cohesive networks

VNS3:turret
NIDS Guide
Sept 2015

Table of Contents

Introduction	3
Configurable Default NIDS Plugin	7
Customizing Default NIDS Plugin	14
Putting it All Together	22
For Developers / DevOps approach	25
Resources	33

Introduction

VNS3:turret provides container based network services

Isolated Linux containers within VNS3 allows Partners and Customers to embed features and functions safely and securely into their Cloud Network.

Proxy | Reverse Proxy | Content Caching | Load Balancer | IDS | Custom Container

VNS3 Core Components

Router | Switch | Firewall | Protocol Redistributor | VPN Concentrator | Scriptable SDN

Getting Help with VNS3

This document assumes you have a VNS3 Controller instance launched and running in a security group, network or similar that has the appropriate access rules included for normal VNS3 operations.

See the specific instructions for your cloud setup and instance launch on our Product Resources page.

Please review the VNS3 Support Plans and Contacts before sending support inquiries.

Requirements

You have a cloud or virtual infrastructure account that Cohesive Networks can use for enabling your access to the VNS3 Controller Images.

Ability to configure a client (whether desktop based or cloud based) to use OpenVPN client software.

You have agreed to the VNS3 Terms and Conditions.

Basic knowledge of Linux software installation and use of command line tools.

Configurable Default NIDs Plugin

VNS3:turret NIDS overview

The VNS3:Turret system uses popular threat detection rules from Sourcefire (Cisco) or Emerging Threats (Proofpoint) with the open source NIDS (network intrusion detection system) tool "Suricata". This combination was chosen due to simplicity of configuration and high performance. Suricata was developed for the United States Department of Homeland Security (DHS) by the Open Information Security Foundation (OSIF).

The Suricata:EmergingThreats combination is deployed to VNS3:turret using the containers mechanism. These instructions cover customisation of the container image that will be used so that customer keys and intrusion rule sets can be employed.

Please be familiar with the VNS3 plug-in configuration guide: https://cohesive.net/dnld/Cohesive-Networks_VNS3-3.5-Docker.pdf

Getting the Default NIDS Plug-In

The Linux Container default plug in is accessible at the following URL:

https://vns3-containers-read-all.s3.amazonaws.com/NIDS_suricata_Base/NIDS_suricata_Base.export.tar.gz

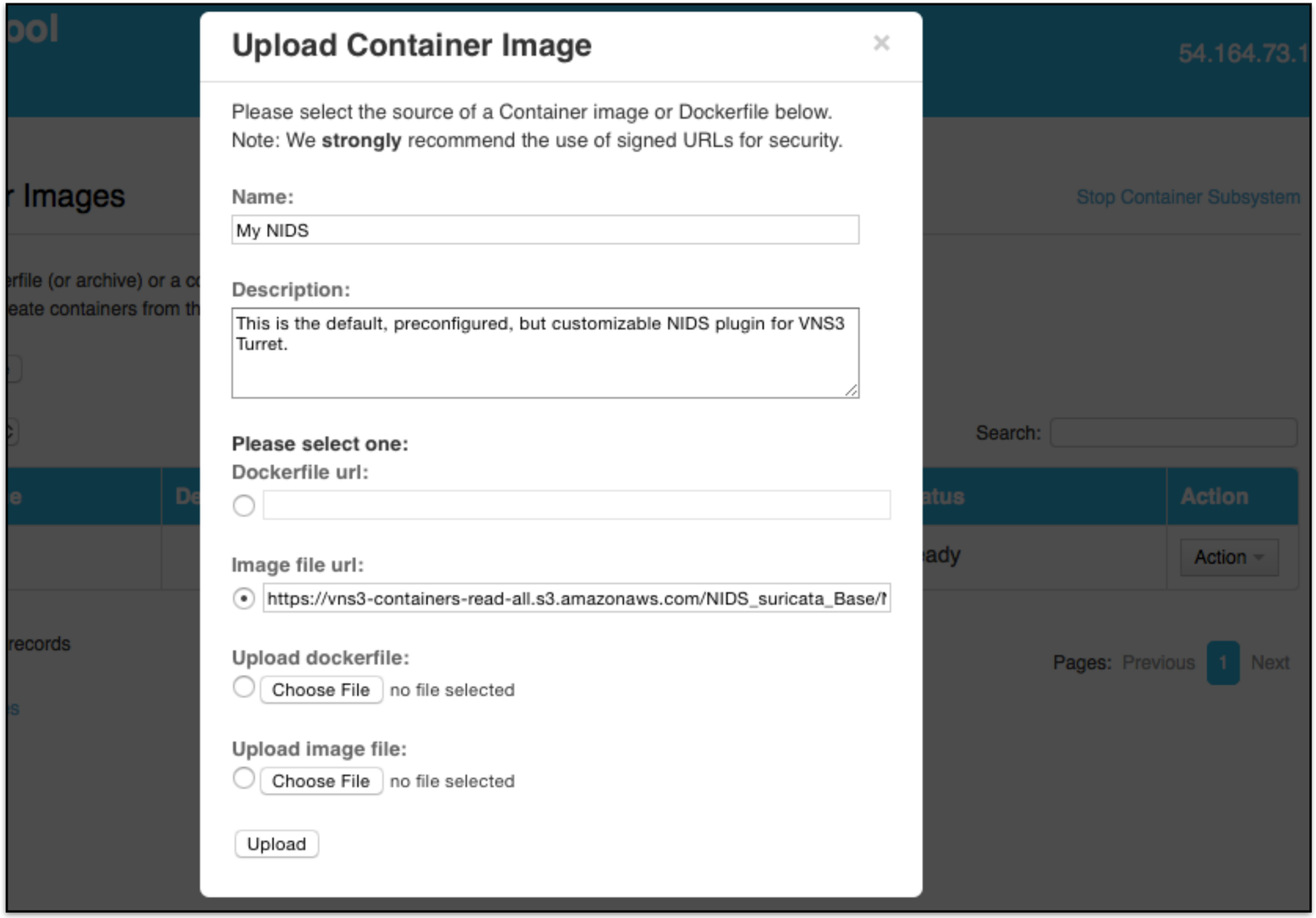
This is a read only Amazon S3 storage location. Only Cohesive Networks can update or modify files stored in this location.

