. Create a new key. (The secret is displayed during creation only.)
4. Copy the **API Key** and the **API Key Secret**. You will need these parameters in the integration script.



Step 3: Create API Key for the Firewall or Firewall Control Center

- For more information on how to configure Administrator accounts on a Control Center, see How to Create a CC Admin to Access the REST API.
- For more information on how to configure Administrator accounts on a stand-alone firewall, see How to Create a New Administrators Account.

Integration Script

This is an example script written in Python. It can run either on the firewall or the SCADAfence instance. API integration ensures a very flexible and customizable integration.
In this example, the source IP for new industrial Communications and PLC Stop commands can be blocked.

```
import requests
import json
import time
import urllib3
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
# Set your own parameters here:
SCADAfenceIP="IP"               #<SCADAfence Instance IP
SECRET = "12345678ABCDEF12345678ABCDEF12345678ABCDEF12345678ABCDEF"
#<SCADAfence API Secret key
KEY="08fe08fe08fe08fe08fe"              #SCADAfence API Key
# Global setup
SEVERITY_LEVEL = {
                'Information': 1,
                'Warning': 2,
                'Threat': 3,
                'Severe': 4,
                'Critical': 5
}
SEARCH_IPS=[
        {
        'IP': "192.168.0.222",
        'alerts': 0,
        'blocked': 0
        }
        ]
PARAMS = {'status': "Created"} # if you open the alerts, the status will move
to: InProgress.  if you resolve them: "Resolved"
HEADERS = {"x-api-key": KEY, "x-api-secret": SECRET, "Accept":
"application/json"}
SCADAFENCE_URL = "https://"+SCADAfenceIP+"/externalApi/alerts"
#Barracuda API Call
```

```
BARRACUDA_IP="IP:8443"
BARRACUDA_KEY="twCEEtwCEEtwCEEtwCEEtwCEE"
BARRACUDA_HEADERS_GET = {"X-API-Token": BARRACUDA_KEY, "Content-Type":
"application/json"}
BARRACUDA_HEADERS_POST = {"X-API-Token": BARRACUDA_KEY, "Content-Type":
"application/json", "accept": "*/*"}
BARRACUDA_PARAMS_POST = {"envelope": "true"}
def block_IP(bl_ip, blocked):  # Change as needed for Barracuda
        hostName = 'SCADAfence_Malicious_User' + str(blocked)
        BARRACUDA_HOST_POST_URL = "https://" + BARRACUDA_IP +
"/rest/config/v1/forwarding-firewall/objects/networks"
        BARRACUDA_HOST_POST = {"name": hostName, "included": [{"entry":
{"ip": bl_ip}}]}
        BARRACUDA_RULE_POST_URL = "https://" + BARRACUDA_IP +
"/rest/config/v1/forwarding-firewall/rules"
        BARRACUDA_RULE_POST = {
                        "name": "BLOCK-SCADAfence-Malicious-Host",
                "source": {
                                "references": hostName,
                },
            "destination": {
                        "references": "Any"
            },
            "service": {
                                        "references": "Any"
                        },
                        "action": {
                        "type": "block"
            },
            "position": {
                        "placement": "top"
            }
            }
        res_post = requests.request("POST",
                    BARRACUDA_HOST_POST_URL,
                    verify=False,
                    params=BARRACUDA_PARAMS_POST,
                    json=BARRACUDA_HOST_POST,
                    headers=BARRACUDA_HEADERS_POST
            )
        print ("res_post = " , res_post.url)
        print (res_post.text)
        time.sleep(2)
        res_post = requests.request("POST",
                        BARRACUDA_RULE_POST_URL,
                        verify=False,
```

```
                        params=BARRACUDA_PARAMS_POST,
                        json=BARRACUDA_RULE_POST,
                        headers=BARRACUDA_HEADERS_POST
                    )
        print ("res_post = " , res_post.url)
        print (res_post.text)
####SCADAFence
#Scadafence API Configuration
def trigger_intergration_alert(ip_addr, alert_name):
    HEADERS = {"x-api-key": KEY, "x-api-secret": SECRET, "Accept":
"application/json"}
    SCADAFENCE_URL = "https://"+SCADAfenceIP+"/externalApi/alerts"
    SCADAFENCE_URL_POST="https://"+SCADAfenceIP+"/externalApi/alert"
    DATA_POST = {'severity': 'Critical', 'ip': ip_addr, 'description':
'Incident Detected - Informing Barracuda', 'explanation' : 'System detected
suspicious activity: ' + alert_name, 'remediation' : 'Check the affected
device for unauthorized activities.', 'details': alert_name ,"active" : True}
    PARAMS_POST = {}
    res_post = requests.request("POST",
                SCADAFENCE_URL_POST,
                verify=False,
                params=PARAMS_POST,
                data=DATA_POST,
                headers=HEADERS
                )
    print ("res_post = " , res_post)
def dot_sleep(ttime, tinterval):
    for t in range(1,ttime):
        print('.', end='', flush=True)
        time.sleep(tinterval)
    print ('')
def main():
    # ---- first checking about alerts in SCADAfence
    try:
        print ('Looking for trouble...', end='')
        # dot_sleep (20,0.3)
        alerts=0 #a counter of alerts. not used in this scenario, can be used
for aggregation of alerts.
        blocked=0 # counter of blocked devices. also serves as unique suffix
for barracuda object names
        new_src_alert=0
        plc_stop_alert=0
        while blocked < 2:   # will only handle blocking 2 devices. In
production logic can be different.
                res = requests.request("GET",
                                            SCADAFENCE_URL,
```

```
                                            verify=False,
                                            params=PARAMS,
                                            headers=HEADERS
                                            )
                if not res.text:
                    print ("ERROR: API returned empty")
                    return None
                x = json.loads(res.text)
                for val in x:
                        IP = val["ip"]
                        # print ("VAL:", val)
                        if val["type"] == "New Source IP Connecting to
industrial device" and new_src_alert == 0:
                                print ("--- Found <" + val["type"] + "> with
severity <" + val["severity"] + "> for " + IP)
                                print ("!!! " + IP + " performs suspicious
activity that looks like an on-going security incident.")
                                new_src_alert = 1
                                alerts = alerts + 1
                                trigger_intergration_alert(IP, val["type"])
                                print ("!!! Blocking firewall access for: ["
+ IP + "]")
                                blocked = blocked + 1
                                block_IP (IP, blocked)  #Call Barracuda Block
Action
                                time.sleep(1)
                        if val["type"] == "PLC stop command issued" and
plc_stop_alert == 0:
                                print ("--- Found <" + val["type"] + "> with
severity <" + val["severity"] + "> for " + IP)
                                print ("!!! " + IP + " performs suspicious
activity that looks like an on-going security incident.")
                                plc_stop_alert = 1
                                alerts = alerts + 1
                                trigger_intergration_alert(IP, val["type"])
                                print ("!!! Blocking firewall access for: ["
+ IP + "]")
                                blocked = blocked + 1
                                block_IP (IP, blocked) #Call Barracuda Block
Action
                                time.sleep(1)
                        if blocked==0:
                                print ("No new threat found...")
                                time.sleep(5)
    except (Exception, e):
        raise e
```

```
if __name__ == '__main__':
    main()
```

Figures