How to Integrate the Barracuda CloudGen Firewall with Rhebo

https://campus.barracuda.com/doc/98219128/

Combining Rhebo network monitoring with the Barracuda CloudGen Firewall extends the passive monitoring approach to active blocking of suspicious traffic. The API-based approach ensures flexibility on possible actions. In order to block malicious traffic detected by Rhebo on the CloudGen Firewall, you must enable the API to receive that information. In addition, the Barracuda Secure Connector can also be leveraged as a sensor to collect network data in dispersed environments through Edge Computing technology. To use the Barracuda Secure Connector to collect and stream network activity to the Rhebo Controller in a dispersed environment, you must enable and configure the LXC container functionality on the Secure Connector.

Before You Begin

- Enable REST API on your Barracuda CloudGen Firewall or Firewall Control Center. For more information, see REST API.
- Enable Edge Computing on the Barracuda Secure Connector. For more information, see Secure Connector Container.

Step 1. Create an API Key

Create an API key for the CloudGen Firewall or Firewall Control Center.

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

Make a note of the created API token.

Step 2. Configure Rhebo Controller

In order to block malicious traffic discovered by the Rhebo solution, the information will be passed via API to the Barracuda CloudGen Firewall. For detailed setup instructions on the Rhebo Controller. please contact Rhebo.

1. Enable CEF syslog streaming.
2. In order to run the script, make sure that the following modules are installed on the Rhebo

Controller:

- python3-systemd
- python3-requests

Example Script:

```
import systemd.journal
import requests
import json
import time
import urllib3

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
def notificationTrigger(type):
notifications = {
    'ICMP_ADDRESS_SCAN': True,
    'TCP_SYN_PORT_SCAN': True,
    'TCP_SYN_ADDRESS_SCAN': True,
    'UDP_ADDRESS_SCAN': True,
    'UDP_PORT_SCAN': True,
    }
return notifications.get(type, False)
######Configure Barracuda API#########
BARRACUDA_IP="<BarracudaIP:8443"
BARRACUDA_KEY="<API-Access-Tocken"
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"}
######Configure Rhebo Trigger Event#######

CEF_IDX_NOTIFICATION_TYPE = 5
CEF_IDX_NOTIFICATION_PARAMS = 7
CEF_SOURCE_IP_IDENTIFIER = "dvc="

def block_IP(bl_ip, blocked):
    hostName = 'Rhebo_Malicious_User_' + str(blocked)
        BARRACUDA_HOST_POST_URL = "http://" + BARRACUDA_IP +
"/rest/config/v1/forwarding-firewall/objects/networks"
        BARRACUDA_HOST_POST = {"name": hostName, "included": [{"entry":
    {"ip": bl_ip}}]}
        BARRACUDA_RULE_POST_URL = "http://" + BARRACUDA_IP +
"/rest/config/v1/forwarding-firewall/rules/lists/rhebo"
        BARRACUDA_RULE_POST = {
                "name": "BLOCK-Rhebo-Malicious-Host-" + str(blocked),
                "source": {
```