This URL can be used directly in a VNS3 Controller via the Web UI or API to import the container for use into that controller. (General screenshot walkthrough and help available in the plug-in configuration document.)

Getting the Default NIDS Plug-In

From the “Container —> Images” menu item, choose “Upload Image”.

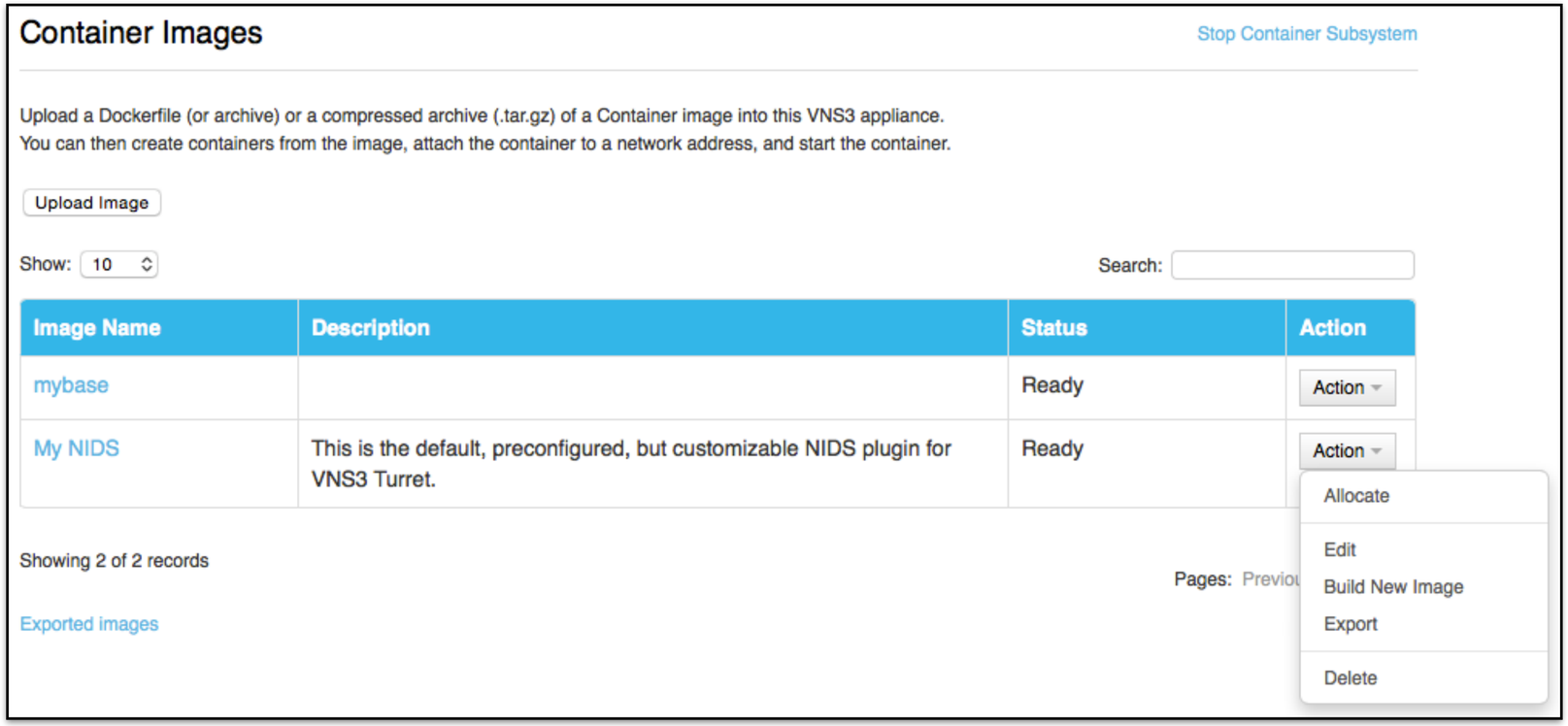
To use the pre-configured plugin paste the URL into the “Image File URL” box.



