Overview

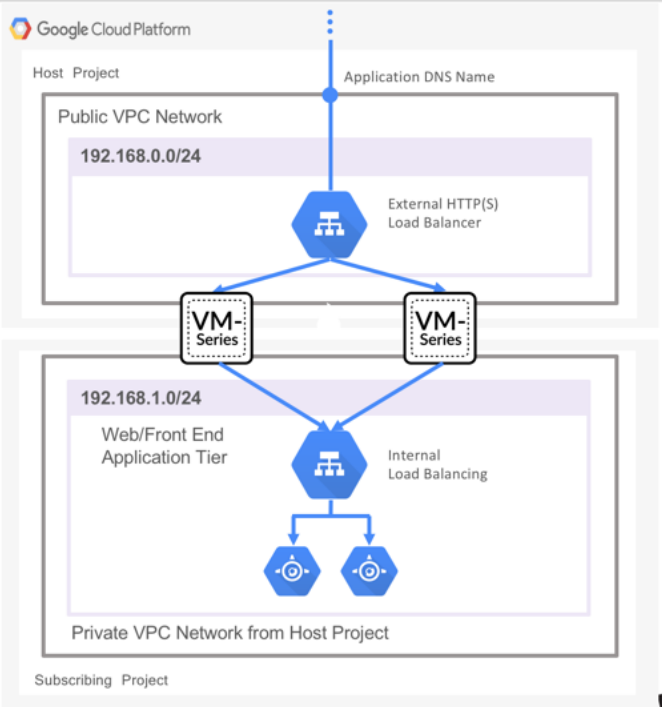
The following components are used in this demo:

- GCP Cloud Armor
- GCP Service Account
- Linux worker node

Prerequisites include:

- HTTP(S) Load Balancer
- Palo Alto Networks Firewall(s) with URLF and Threat subscriptions (duh)
- Webserver(s)

A typical deployment might resemble the diagram below (Shared VPC optional):



The gcp-aolf.py script may be downloaded from GitHub

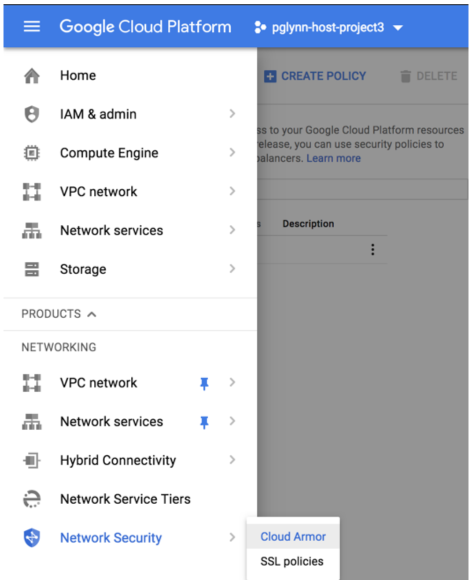
Introduction

1) In this demo we will use detectable threats between a browser and the web server (in this case we use a SQL Injection attack) to trigger action-oriented log forwarding.
2) The firewall will detect the SQL Injection threat and forward the log data to the worker node via an HTTP log forwarding action.

3) The worker node queries the firewall for additional information based on the sessionID, NAT Source Port, and received time of the detected threat. The response includes the IP address in the X-Forwarded-For HTTP header.
4) The worker node extracts the IP of the attacker from the X-Forwarded-For.
5) The worker node determines the correct rule priority and adds a rule to the Cloud Armor security policy.

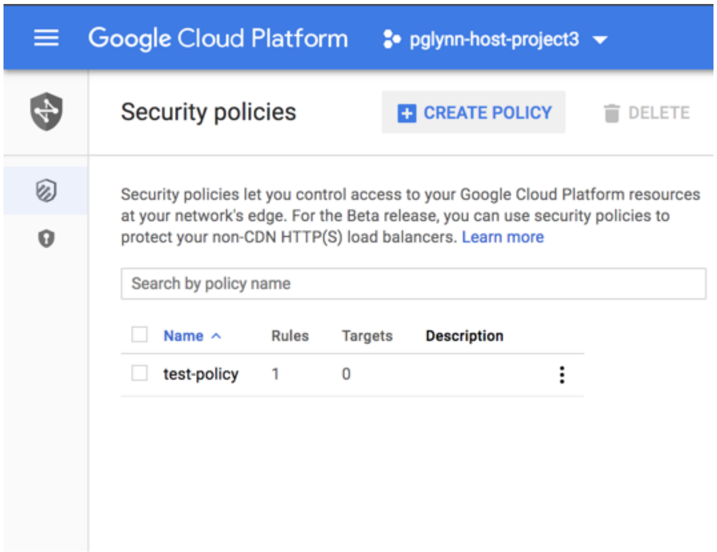
Initial Setup

1) Create the initial Cloud Armor security policy. Navigate to **Networking > Network Security > Cloud Armor**:



ss to your Google Cloud Platform resources elease, you can use security policies to alancers. Learn more

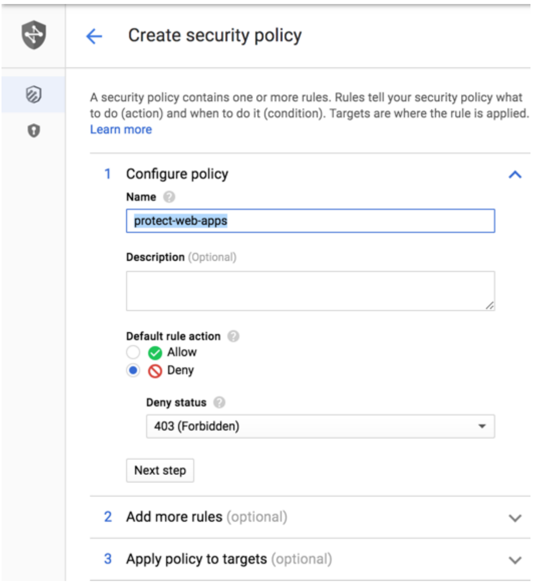
2) Click **CREATE POLICY**



Security policies let you control access to your Google Cloud Platform resources at your network's edge. For the Beta release, you can use security policies to protect your non-CDN HTTP(S) load balancers. Learn more

	Name	Rules	Targets	Description
□	test-policy	1	0	

3) Specify a **Name**



Create security policy

A security policy contains one or more rules. Rules tell your security policy what to do (action) and when to do it (condition). Targets are where the rule is applied. Learn more

1 Configure policy

Name: protect-web-apps

Description (Optional)

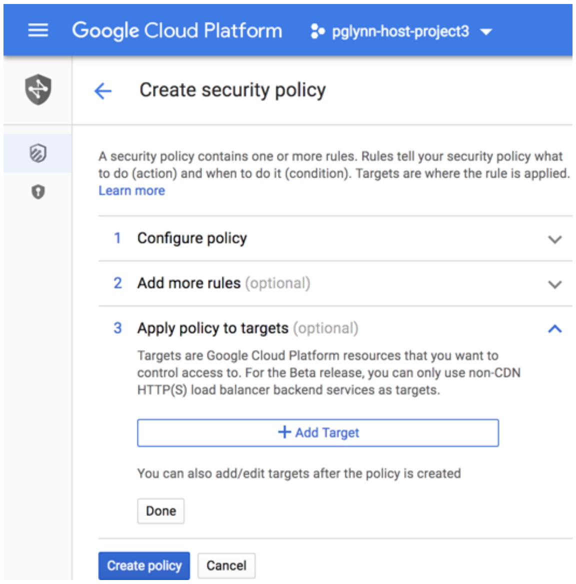
Default rule action
○ Allow
● Deny

Deny status: 403 (Forbidden)

2 Add more rules (optional)

3 Apply policy to targets (optional)

4) Click **Apply policy to targets**



Create security policy

A security policy contains one or more rules. Rules tell your security policy what to do (action) and when to do it (condition). Targets are where the rule is applied. Learn more

1 Configure policy

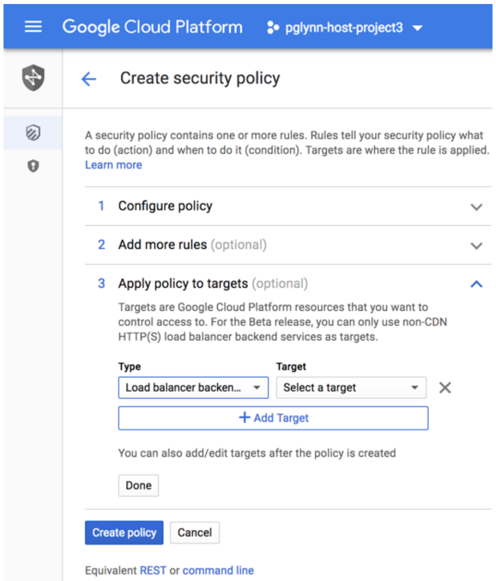
2 Add more rules (optional)

3 Apply policy to targets (optional)

Targets are Google Cloud Platform resources that you want to control access to. For the Beta release, you can only use non-CDN HTTP(S) load balancer backend services as targets.

You can also add/edit targets after the policy is created

5) Click **+ Add Target**



Google Cloud Platform pglynn-host-project3

Create security policy

A security policy contains one or more rules. Rules tell your security policy what to do (action) and when to do it (condition). Targets are where the rule is applied. Learn more

1 Configure policy

2 Add more rules (optional)

3 Apply policy to targets (optional)

Targets are Google Cloud Platform resources that you want to control access to. For the Beta release, you can only use non-CDN HTTP(S) load balancer backend services as targets.

Type	Target
Load balancer backen...	Select a target

+ Add Target

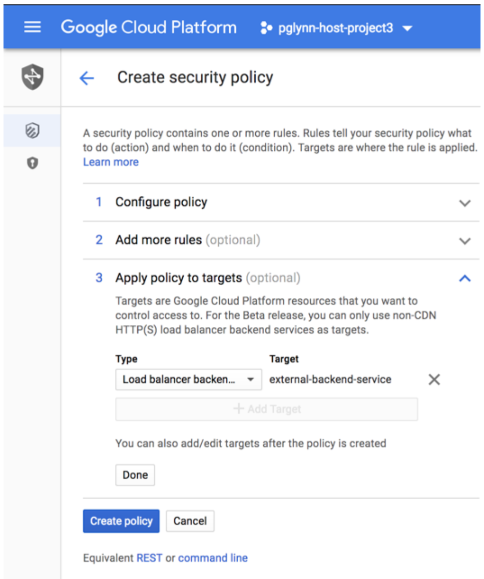
You can also add/edit targets after the policy is created

Done

Create policy Cancel

Equivalent REST or command line

6) Select the public HTTP(S) Load Balancer



Google Cloud Platform pglynn-host-project3

Create security policy

A security policy contains one or more rules. Rules tell your security policy what to do (action) and when to do it (condition). Targets are where the rule is applied. Learn more

1 Configure policy

2 Add more rules (optional)

3 Apply policy to targets (optional)

Targets are Google Cloud Platform resources that you want to control access to. For the Beta release, you can only use non-CDN HTTP(S) load balancer backend services as targets.