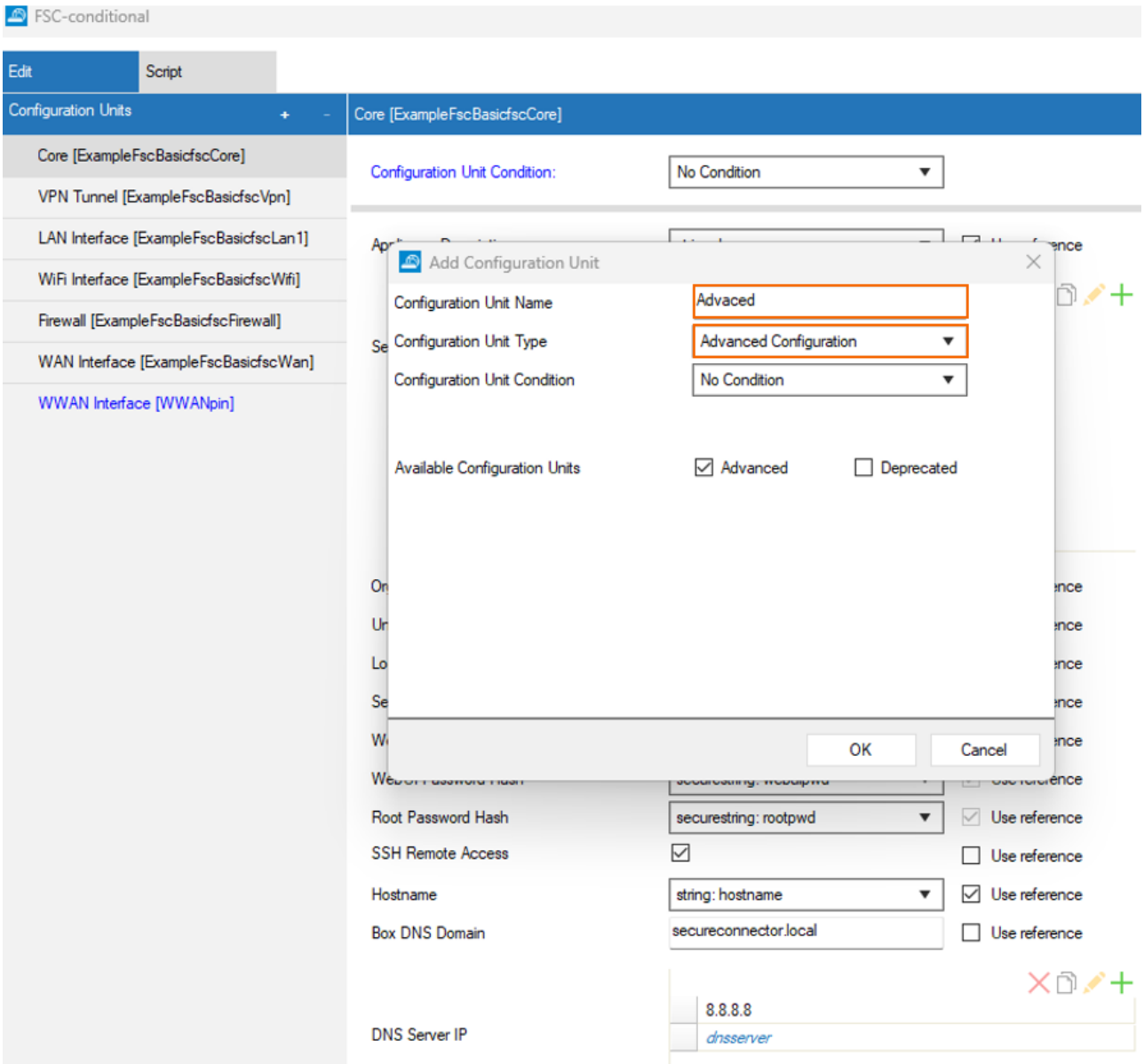
```
                "references": hostName,
                },
                "destination": {
                     "references": "Any"
                },
                "service": {
                "references": "Any"
                },
                "action": {
                        "type": "block"
                },
                "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)

def parseSourceIP(CEFparams):
    return CEFparams[CEFparams.find(CEF_SOURCE_IP_IDENTIFIER) +
len(CEF_SOURCE_IP_IDENTIFIER):].split(' ')[0]
def main():
    j = systemd.journal.Reader()
        j.seek_tail()
    j.get_previous()
    while True:
          event = j.wait(-1)
          if event == systemd.journal.APPEND:
         for entry in j:
          if '_COMM' in entry and entry['_COMM'] == 'baldrick':
              lineList = entry['MESSAGE'].split('|')
                  if len(lineList) >= 8:
                      if notificationTrigger(lineList
[CEF_IDX_NOTIFICATION_TYPE]):
                                                                    IP =
```

```
parseSourceIP(lineList
[CEF_IDX_NOTIFICATION_PARAMS])
                                        block_IP (IP, IP.replace('.', '-
'))
if __name__ == '__main__':
    main()
```