Getting the Default NIDS Plug-In

When the Image has imported it will say “Ready” in the Status Column.

To then launch a running NIDS container, choose “Allocate” from the “Action” menu.



Launching a NIDS Container

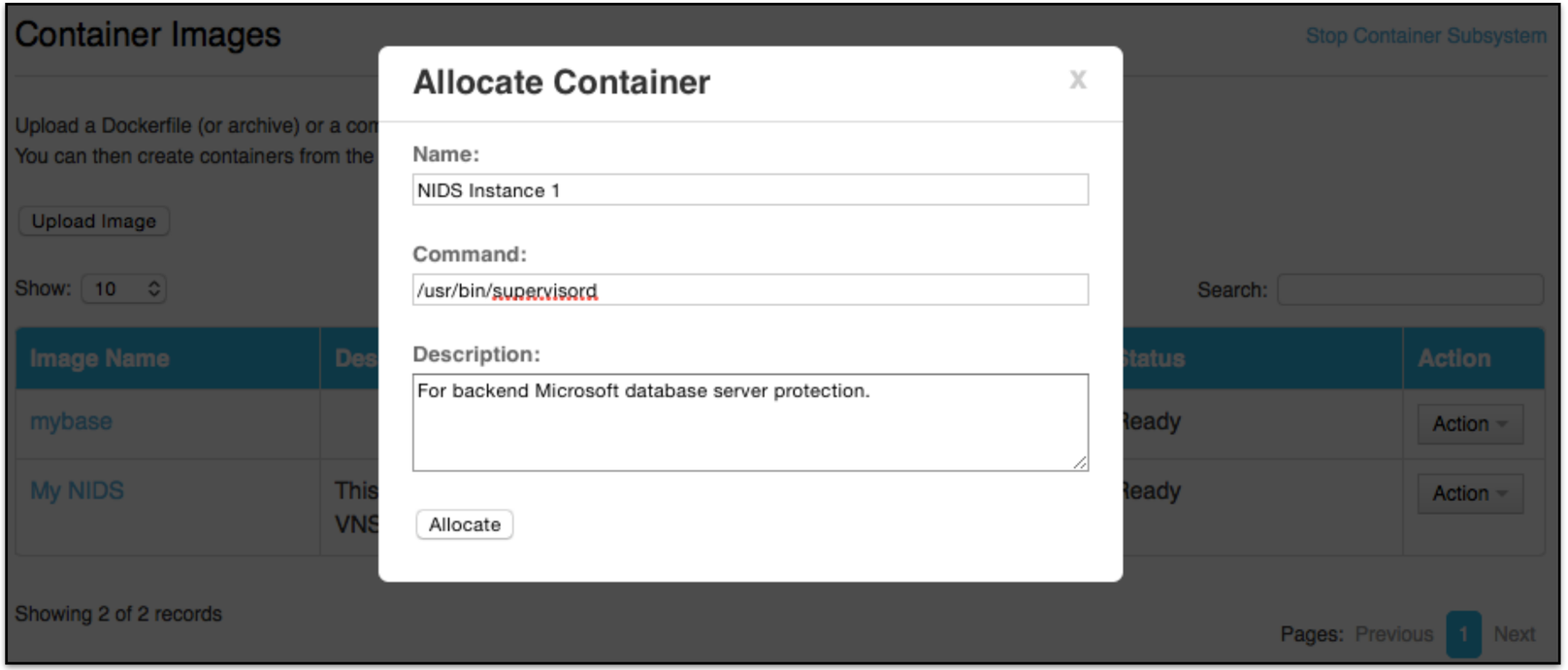
After selecting “Allocate” from the “Actions” menu you then name your container, provide a description and the command used to execute the container.

The name and description should be something meaningful within the context of your organization and its policies.

In MOST cases the command used to run plugin containers will be: /usr/bin/supervisord

However, this may vary with individual containers, please consult each plug-in’s specific documentation.