Type	Target
Load balancer backen...	external-backend-service

+ Add Target

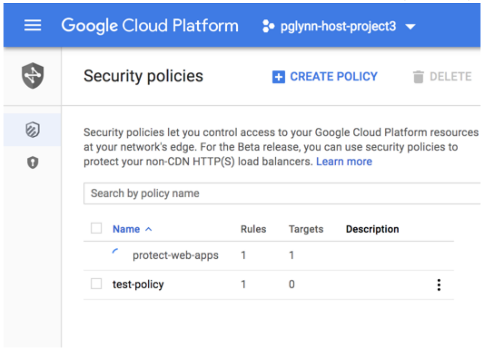
You can also add/edit targets after the policy is created

Done

Create policy Cancel

Equivalent REST or command line

7) Click **Create policy.** It will take a few moments to create the policy.



Google Cloud Platform pglynn-host-project3

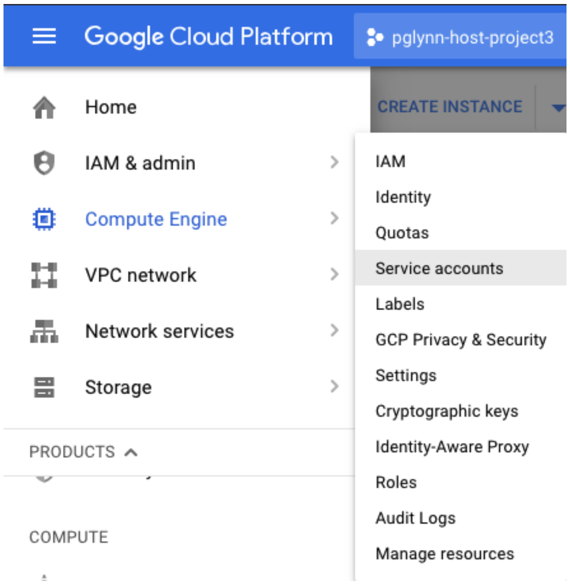
Security policies CREATE POLICY DELETE

Security policies let you control access to your Google Cloud Platform resources at your network's edge. For the Beta release, you can use security policies to protect your non-CDN HTTP(S) load balancers. Learn more

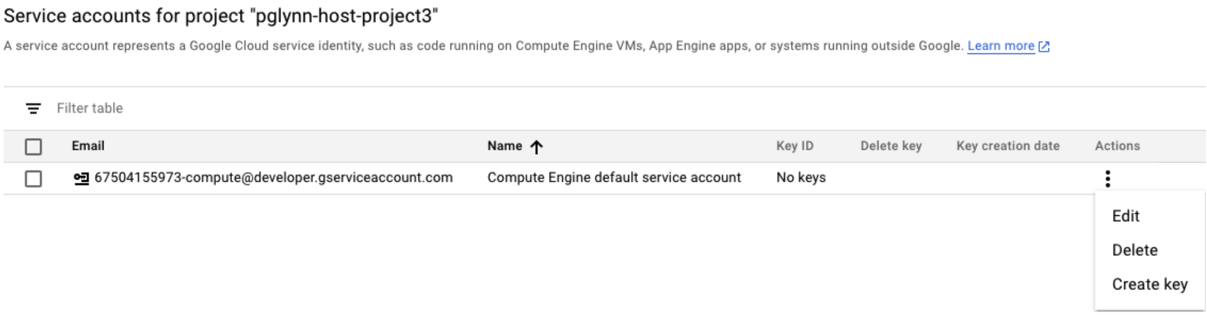
Search by policy name

Name	Rules	Targets	Description
protect-web-apps	1	1	
test-policy	1	0	

8) Create a service account key. Navigate to **IAM & admin > Service accounts**



9) Click on the vertical ellipses beside the default service account and select **Create key**



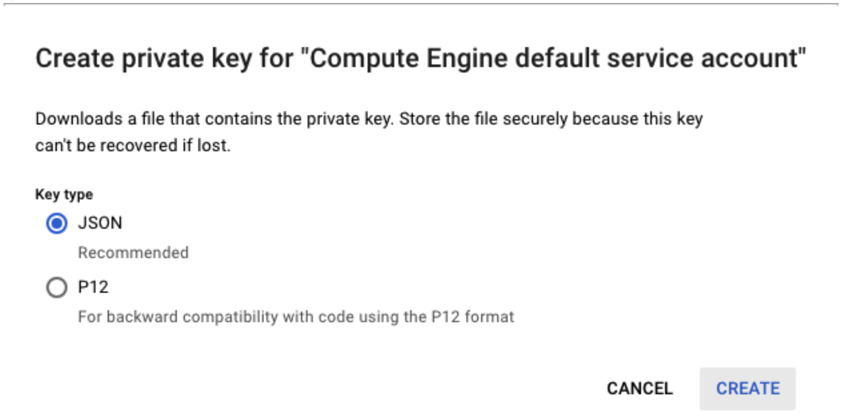
Service accounts for project "pglynn-host-project3"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more

Filter table

□	Email	Name ↑	Key ID	Delete key	Key creation date	Actions
□	67504155973-compute@developer.gserviceaccount.com	Compute Engine default service account	No keys			⋮

10) Leave the key type as JSON and click **CREATE**



Create private key for "Compute Engine default service account"

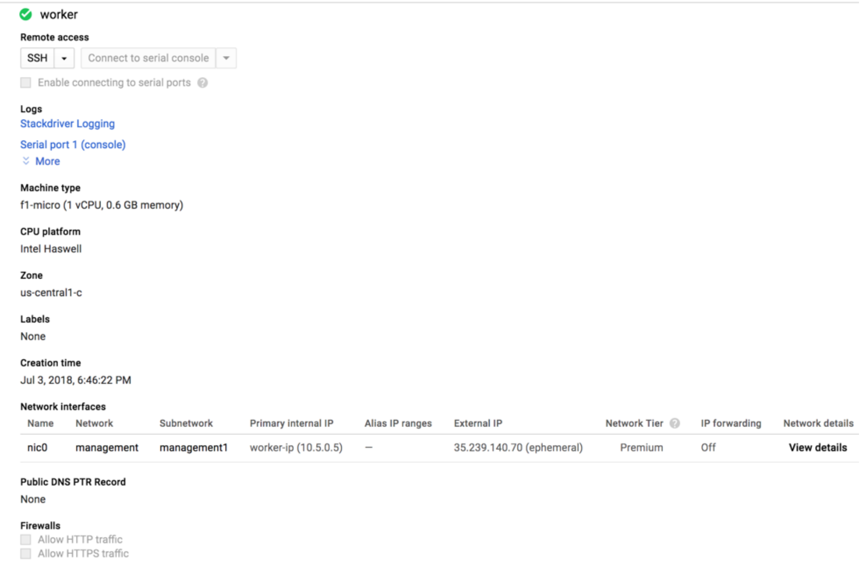
Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

● JSON
Recommended

○ P12
For backward compatibility with code using the P12 format

11) Save the key to a secure location as it allows API access to GCP resources
12) (optional) Rename the key to reflect the service account to which it is attached
13) Deploy a worker node with the following settings:
    a. Machine type: F1-micro
    b. Network: FW management subnet
    c. Internal IP: static
    d. External IP (optional): ephemeral



worker

Remote access

☐ Enable connecting to serial ports

Logs

Stackdriver Logging

Serial port 1 (console)

More

Machine type

f1-micro (1 vCPU, 0.6 GB memory)

CPU platform

Intel Haswell

Zone

us-central1-c

Labels

None

Creation time

Jul 3, 2018, 6:46:22 PM

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	management	management1	worker-ip (10.5.0.5)	—	35.239.140.70 (ephemeral)	Premium	Off	View details