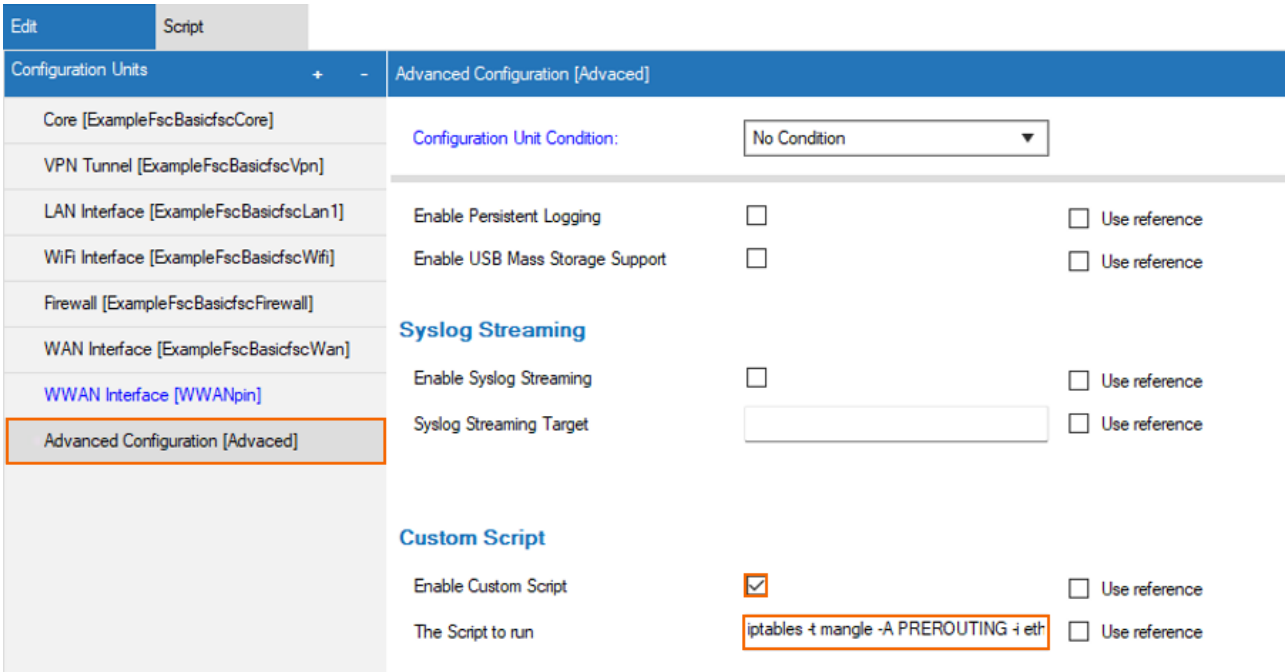
For detailed integration steps and customization options, please contact Barracuda Networks Technical Support or Rhebo.

Step 3. Configure the Secure Connector as Rhebo Sensor

1. On the Control Center, go to? ***your cluster?*** **> Cluster Settings > Configuration Templates**.
2. Verify that the container is enabled on the related template and set to **LXC**.
3. **Lock** the configuration template.
4. Double-click to edit the configuration template.
5. Click **+** and add a new **Advanced Configuration Unit**.



6. In the left menu, select the configuration unit you just created.
7. Select **Enable Custom Script** and set the script to mirror the traffic from related interfaces to container:
   - `iptables -t mangle -A PREROUTING -i eth0 -j TEE --gateway <containerIP>`



8. Click?**OK**.
9. Click **Send Changes** and?**Activate**.

Step 4. Install the Secure Connector Container

1. Go to **Control > Firmware Update**.
2. Modify *justin.yaml* within the container package *Rhebo_container.tgz* to match your environment.
3. Upload the sensor software *Rhebo_container.tgz* to your Secure Connector(s).
4. Install the container.

?For more information, see?Secure Connector Container.

Figures

© Barracuda Networks Inc., 2024 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.