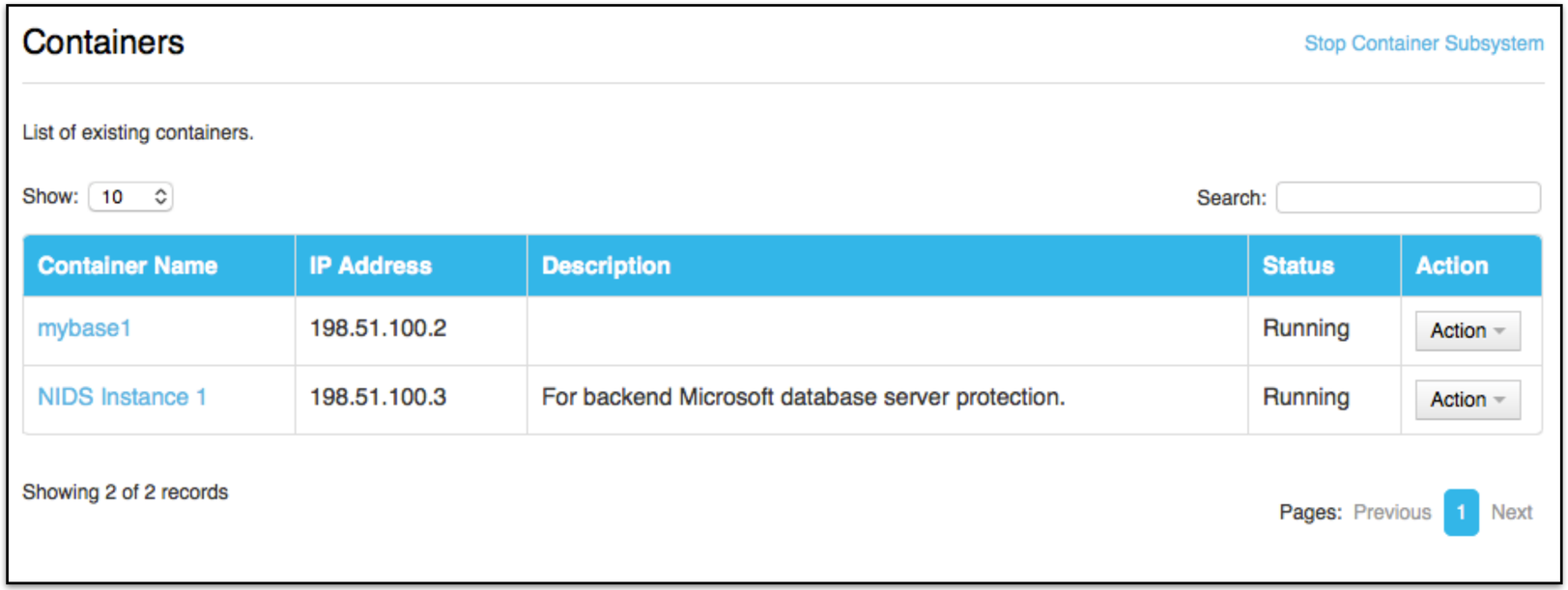
The command to run the NIDS container is: /usr/bin/supervisord



Confirming the NIDS Container is running

After executing the “Allocate” operation you will be taken to the Container Display page.

You should see your NIDS Container with the name you specified. The Status should be “Running” and it should have been given an IP address on your internal plug-in subnet (in this case 192.51.100.3).



Customizing Default NIDS Plugin

Accessing the NIDS Container

Accessing a Container from the Public Internet or your internal subnets will require additions to the inbound hypervisor firewall rules with the VNS3 Controller as well as VNS3 Firewall.

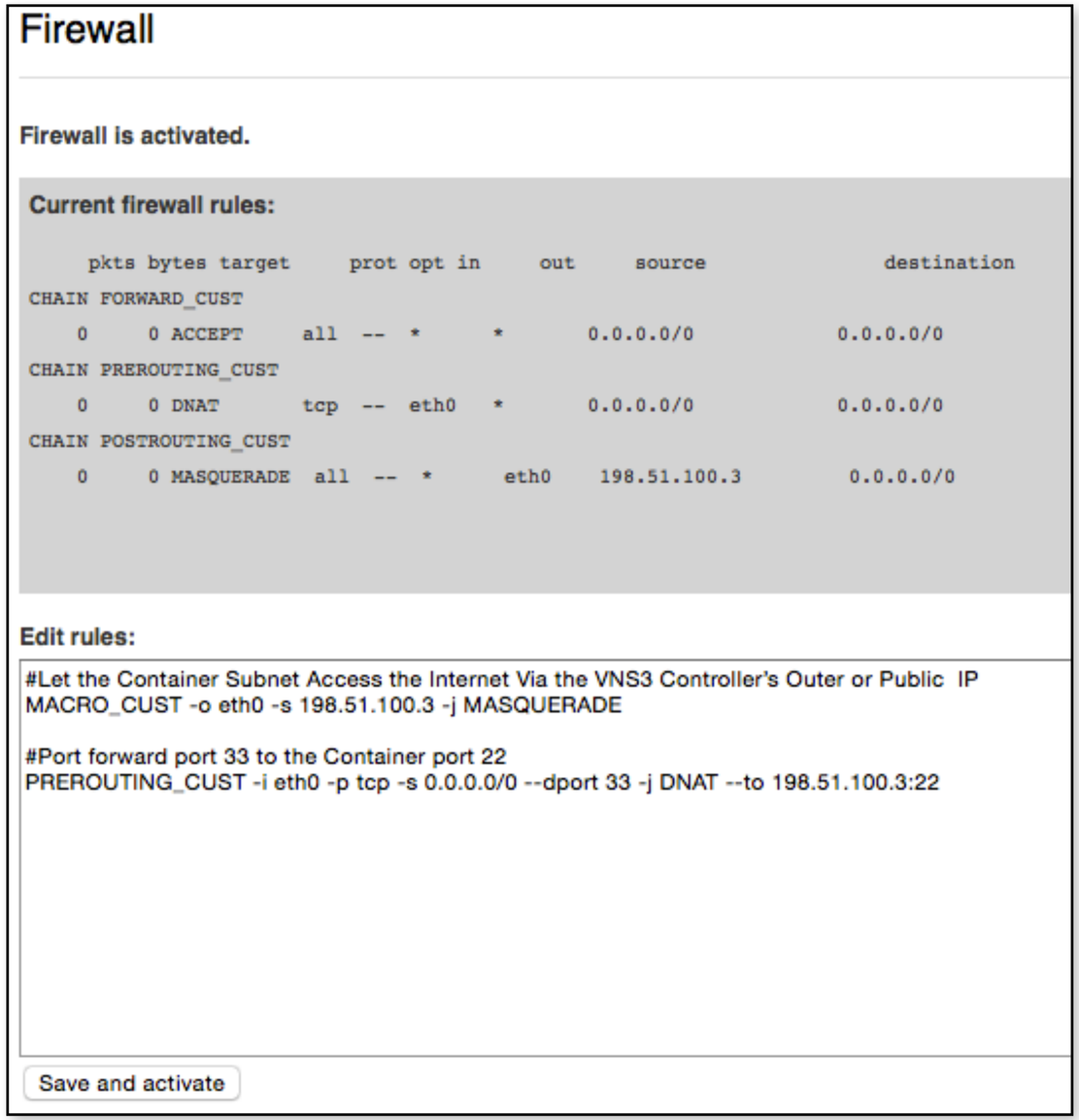
The following example shows how to access an SSH server running as a Container listening on port 22.