Public DNS PTR Record

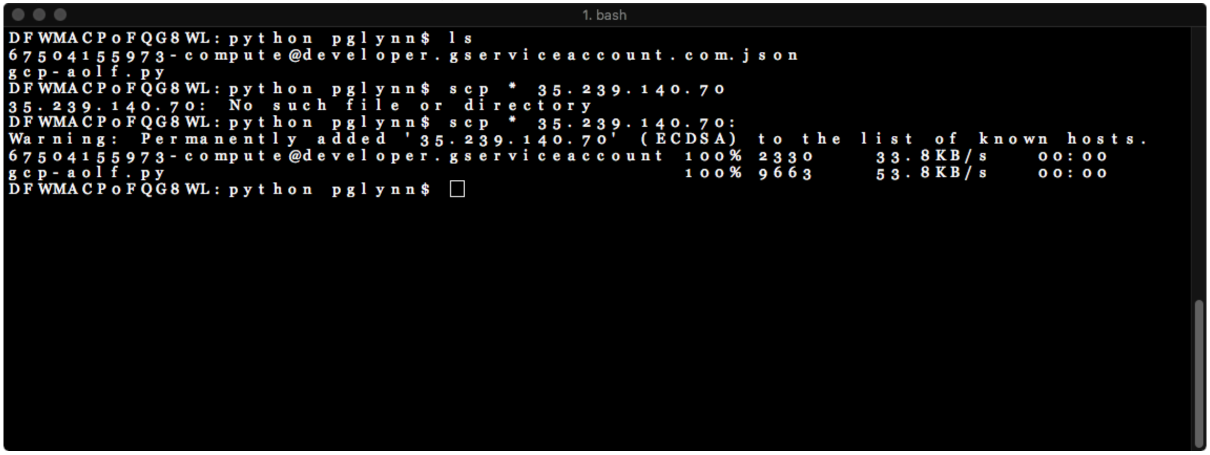
None

Firewalls

☐ Allow HTTP traffic

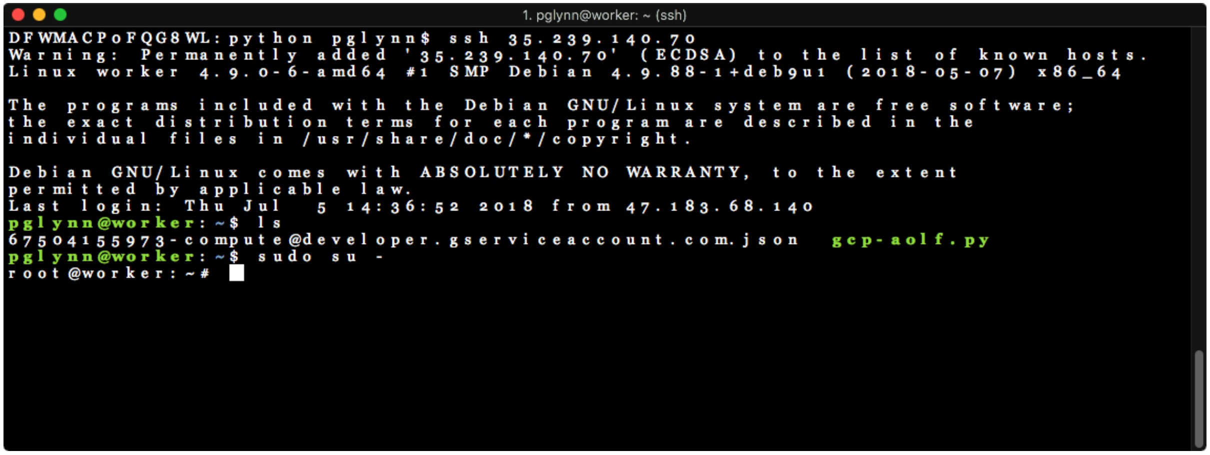
☐ Allow HTTPS traffic

14) Copy the service account key and Python code to the worker node



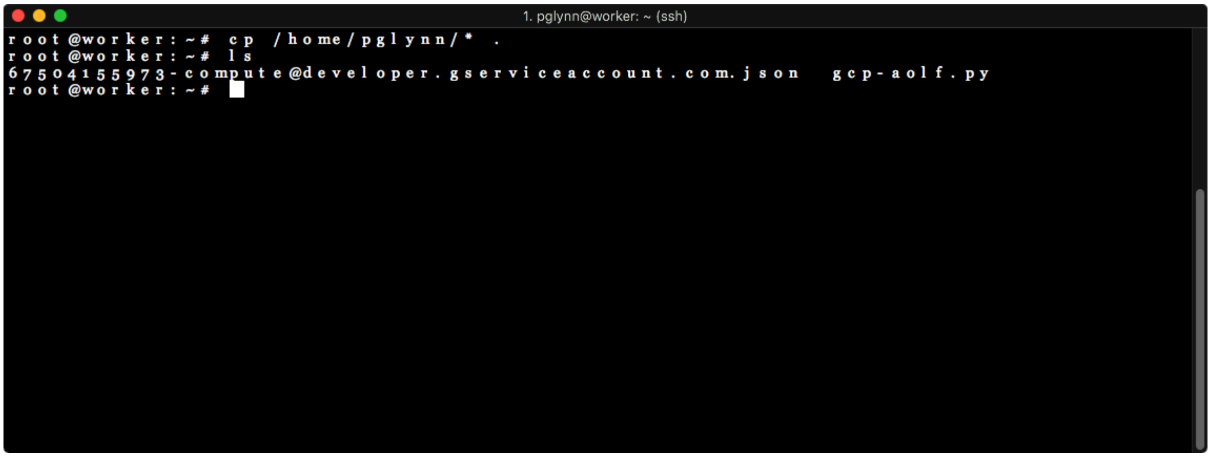
```
DFWMACPoFQG8WL:python pglynn$ ls
675041559973-compute@developer.gserviceaccount.com.json
gcp-aolf.py
DFWMACPoFQG8WL:python pglynn$ scp * 35.239.140.70
35.239.140.70: No such file or directory
DFWMACPoFQG8WL:python pglynn$ scp * 35.239.140.70:
Warning: Permanently added '35.239.140.70' (ECDSA) to the list of known hosts.
675041559973-compute@developer.gserviceaccount 100% 2330     33.8KB/s   00:00
gcp-aolf.py                                    100% 9663     53.8KB/s   00:00
DFWMACPoFQG8WL:python pglynn$
```

15) Connect to the worker node and “su” to root



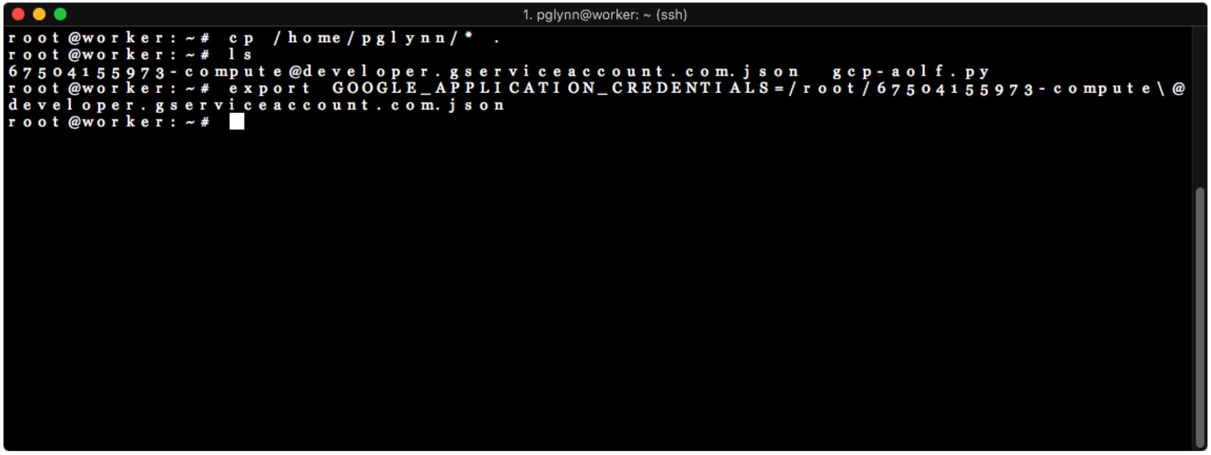
```
DFWMACPoFQG8WL:python pglynn$ ssh 35.239.140.70
Warning: Permanently added '35.239.140.70' (ECDSA) to the list of known hosts.
Linux worker 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1+deb9u1 (2018-05-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul  5 14:36:52 2018 from 47.183.68.140
pglynn@worker:~$ ls
675041559973-compute@developer.gserviceaccount.com.json  gcp-aolf.py
pglynn@worker:~$ sudo su -
root@worker:~#
```

16) Copy the Python script and service account key to root’s home directory



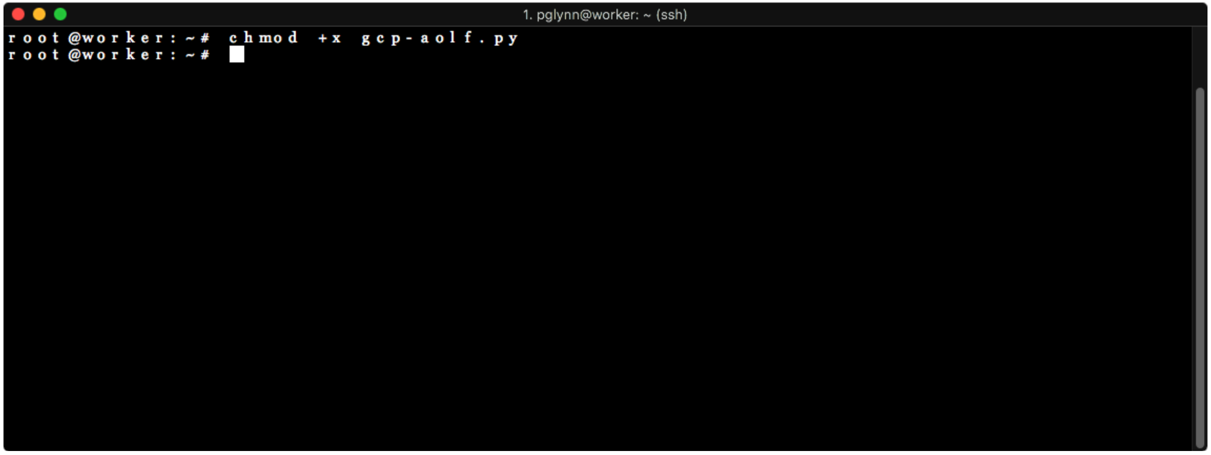
```
root@worker:~# cp /home/pglynn/* .
root@worker:~# ls
67504155973-compute@developer.gserviceaccount.com.json   gcp-aolf.py
root@worker:~# 
```

17) Create an environment variable pointing to the service account key. This is necessary as the Python script will need the key to authenticate to the GCP environment. The format is: export GOOGLE_APPLICATION_CREDENTIALS=/path/to/service_account_key.json



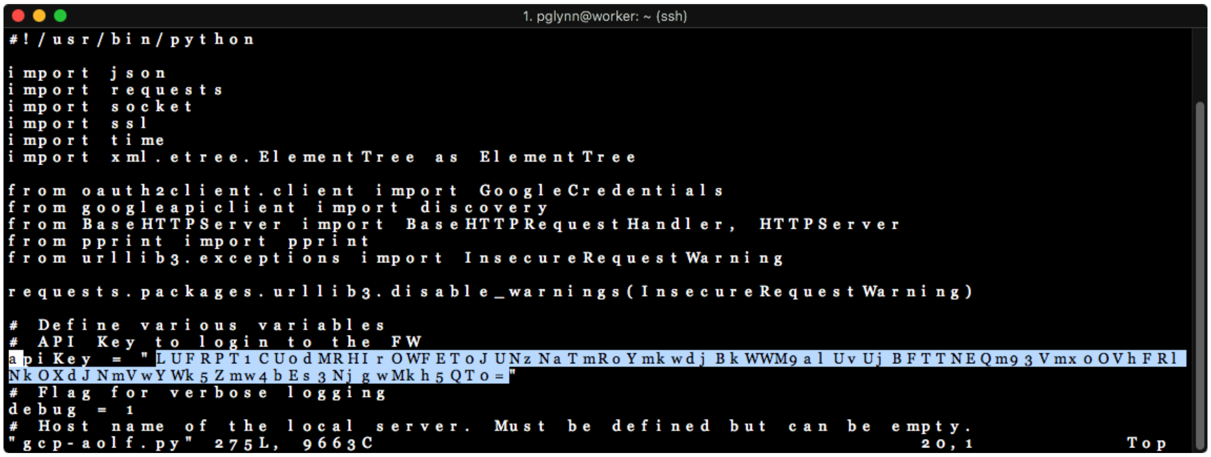
```
root@worker:~# cp /home/pglynn/* .
root@worker:~# ls
67504155973-compute@developer.gserviceaccount.com.json   gcp-aolf.py
root@worker:~# export GOOGLE_APPLICATION_CREDENTIALS=/root/67504155973-compute\@
developer.gserviceaccount.com.json
root@worker:~# 
```

18) Add execute permission to the Python script:



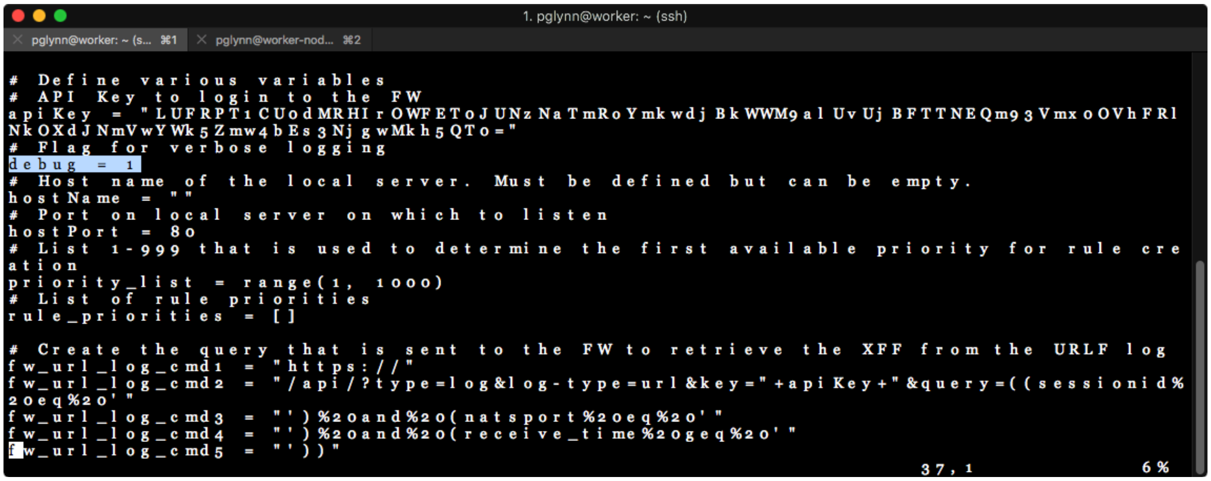
```
root@worker:~# chmod +x gcp-aolf.py
root@worker:~# 
```

19) Edit the Python script and replace the FW API key with the one specific to your implementation



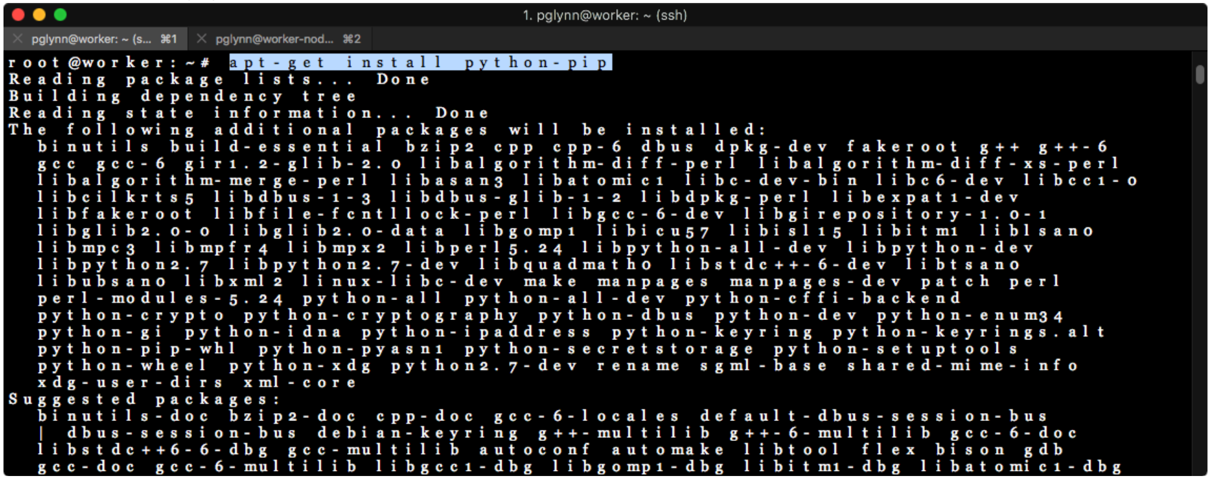
```
#!/usr/bin/python

import json
import requests
import socket
import ssl
import time
import xml.etree.ElementTree as ElementTree

from oauth2client.client import GoogleCredentials
from googleapiclient import discovery
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
from pprint import pprint
from urllib3.exceptions import InsecureRequestWarning

requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

# Define various variables
# API Key to login to the FW
apiKey = "LUFRPT1CU0dMRHIrOWFET0JUNzNaTmRoYmkwdjBkWWM9alUvUjBFTTNEQm93Vmx0OVhFRl
NkOXdJNmVwYWk5Zmw4bEs3NjgwMk5QT0="
# Flag for verbose logging
debug = 1
# Host name of the local server. Must be defined but can be empty.
"gcp-aolf.py" 275L, 9663C                                          20,1          Top
```

20) (optional) To monitor script execution or for debugging purposes, set the debug flag to "1"



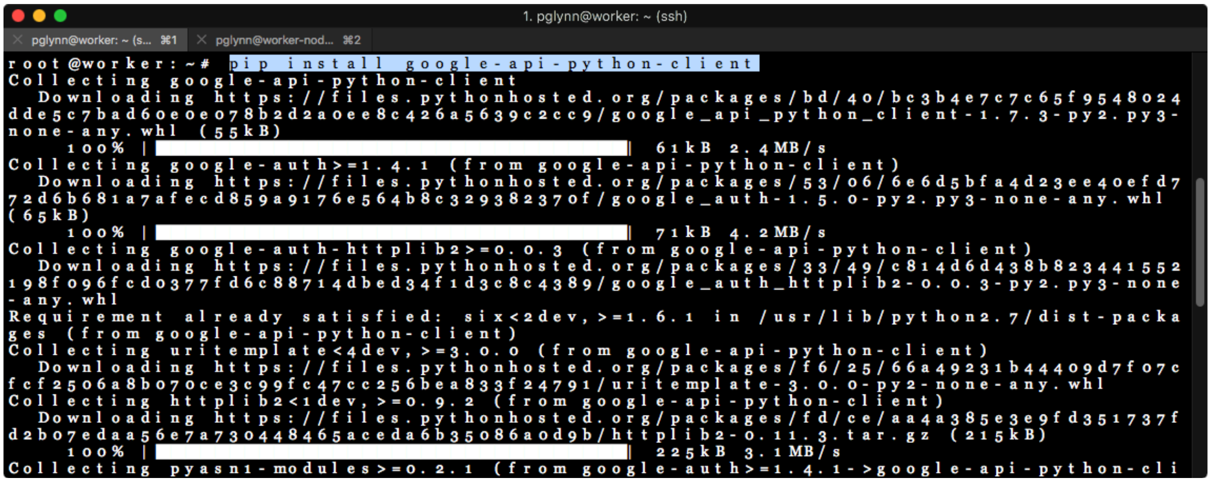
```

# Define various variables
# API Key to login to the FW
apiKey = "LUFRPT1CU0dMRHIrOWFET0JUNzNaTmRoYmkwdjBkWWM9alUvUjBFTTNEQm93Vmx0OVhFRl
NkOXdJNmVwYWk5Zmw4bEs3NjgwMk5QT0="
# Flag for verbose logging
debug = 1
# Host name of the local server. Must be defined but can be empty.
hostName = ""
# Port on local server on which to listen
hostPort = 80
# List 1-999 that is used to determine the first available priority for rule cre
ation
priority_list = range(1, 1000)
# List of rule priorities
rule_priorities = []

# Create the query that is sent to the FW to retrieve the XFF from the URLF log
fw_url_log_cmd1 = "https://"
fw_url_log_cmd2 = "/api/?type=log&log-type=url&key="+apiKey+"&query=((sessionid%
20eq%20'"
fw_url_log_cmd3 = "')%20and%20(natsport%20eq%20'"
fw_url_log_cmd4 = "')%20and%20(receive_time%20geq%20'"
fw_url_log_cmd5 = "'))"
                                                                   37,1          6%
```

21) Install pip



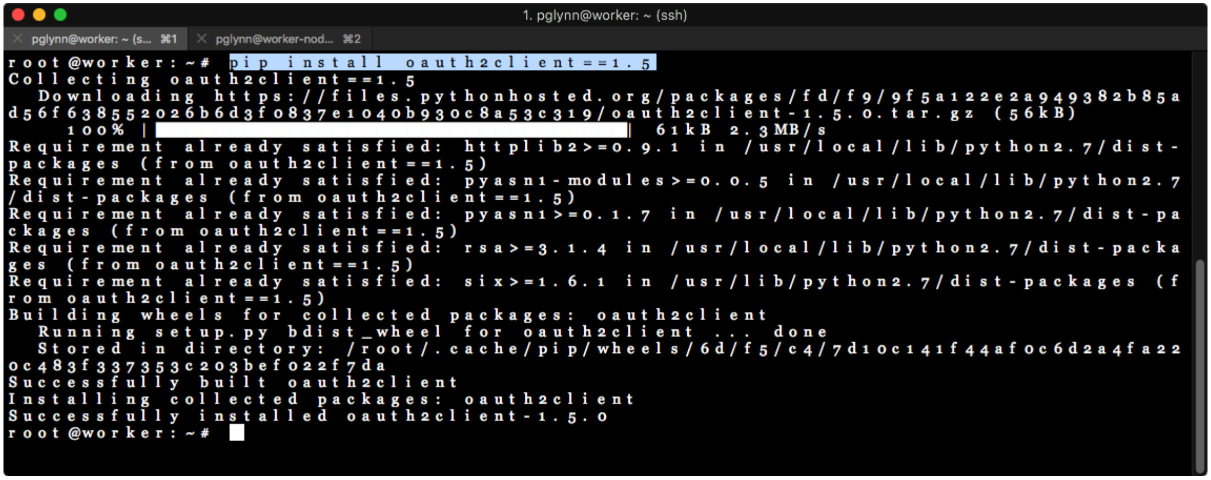
```
root@worker:~# apt-get install python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils build-essential bzip2 cpp cpp-6 dbus dpkg-dev fakeroot g++ g++-6
  gcc gcc-6 gir1.2-glib-2.0 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan3 libatomic1 libc-dev-bin libc6-dev libcc1-0
  libcilkrts5 libdbus-1-3 libdbus-glib-1-2 libdpkg-perl libexpat1-dev
  libfakeroot libfile-fcntllock-perl libgcc-6-dev libgirepository-1.0-1
  libglib2.0-0 libglib2.0-data libgomp1 libicu57 libisl15 libitm1 liblsan0
  libmpc3 libmpfr4 libmpx2 libperl5.24 libpython-all-dev libpython-dev
  libpython2.7 libpython2.7-dev libquadmath0 libstdc++-6-dev libtsan0
  libubsan0 libxml2 linux-libc-dev make manpages manpages-dev patch perl
  perl-modules-5.24 python-all python-all-dev python-cffi-backend
  python-crypto python-cryptography python-dbus python-dev python-enum34
  python-gi python-idna python-ipaddress python-keyring python-keyrings.alt
  python-pip-whl python-pyasn1 python-secretstorage python-setuptools
  python-wheel python-xdg python2.7-dev rename sgml-base shared-mime-info
  xdg-user-dirs xml-core
Suggested packages:
  binutils-doc bzip2-doc cpp-doc gcc-6-locales default-dbus-session-bus
  | dbus-session-bus debian-keyring g++-multilib g++-6-multilib gcc-6-doc
  libstdc++6-6-dbg gcc-multilib autoconf automake libtool flex bison gdb
  gcc-doc gcc-6-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
```

22) Install the Google API Client Library



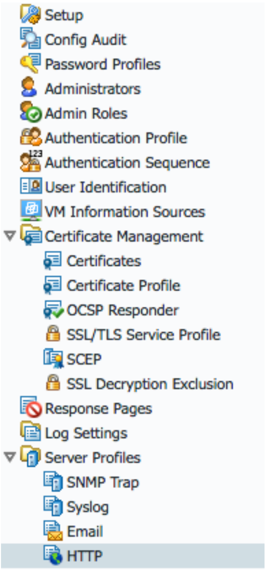
```
root@worker:~# pip install google-api-python-client
Collecting google-api-python-client
  Downloading https://files.pythonhosted.org/packages/bd/40/bc3b4e7c7c65f9548024
dde5c7bad60e0e078b2d2a0ee8c426a5639c2cc9/google_api_python_client-1.7.3-py2.py3-
none-any.whl (55kB)
    100% |████████████████████████████████| 61kB 2.4MB/s
Collecting google-auth>=1.4.1 (from google-api-python-client)
  Downloading https://files.pythonhosted.org/packages/53/06/6e6d5bfa4d23ee40efd7
72d6b681a7afecd859a9176e564b8c3293823 70f/google_auth-1.5.0-py2.py3-none-any.whl
(65kB)
    100% |████████████████████████████████| 71kB 4.2MB/s
Collecting google-auth-httplib2>=0.0.3 (from google-api-python-client)
  Downloading https://files.pythonhosted.org/packages/33/49/c814d6d438b823441552
198f096fcd0377fd6c88714dbed34f1d3c8c4389/google_auth_httplib2-0.0.3-py2.py3-none
-any.whl
Requirement already satisfied: six<2dev,>=1.6.1 in /usr/lib/python2.7/dist-packa
ges (from google-api-python-client)
Collecting uritemplate<4dev,>=3.0.0 (from google-api-python-client)
  Downloading https://files.pythonhosted.org/packages/f6/25/66a49231b44409d7f07c
fcf2506a8b070ce3c99fc47cc256bea833f24791/uritemplate-3.0.0-py2-none-any.whl
Collecting httplib2<1dev,>=0.9.2 (from google-api-python-client)
  Downloading https://files.pythonhosted.org/packages/fd/ce/aa4a385e3e9fd351737f
d2b07edaa56e7a730448465aceda6b35086a0d9b/httplib2-0.11.3.tar.gz (215kB)
    100% |████████████████████████████████| 225kB 3.1MB/s
Collecting pyasn1-modules>=0.2.1 (from google-auth>=1.4.1->google-api-python-cli
```