Network Firewall/Security Group Rule
Allow port 22 from your source IP or subnets.

VNS3 Firewall
Enter rules to port forward incoming traffic to the Container Network and Masquerade outgoing traffic off the VNS3 Controller's outer network interface.

#Let the Container Subnet Access the Internet Via the VNS3 Controller's Outer or Public IP
MACRO_CUST -o eth0 -s <NIDS Container Network IP> -j MASQUERADE

#Port forward port 33 to the WAF Container port 22
PREROUTING_CUST -i eth0 -p tcp -s 0.0.0.0/0 --dport 33 -j DNAT --to <NIDS Container Network IP>:22



Securing the NIDS container

By default the NIDS container has the following accounts, configured as described.

“root” - The root account is locked. The root account is not allowed to remote shell into the container. This is our recommended approach. However, if you wish to, you can use the “container_admin” account to unlock root, provide a root password, and edit /etc/ssh/sshd_config to allow remote login by root.

“container_admin” - The default password is “container_admin_123!”
The default demo public key is also installed in the /home/container_admin/.ssh/authorized_keys. **PLEASE change this password and this key** when configuring, or create a new default NIDS image as your base for future use, following your authentication procedures. The account “container_admin” has “sudo” or superuser privileges, and is allowed to remote shell into the container.

Accessing via the default private key

THIS IS FOR INITIAL / DEMONSTRATION ACCESS ONLY! Delete the contents of /home/container_admin/.ssh/authorized keys to secure your containers.

Here is the default private key for initial login.