23) Install the OAuth client



```
root@worker:~# pip install oauth2client==1.5
Collecting oauth2client==1.5
  Downloading https://files.pythonhosted.org/packages/fd/f9/9f5a122e2a949382b85a
d56f63855 2026b6d3f0837e1040b930c8a53c319/oauth2client-1.5.0.tar.gz (56kB)
    100% |████████████████████████████████| 61kB 2.3MB/s
Requirement already satisfied: httplib2>=0.9.1 in /usr/local/lib/python2.7/dist-
packages (from oauth2client==1.5)
Requirement already satisfied: pyasn1-modules>=0.0.5 in /usr/local/lib/python2.7
/dist-packages (from oauth2client==1.5)
Requirement already satisfied: pyasn1>=0.1.7 in /usr/local/lib/python2.7/dist-pa
ckages (from oauth2client==1.5)
Requirement already satisfied: rsa>=3.1.4 in /usr/local/lib/python2.7/dist-packa
ges (from oauth2client==1.5)
Requirement already satisfied: six>=1.6.1 in /usr/lib/python2.7/dist-packages (f
rom oauth2client==1.5)
Building wheels for collected packages: oauth2client
  Running setup.py bdist_wheel for oauth2client ... done
  Stored in directory: /root/.cache/pip/wheels/6d/f5/c4/7d10c141f44af0c6d2a4fa22
0c483f337353c203bef022f7da
Successfully built oauth2client
Installing collected packages: oauth2client
Successfully installed oauth2client-1.5.0
root@worker:~# 
```

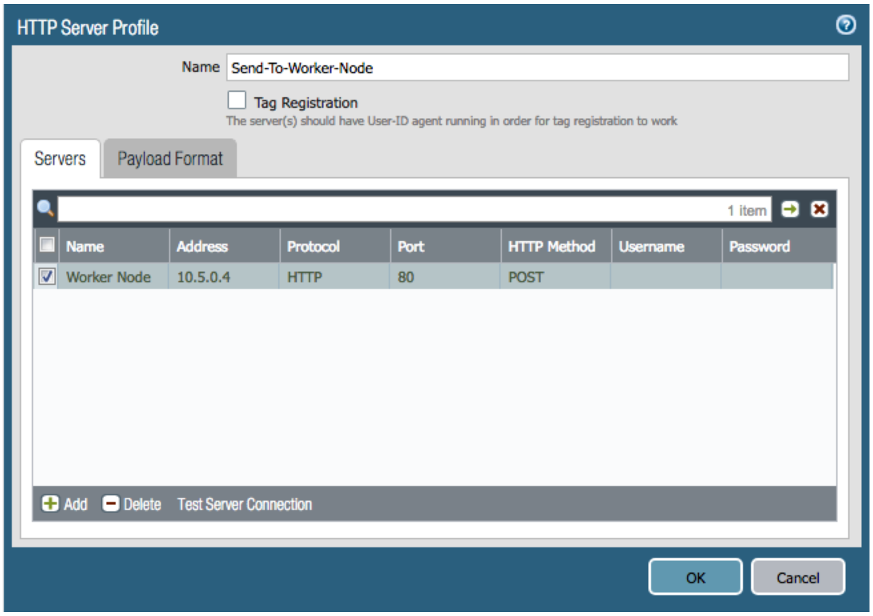
24) Login to the firewall and navigate to **Device > Server Profiles > HTTP**



25) Add a new server profile with the following parameters:

a. Name: free-form text (e.g. Worker Node)
b. Address: Internal IP of the worker node

c. Protocol: HTTP
d. Port 80
e. HTTP Method: POST



Name: Send-To-Worker-Node

□ Tag Registration

The server(s) should have User-ID agent running in order for tag registration to work

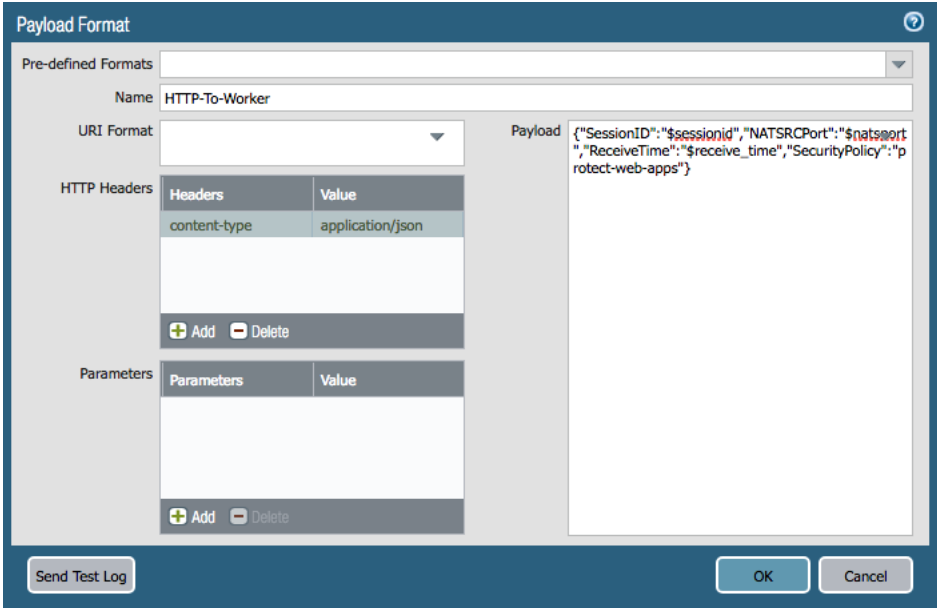
☐	Name	Address	Protocol	Port	HTTP Method	Username	Password
☑	Worker Node	10.5.0.4	HTTP	80	POST		

26) Under **Payload Format**, edit the log type for **Threat** and create a new payload format:

a. Name: Free-form text
b. Content-type: application/json
c. Payload:

{"SessionID":"$sessionid","NATSRCPort":"$natsport","ReceiveTime":"$receive_time","SecurityPolicy":"protect-web-apps"}

(replace "protect-web-apps" with the name of the security policy created earlier!)



Pre-defined Formats:

Name: HTTP-To-Worker

URI Format:

Payload: {"SessionID":"$sessionid","NATSRCPort":"$natsport","ReceiveTime":"$receive_time","SecurityPolicy":"protect-web-apps"}

HTTP Headers

Headers	Value
content-type	application/json

Parameters

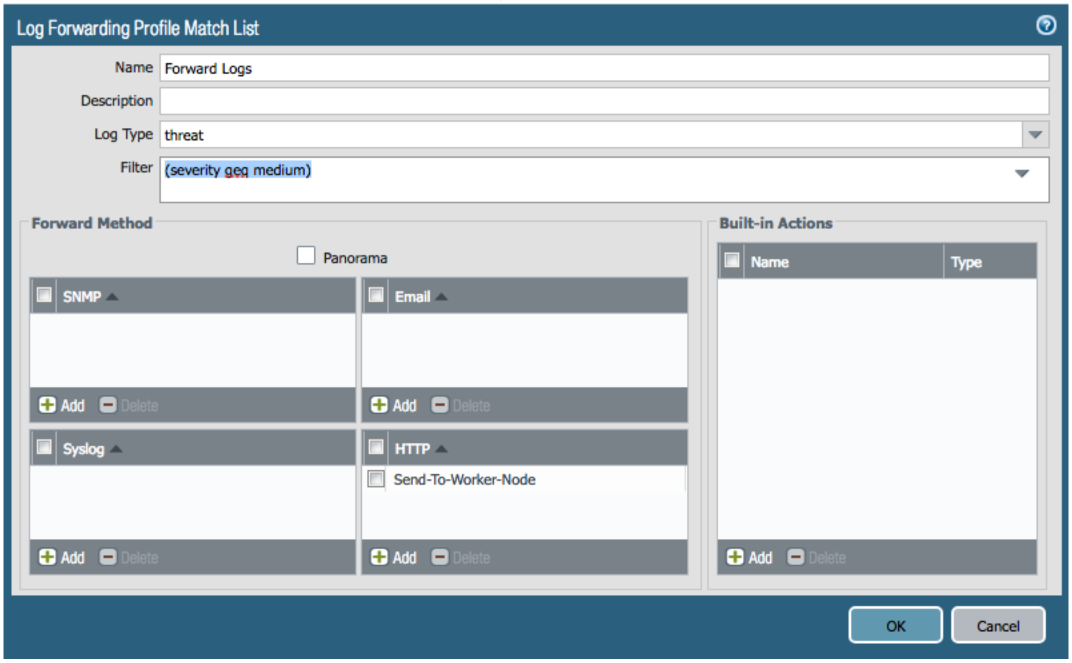
Parameters	Value

27) Navigate to **Objects > Log Forwarding** and add a new log forwarding profile with the following parameters:

a. Name: Free-form text
b. Log Type: Threat

c. Filter: (severity geq medium)

d. Forward Method: HTTP > (your HTTP Sever Profile)



Log Forwarding Profile Match List

Name: Forward Logs

Description:

Log Type: threat

Filter: (severity geq medium)

Forward Method

□ Panorama

SNMP

Email

Syslog

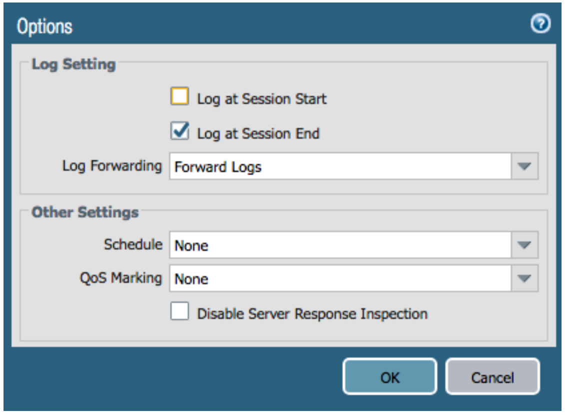
HTTP

□ Send-To-Worker-Node

Built-in Actions

Name	Type

28) Edit the policy permitting web traffic from the untrust/internet side of the FW and add the log forwarding profile to the **Options**