-----BEGIN RSA PRIVATE KEY-----

MIIEoAIBAAKCAQEA1pIQ/2VxIR6DJx4/mKKfZJ2EuhAe+jJaXnbYMq33Zryum5ku

/r7KKcgR97R7GV0McHo23BJP/SoQrbyvIwRVBurnH32Okxl/ymX0YeudOlLh2/R/

palDnPVOtuQnY836poGxp3/X2H86/MgrHOclbeGy8Ezm6+zwnl18VccqiGYMW06c

a2qLGVMIh6WD03/p++l+QEPRmhAzfqWZJ02GG12lCK7ECODRELR0Y+ppe+yg2DaF

QI8gywRDa6l9v7BTEc5l/k3j2xqJxNXaBVzgjCJmVc7dfgfR1io31IHiTw1M8YPf

5lNpMdfiV4DjcG9f6GcUuO6uXgMZucnQT3ldfwIBIwKCAQAGIW4zLsi3zav5zaoL

rN/7j3jSHbe+AXBL14KFGunPvD+AydzFcypY9xZ0yqRucF9w7YyJ8eUHO7dVa8p9

V+UsFVcPhz6WfRJHnINTQT8Bqpi9JD4pTfqeFaMpzAEgG9P2IPZyf/7aTMcryzRu

ikLl4eCKhdq2SJkpGJ0nBbDCEX3p8H9jDWKlPxZ4vEbeqZeDMV+PPhVjUtrElAMB

amJY3/WmGPRH90pOO47vnZ+rSd/GLDpEuGYvjU7F64cBZUQbf4rYTCGW3dCyuw5g

iChEeiOvbYEYRffEh0/fv3Bn31qFteeY7HXOSAGrRm/KuUxejkTTs3RZBOjFLmBj

UuCrAoGBAPbWMrEueimj0zQcfxBlKFaph0DQQTFEXg0evgv+RitXdChooB9SmOe2

sOYbY36DX6V6QTzNsHOEoLuqdShPi3a9JIDyOAXdIBMTqt2SvywRBPJQffFoCJ+/

AbrfVr6Seu45C5t+aYlS8nULbphqp8Cvyof4ldV+5KyGtbllaNIPAoGBAN6JOoCy

G+Td38HpaML9J9xioizahbPBXj1/qyP3e+idSubqpT7feMCn3wOF+haNc2NF6VEN

qLTGEcKyAOA/TIySOel5rUZdpu5BmAVAADMeapMJWEXEbll4qJFd/sWJCP5wmZp/

lcSrDTLhcQJOci5LKSPOz/Czcpo9vOlVu8zRAoGAd+Rhw8YeFDmhGU+rbl0E9uSg

x7WcAfyitepcTvfY8HrvRtO7fO2aubCBztoaYgVLtsZaM3nZXK4iL0QqRseM4ebX

N1ET5ZdKF+T7OGvZMqkuSc9THXusatkeGPAi0Zeay3rLH6PM3EzcKjjAsG5RetkK

mdCDSnDVeF6wCZen9IUCgYAMt2JtwQjogbUDxDHfQaqBnzx3l3VaupervicJXpld

v9hk93coKgbmb/4ddV6/dcwUTSNGdc8gRdUhEXxklecd+boqmT0Z9rkU7c4sL4r7

m1aMDymdljIwIYX5rZmHoW46bNWTzMa6x/IgKiO2/SsYlpSi9d//IDJvNrpWee15

awKBgAczjW0Ag+nosFzklHhDAWIEZ+qgvdMcXf8pTOzgo0wyOl4SYTccp82Ffxee

25d8DyolvGgRjfDXKMyw7zfzwiknsZozEGNFDW+sgsPR9Pe1SQx07PtnUUflb3/C

v5LiLZmgW+RFvQf7lGqQpQSpfPuY6H8vwjxlA89SP3UwTi4N

-----END RSA PRIVATE KEY-----

Primary files for customization

There are two significant files for securing the NIDS container:

/etc/ssh/sshd_config
Please ensure this file is configured to your organization's best practices.

/home/container_admin/.ssh/authorized_keys
The base container comes with an example public key installed, and private key for use in VNS3 documentation. Please remove after initial use or programmatic configuration.

Primary files for customization

The following files are the major elements for customizing the NIDS container. It is not within the scope of this document to cover all elements of the included Suricata server, nor the collective elements of the Emerging Threats or Sourcefire rules sets.

/home/container_admin/.ssh/authorized_keys (already discussed)

/etc/suricata/custom.yaml
The custom.yaml file provides some primary settings for Siracata including pointing to the rules file, which by default is custom.rules. It also defines the location of the primary log file "fast.log".

On approximately line 65 you will see the entry defining the directory where log files go. The default is "/var/log/suricata/".

On approximately line 80 of the default custom.yaml you will see this monitoring enabled, and the log name default of "fast.log".

These two together define the default log as /var/run/suricata/fast.log

Primary files for customization

/etc/suricata/custom.rules
On approximately line 910 of custom.yaml are the entries defining the default directory for rules, and the default container demonstration rule file.

The default directory is "/etc/suricata/rules/" and the default rule file defined is custom.rule. Multiple rule files can be referenced in this section. This allows different source and types of rules to be used without needing to be combined into one input file.

The supplied custom.rules file contains a single demo rule designed to detect MasterCard numbers. The demo rule should be replaced by customised rules suitable for the application being protected. For example the entire free rule set is easily gotten from the Emerging Threats community site: http://rules.emergingthreats.net/open/suricata/emerging-all.rules Or with some editing done on the rule set, a subset of rules specific to a piece of infrastructure can be used. Here is an example of the approximate 20 rules out of the whole rule set that are specific to Microsoft SQLServer, https://vns3-containers-read-all.s3.amazonaws.com/NIDS_suricata_Base/mssql.rules