Options

Log Setting

□ Log at Session Start

☑ Log at Session End

Log Forwarding: Forward Logs

Other Settings

Schedule: None

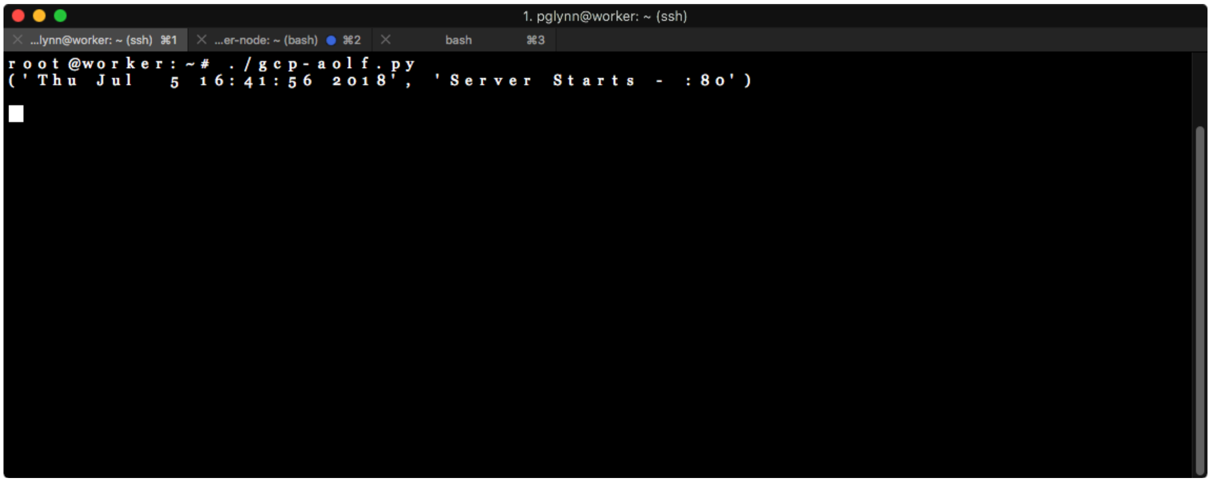
QoS Marking: None

□ Disable Server Response Inspection

29) Commit the changes and replicate as required to other FW

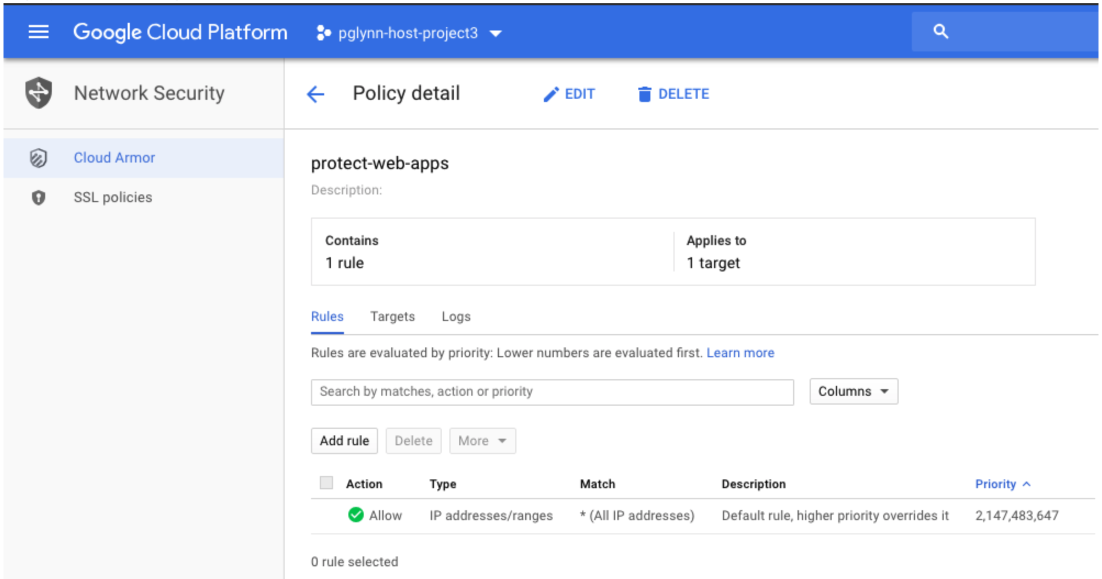
Testing/Verification

1) Launch the Python script from the worker node. You will have to be root if launching it manually as the script listens on a well-known port (TCP/80)



```
root@worker:~# ./gcp-aolf.py
('Thu Jul  5 16:41:56 2018', 'Server Starts - :80')
```

2) Check the Security Policy. In a production environment, there may be multiple rules blocking/permitting access.



protect-web-apps

Description:

Contains 1 rule

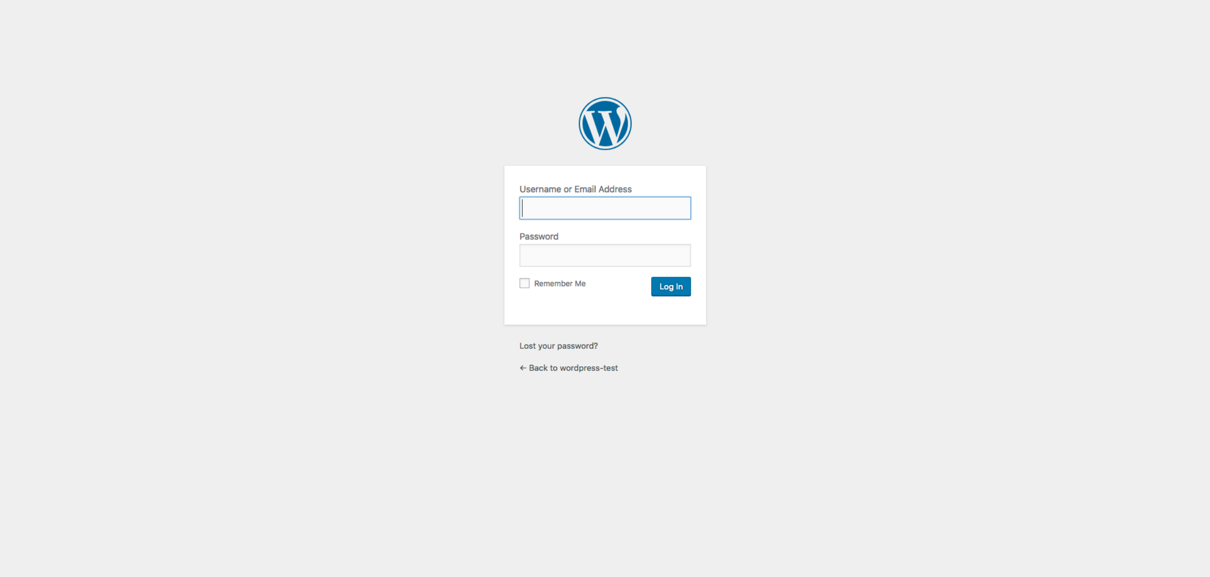
Applies to 1 target

Rules are evaluated by priority: Lower numbers are evaluated first. Learn more

Action	Type	Match	Description	Priority
Allow	IP addresses/ranges	* (All IP addresses)	Default rule, higher priority overrides it	2,147,483,647

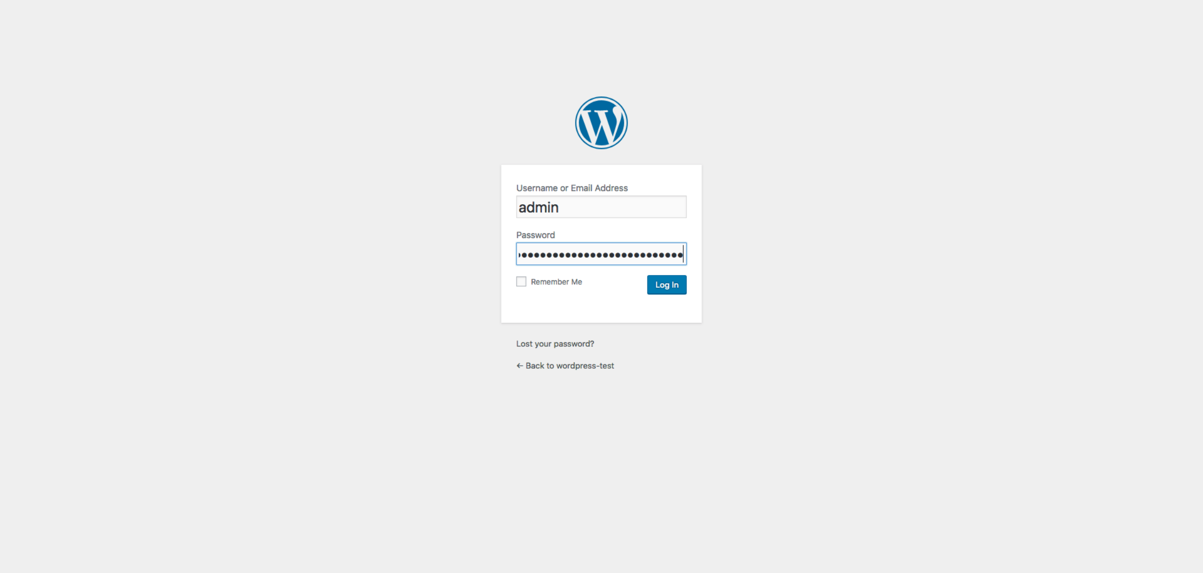
0 rule selected

3) Navigate to the web page (we are using a wordpress server for this example)



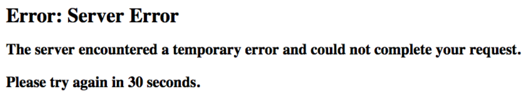
4) Input a valid username and for the password, simulate an XSS or SQL Injection attack. Examples include

```
<script>alert("HI")</script>
%' or '0'='0
1' or '1' = '1
```


Username or Email Address: admin

Password: ••••••••••••••••••••••••••••

5) The firewall should block the attempted login

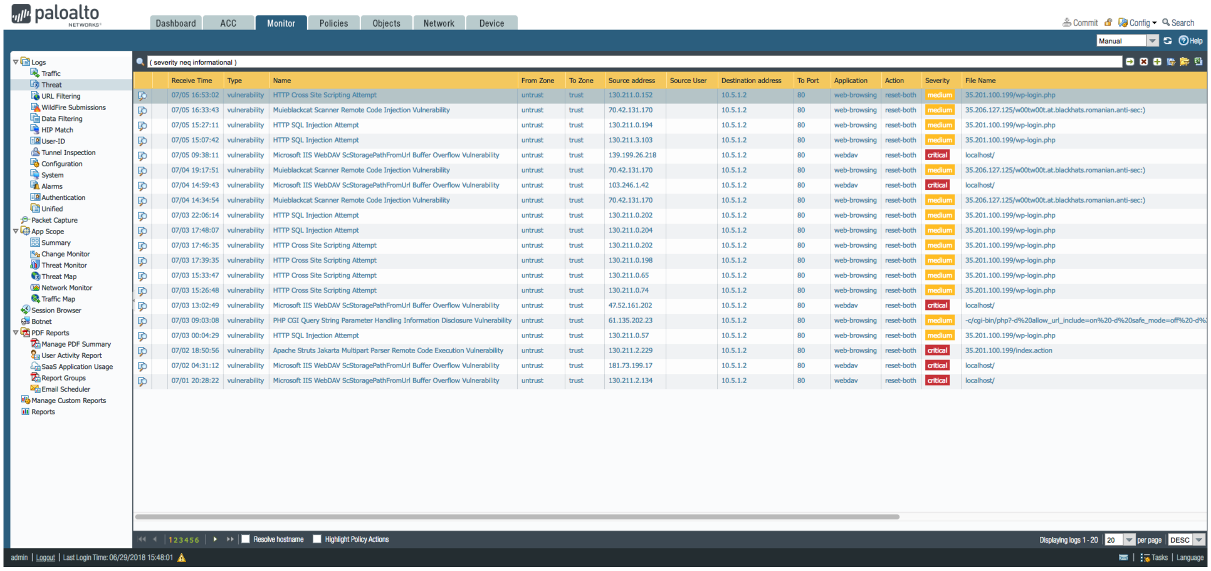


Error: Server Error

The server encountered a temporary error and could not complete your request.