Primary files for customization

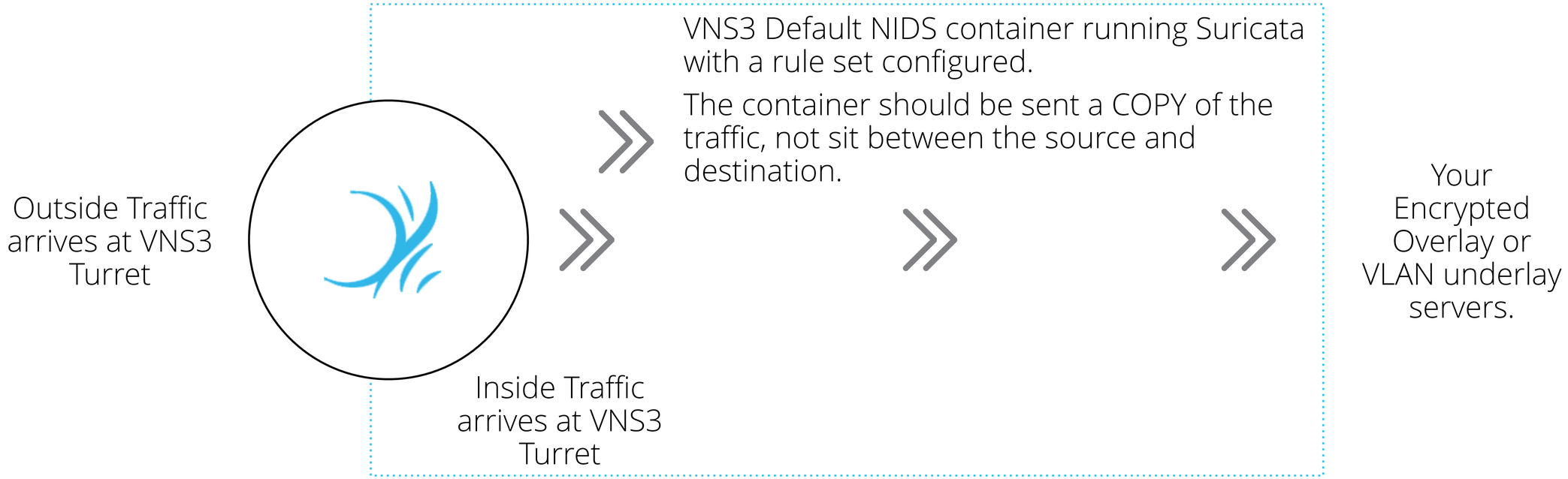
/etc/supervisor/conf.d/supervisord.conf

This file defines what services are started when the container is started. Looking at the default you will see Suricata, SSH, rsyslog.

Note: The rsylog component can be configured to copy information logged by the NIDS to an external syslog server.

Putting it all together -
Analyzing traffic via the VNS3 NIDS Plugin

NIDS Container Flow



User or interior traffic arrives at the VNS3 Controller. Firewall rules can filter and send a subset of traffic to the NIDS_suricata_Base container for analysis.

Forwarding Traffic to the NIDS Container

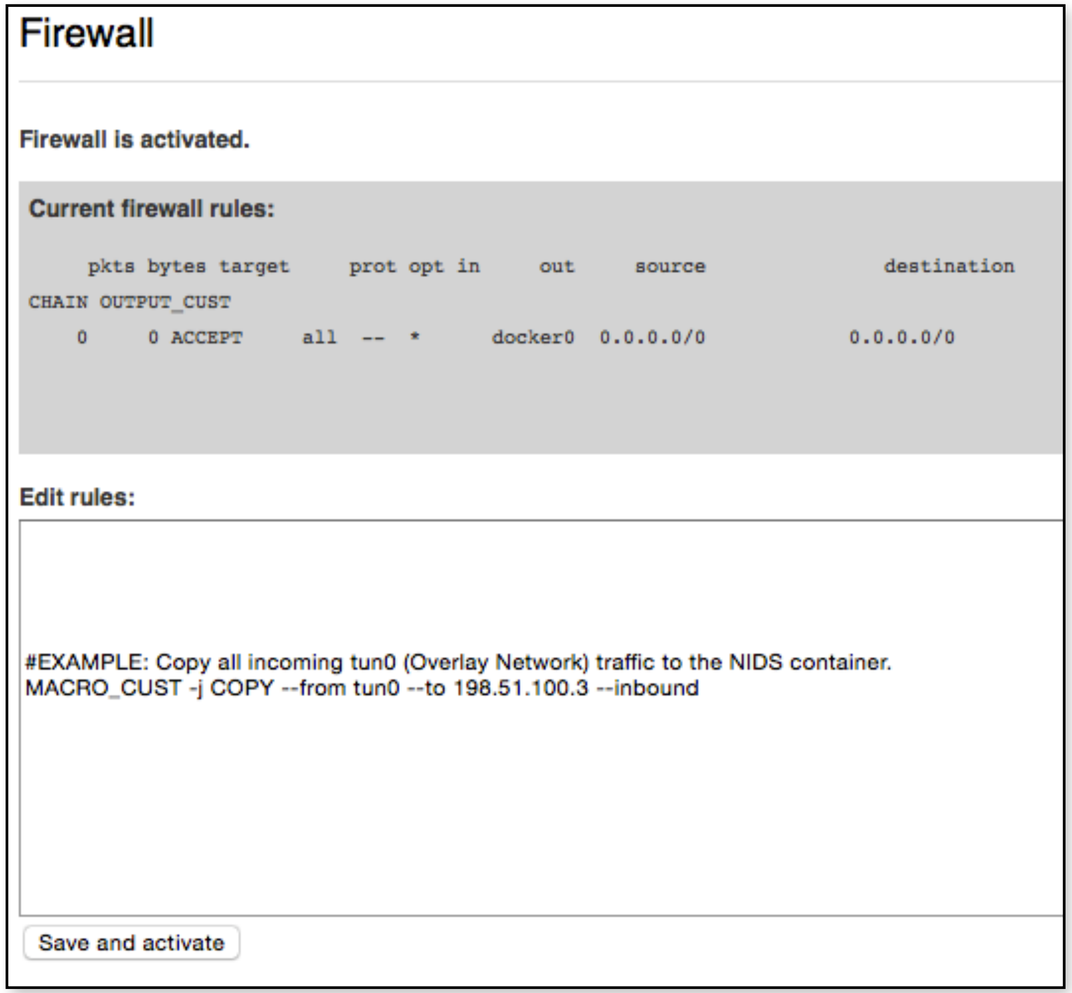
Forwarding traffic to the container uses the same technique as was shown for accessing the container via Remote Shell.

VNS3 Firewall
Enter rules to send a copy of either incoming traffic (arriving on eth0 or tun0) or outgoing traffic (leaving eth0 or tun0).

#EXAMPLE: Copy all incoming tun0 (Overlay Network) traffic to the NIDS container.
MACRO_CUST -j COPY --from tun0 --to <Container Network IP> --inbound

#EXAMPLE: Copy all outgoing tun0 (Overlay Network) traffic to the TCP Tools Container
MACRO_CUST -j COPY --from tun0 --to <Container Network IP> --outbound

NOTE: At this time analyze inbound OR outbound at any given time in order to prevent accidental traffic loops. It IS POSSIBLE to create a traffic cycle which could "brick" your controller if you create simultaneous inbound AND outbound rules with improper parameters.



For Developers / DevOps approach

Getting the NIDS container source

The Docker image source is distributed as a Dockerfile along with accompanying config files.

To get the source:

git clone https://github.com/cohesivenet/dockerfiles.git

cd suricata-custom

SSH access

Containers launched from the image that will be built use the included authorized_keys file to specify who can gain access to the container (as root).

Insert appropriate public keys e.g.:

cp ~/.ssh/id_rsa.pub authorized_keys

cat ~/.ssh/my_other_key.pub >> authorized_keys

If you need to generate a key then:

ssh-keygen -t rsa

Making a custom NIDS image

A customised Docker image can be built using:

sudo docker build -t cohesivenet/suricata-custom .

The tag ‘cohesivenet/suricata-custom’ may be replaced with something to suit your own environment and naming conventions.

To export a container image:

**SURICATA_CUSTOM=$(sudo docker run -d **
cohesivenet/suricata-custom)

sudo docker export $SURICATA_CUSTOM > suricata_custom.tar

gzip suricata_custom.tar

sudo docker kill $SURICATA_CUSTOM

Installing the custom NIDS image

(Detailed screenshots of these general plugin operations found in https://cohesive.net/dnld/Cohesive-Networks_VNS3-3.5-Docker.pdf)

First copy the nids-custom.tar.gz file to a URL capable server (Object Storage, Amazon S3, local WebDaV, Dropbox, etc) that's reachable from the VNS3:turret.

Click on the "Images" item in the Container section of the VNS3 menu.
Then select "Upload Image".

Give the image a Name: e.g. nids-custom

Paste the URL for the web server holding nids-custom.tar.gz into the Image file url: box.

Click "Upload"

Running the custom NIDS image

Once the Status of the imported image is Ready then click the “Action” button and select “Allocate”.

Give the container a Name: e.g. nids-custom

The command for running the container is: "/usr/bin/suricata -c /etc/suricata/custom.yaml -i eth0"

Click “Allocate”

Make a note of the IP Address given to the container e.g. 198.51.100.3

Running the Custom NIDS Image

This specifies that the custom.yaml file is used for config, and this in turn references custom.rules. The supplied custom.rules file contains a single demo rule designed to detect MasterCard numbers. The demo rule should be replaced by customised rules suitable for the application being protected.

For example a subset of the Emerging Threats free database rules for MS SQL could be employed:

wget -O custom.rules http://is.gd/mssqlrules

Routing traffic to the NIDS container

Click on the **Firewall** item in the **Connections** section of the VNS3 menu.

Add firewall rules such as:

```
MACRO_CUST -j COPY --from tun0 --to 198.51.100.2 --inbound
MACRO_CUST -o eth0 -s 198.51.100.0/28 -j MASQUERADE
PREROUTING_CUST -i eth0 -p tcp -s 0.0.0.0/0 --dport 2222 -j DNAT --to 198.51.100.2:22
```

Where 198.51.100.2 is the IP of the container once allocated. Then click "Save and Activate".

SSH is now available onto the container (on port 2222 of the VNS3:turret)

The /var/log/suricata/fast.log will now be recording NIDS events (and should be forwarded to a suitable security monitoring system).

Resources

Questions or Corrections for this document: support@cohesive.net

Questions about configuring the NIDS elements effectively: support@cohesive.net

More about Suricata:
http://suricata-ids.org/
http://oisf.net/

More about Emerging Threats Rules:
http://www.emergingthreats.net/open-source/etopen-ruleset