Please try again in 30 seconds.

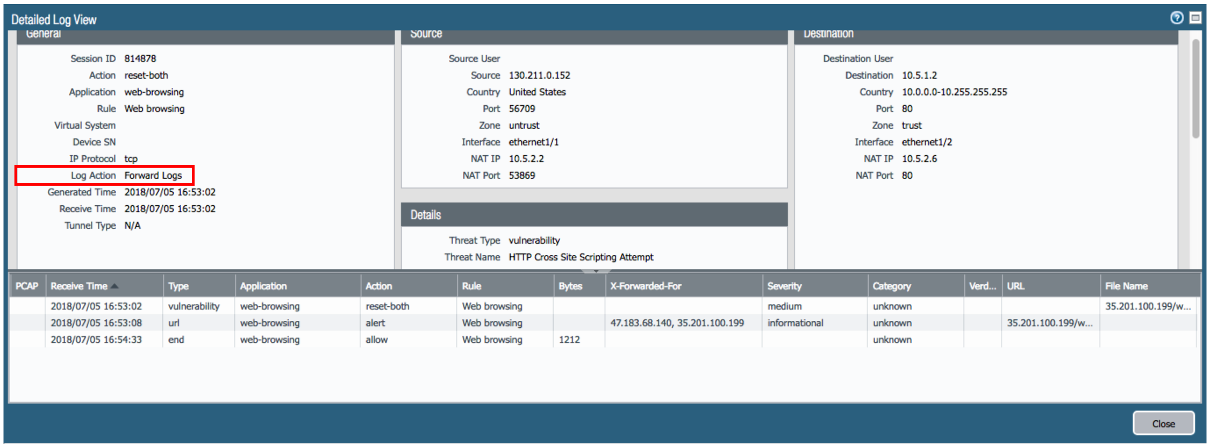
6) A Threat log event will be generated



(severity neq informational)

Receive Time	Type	Name	From Zone	To Zone	Source address	Source User	Destination address	To Port	Application	Action	Severity	File Name
07/05 16:53:02	vulnerability	HTTP Cross Site Scripting Attempt	untrust	trust	130.211.0.152		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/05 16:33:43	vulnerability	Muieblackcat Scanner Remote Code Injection Vulnerability	untrust	trust	70.42.131.170		10.5.1.2	80	web-browsing	reset-both	medium	35.206.127.125/w00tw00t.at.blackhats.romanian.anti-sec:)
07/05 15:27:11	vulnerability	HTTP SQL Injection Attempt	untrust	trust	130.211.0.194		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/05 15:07:42	vulnerability	HTTP SQL Injection Attempt	untrust	trust	130.211.3.103		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/05 09:38:11	vulnerability	Microsoft IIS WebDAV ScStoragePathFromUrl Buffer Overflow Vulnerability	untrust	trust	139.199.26.218		10.5.1.2	80	webdav	reset-both	critical	localhost/
07/04 19:17:51	vulnerability	Muieblackcat Scanner Remote Code Injection Vulnerability	untrust	trust	70.42.131.170		10.5.1.2	80	web-browsing	reset-both	medium	35.206.127.125/w00tw00t.at.blackhats.romanian.anti-sec:)
07/04 14:59:43	vulnerability	Microsoft IIS WebDAV ScStoragePathFromUrl Buffer Overflow Vulnerability	untrust	trust	103.246.1.42		10.5.1.2	80	webdav	reset-both	critical	localhost/
07/04 14:34:54	vulnerability	Muieblackcat Scanner Remote Code Injection Vulnerability	untrust	trust	70.42.131.170		10.5.1.2	80	web-browsing	reset-both	medium	35.206.127.125/w00tw00t.at.blackhats.romanian.anti-sec:)
07/03 22:06:14	vulnerability	HTTP SQL Injection Attempt	untrust	trust	130.211.0.202		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/03 17:48:07	vulnerability	HTTP SQL Injection Attempt	untrust	trust	130.211.0.204		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/03 17:46:35	vulnerability	HTTP Cross Site Scripting Attempt	untrust	trust	130.211.0.202		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/03 17:39:35	vulnerability	HTTP Cross Site Scripting Attempt	untrust	trust	130.211.0.198		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/03 15:33:47	vulnerability	HTTP Cross Site Scripting Attempt	untrust	trust	130.211.0.65		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/03 15:26:48	vulnerability	HTTP Cross Site Scripting Attempt	untrust	trust	130.211.0.74		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/03 13:02:49	vulnerability	Microsoft IIS WebDAV ScStoragePathFromUrl Buffer Overflow Vulnerability	untrust	trust	47.52.161.202		10.5.1.2	80	webdav	reset-both	critical	localhost/
07/03 09:03:08	vulnerability	PHP CGI Query String Parameter Handling Information Disclosure Vulnerability	untrust	trust	61.135.202.23		10.5.1.2	80	web-browsing	reset-both	medium	-c/cgi-bin/php?-d%20allow_url_include=on%20-d%20safe_mode=off%20-d%
07/03 00:04:29	vulnerability	HTTP SQL Injection Attempt	untrust	trust	130.211.0.57		10.5.1.2	80	web-browsing	reset-both	medium	35.201.100.199/wp-login.php
07/02 18:50:56	vulnerability	Apache Struts Jakarta Multipart Parser Remote Code Execution Vulnerability	untrust	trust	130.211.2.229		10.5.1.2	80	web-browsing	reset-both	critical	35.201.100.199/index.action
07/02 04:31:12	vulnerability	Microsoft IIS WebDAV ScStoragePathFromUrl Buffer Overflow Vulnerability	untrust	trust	181.73.199.17		10.5.1.2	80	webdav	reset-both	critical	localhost/
07/01 20:28:22	vulnerability	Microsoft IIS WebDAV ScStoragePathFromUrl Buffer Overflow Vulnerability	untrust	trust	130.211.2.134		10.5.1.2	80	webdav	reset-both	critical	localhost/

7) Details on the log entry will show the Traffic, URL, and Threat logs as well as the log forwarding action



General

Session ID 814878
Action reset-both
Application web-browsing
Rule Web browsing
Virtual System
Device SN
IP Protocol tcp
Log Action Forward Logs
Generated Time 2018/07/05 16:53:02
Receive Time 2018/07/05 16:53:02
Tunnel Type N/A

Source

Source User
Source 130.211.0.152
Country United States
Port 56709
Zone untrust
Interface ethernet1/1
NAT IP 10.5.2.2
NAT Port 53869

Destination

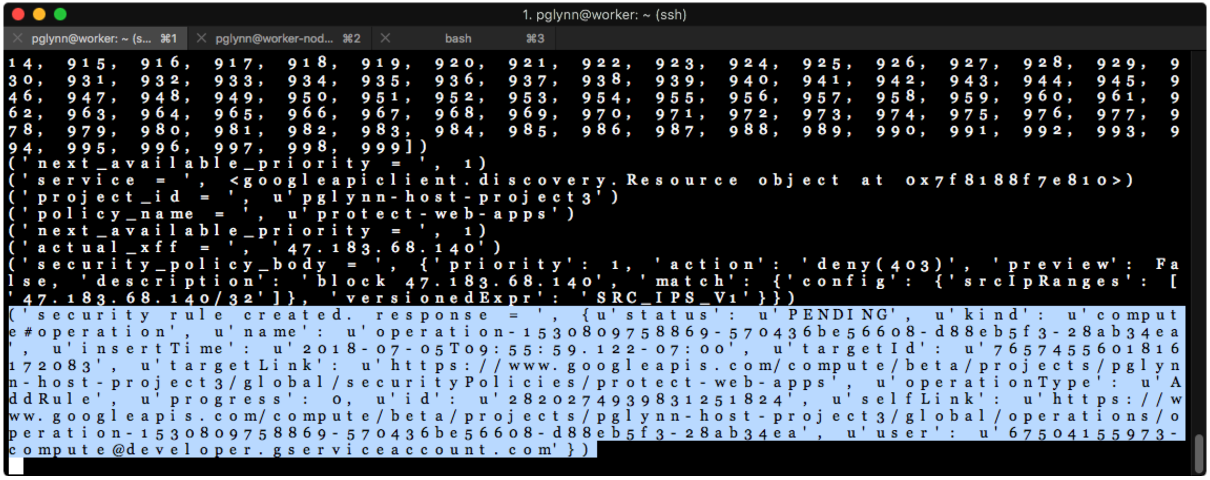
Destination User
Destination 10.5.1.2
Country 10.0.0.0-10.255.255.255
Port 80
Zone trust
Interface ethernet1/2
NAT IP 10.5.2.6
NAT Port 80

Details

Threat Type vulnerability
Threat Name HTTP Cross Site Scripting Attempt

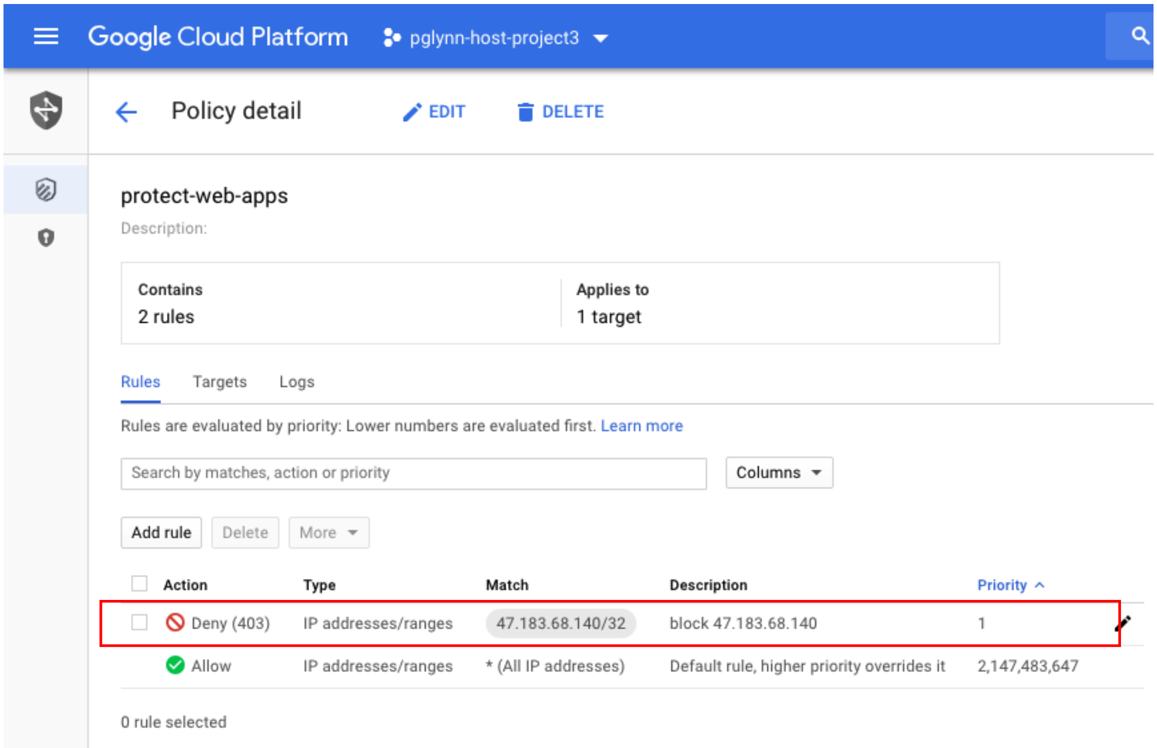
PCAP	Receive Time	Type	Application	Action	Rule	Bytes	X-Forwarded-For	Severity	Category	Verd...	URL	File Name
	2018/07/05 16:53:02	vulnerability	web-browsing	reset-both	Web browsing			medium	unknown			35.201.100.199/w...
	2018/07/05 16:53:08	url	web-browsing	alert	Web browsing		47.183.68.140, 35.201.100.199	informational	unknown		35.201.100.199/w...	
	2018/07/05 16:54:33	end	web-browsing	allow	Web browsing	1212			unknown			

8) If debugging is enabled, you will see a large amount of output culminating in the acceptance of the request to create a new rule



```
14, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 9
30, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 9
46, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 9
62, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 9
78, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 9
94, 995, 996, 997, 998, 999])
('next_available_priority = ', 1)
('service = ', <googleapiclient.discovery.Resource object at 0x7f8188f7e810>)
('project_id = ', u'pglynn-host-project3')
('policy_name = ', u'protect-web-apps')
('next_available_priority = ', 1)
('actual_xff = ', '47.183.68.140')
('security_policy_body = ', {'priority': 1, 'action': 'deny(403)', 'preview': Fa
lse, 'description': 'block 47.183.68.140', 'match': {'config': {'srcIpRanges': [
'47.183.68.140/32']}, 'versionedExpr': 'SRC_IPS_V1'}})
('security rule created. response = ', {u'status': u'PENDING', u'kind': u'comput
e#operation', u'name': u'operation-1530809758869-570436be56608-d88eb5f3-28ab34ea
', u'insertTime': u'2018-07-05T09:55:59.122-07:00', u'targetId': u'7657455601816
172083', u'targetLink': u'https://www.googleapis.com/compute/beta/projects/pglyn
n-host-project3/global/securityPolicies/protect-web-apps', u'operationType': u'A
ddRule', u'progress': 0, u'id': u'2820274939831251824', u'selfLink': u'https://w
ww.googleapis.com/compute/beta/projects/pglynn-host-project3/global/operations/o
peration-1530809758869-570436be56608-d88eb5f3-28ab34ea', u'user': u'67504155973-
compute@developer.gserviceaccount.com'})
```

9) Check the security policy after a few moments (a browser refresh may be required)



Google Cloud Platform · pglynn-host-project3

Policy detail · EDIT · DELETE

protect-web-apps

Description:

Contains	Applies to
2 rules	1 target

Rules · Targets · Logs

Rules are evaluated by priority: Lower numbers are evaluated first. Learn more

Search by matches, action or priority · Columns

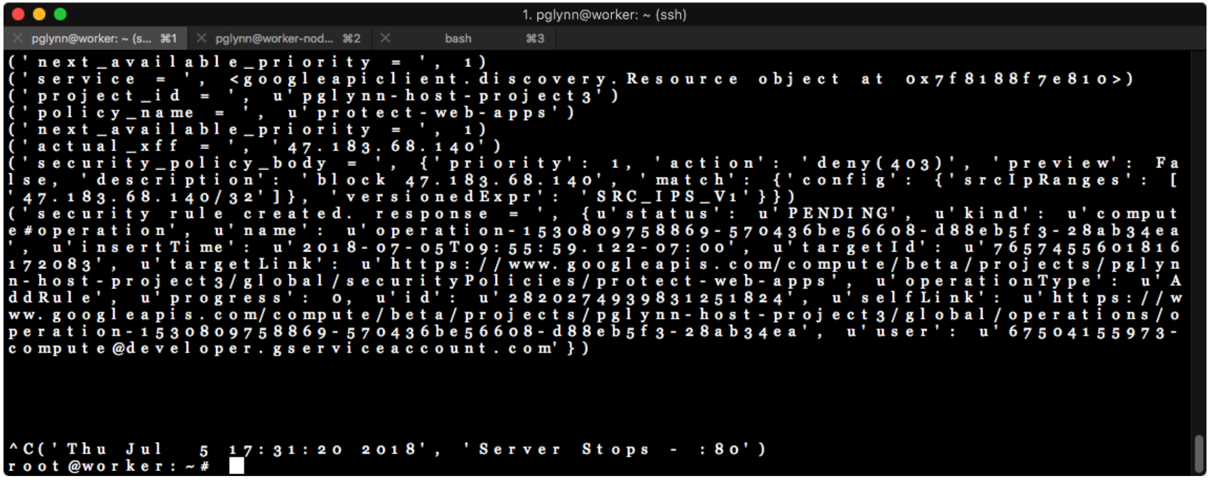
Add rule · Delete · More

	Action	Type	Match	Description	Priority
□	Deny (403)	IP addresses/ranges	47.183.68.140/32	block 47.183.68.140	1
	Allow	IP addresses/ranges	* (All IP addresses)	Default rule, higher priority overrides it	2,147,483,647

0 rule selected

10) Verify by re-attempting the XSS/SQL Injection attempt from the browser. You should see a **403 Forbidden** error

11) Interrupt the Python script with <CRTL>-<C>



1. pglynn@worker: ~ (ssh)

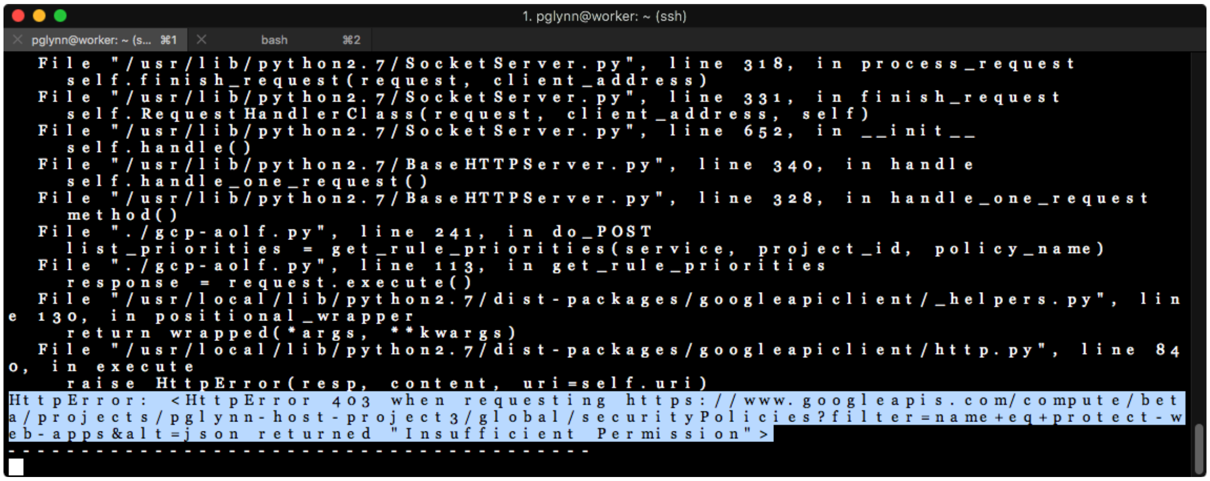
pglynn@worker: ~ (s... ⌘1 · pglynn@worker-nod... ⌘2 · bash ⌘3

```
('next_available_priority = ', 1)
('service = ', <googleapiclient.discovery.Resource object at 0x7f8188f7e810>)
('project_id = ', u'pglynn-host-project3')
('policy_name = ', u'protect-web-apps')
('next_available_priority = ', 1)
('actual_xff = ', '47.183.68.140')
('security_policy_body = ', {'priority': 1, 'action': 'deny(403)', 'preview': Fa
lse, 'description': 'block 47.183.68.140', 'match': {'config': {'srcIpRanges': [
'47.183.68.140/32']}, 'versionedExpr': 'SRC_IPS_V1'}})
('security rule created. response = ', {u'status': u'PENDING', u'kind': u'comput
e#operation', u'name': u'operation-1530809758869-570436be56608-d88eb5f3-28ab34ea
', u'insertTime': u'2018-07-05T09:55:59.122-07:00', u'targetId': u'7657455601816
172083', u'targetLink': u'https://www.googleapis.com/compute/beta/projects/pglyn
n-host-project3/global/securityPolicies/protect-web-apps', u'operationType': u'A
ddRule', u'progress': 0, u'id': u'2820274939831251824', u'selfLink': u'https://w
ww.googleapis.com/compute/beta/projects/pglynn-host-project3/global/operations/o
peration-1530809758869-570436be56608-d88eb5f3-28ab34ea', u'user': u'67504155973-
compute@developer.gserviceaccount.com'})



^C('Thu Jul  5 17:31:20 2018', 'Server Stops - :80')
root@worker:~# 
```

Troubleshooting

- For the script to be able to execute, it needs to load two python libraries: google-api-python3-client and oauth2client==1.5. If those two libraries are not installed prior to running the script, it will exit with an error
- If the service account does not have the correct permissions or the authentication key has not been loaded, the script will run but fail when attempting to query the GCP environment. For more details on the API calls, including required IAM permissions, see

https://cloud.google.com/compute/docs/reference/rest/beta/securityPolicies/list
https://cloud.google.com/compute/docs/reference/rest/beta/securityPolicies/addRule



```
  File "/usr/lib/python2.7/SocketServer.py", line 318, in process_request
    self.finish_request(request, client_address)
  File "/usr/lib/python2.7/SocketServer.py", line 331, in finish_request
    self.RequestHandlerClass(request, client_address, self)
  File "/usr/lib/python2.7/SocketServer.py", line 652, in __init__
    self.handle()
  File "/usr/lib/python2.7/BaseHTTPServer.py", line 340, in handle
    self.handle_one_request()
  File "/usr/lib/python2.7/BaseHTTPServer.py", line 328, in handle_one_request
    method()
  File "./gcp-aolf.py", line 241, in do_POST
    list_priorities = get_rule_priorities(service, project_id, policy_name)
  File "./gcp-aolf.py", line 113, in get_rule_priorities
    response = request.execute()
  File "/usr/local/lib/python2.7/dist-packages/googleapiclient/_helpers.py", lin
e 130, in positional_wrapper
    return wrapped(*args, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/googleapiclient/http.py", line 84
0, in execute
    raise HttpError(resp, content, uri=self.uri)
HttpError: <HttpError 403 when requesting https://www.googleapis.com/compute/bet
a/projects/pglynn-host-project3/global/securityPolicies?filter=name+eq+protect-w
eb-apps&alt=json returned "Insufficient Permission">
----------------------------------------
```