

HoneyPot with Google Cloud & T-pot

Create a VM on Google Cloud

1. Create a new project on Google Cloud.
2. Compute Engine -> VM Instances -> Create Instance -> Configure VM as below

Google Cloud Platform

HP server

Search products and resources

Create an instance

To create a VM instance, select one of the options:

New VM instance

Create a single VM instance from scratch

New VM instance from template

Create a single VM instance from an existing template

New VM instance from machine image

Create a single VM instance from an existing machine image

Marketplace

Deploy a ready-to-go solution onto a VM instance

Machine family

GENERAL-PURPOSE

COMPUTE-OPTIMIZED

MEMORY-OPTIMIZED

GPU

Machine types for common workloads, optimized for cost and flexibility

Series

N2

Powered by Intel Cascade Lake and Ice Lake CPU platforms

Machine type

n2-standard-4 (4 vCPU, 16 GB memory)

vCPU

4

Memory

16 GB

CPU PLATFORM AND GPU

Display device

Enable to use screen capturing and recording tools.

☐ Enable display device

Confidential VM service

☐ Enable the Confidential Computing service on this VM instance.

Container

Deploy a container image to this VM instance

DEPLOY CONTAINER

Boot disk

Type	New balanced persistent disk
Size	20 GB
Image	Debian GNU/Linux 10 (buster)

CHANGE

Create a vulnerable firewall

- Press navigation menu -> VPC Network -> Firewall -> Configure firewall rules as below

Create a vulnerable firewall

- Press navigation menu -> VPC Network -> Firewall -> Configure firewall rules as below

VPC network

VPC networks

External IP addresses

Bring your own IP

Firewall

Routes

VPC network peering

Shared VPC

Serverless VPC access

Packet mirroring


Firewall

CREATE FIREWALL RULE

REFRESH

CONFIGURE LOGS

DELETE



Get real-time analytics with Network Intelligence Center

Use Network Intelligence Center for comprehensive monitoring and troubleshooting. [Learn more](#)

Visualize your network resources

Diagnose and prevent connectivity issues

View packet loss and latency metrics

Keep your firewall rules strict and efficient

GO TO NETWORK INTELLIGENCE CENTER

REMIND ME LATER

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Note: App Engine firewalls are managed in the [App Engine Firewall rules section](#).

Filter Enter property name or value

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑	Logs
<input type="checkbox"/>	allow-other	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:0-64294 udp icmp	Allow	1000	default	Off
<input type="checkbox"/>	allow-other-higher	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:64298-65535 udp icmp	Allow	1000	default	Off
<input type="checkbox"/>	default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	all	Allow	1000	default	Off
<input type="checkbox"/>	default-allow-https	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:64297	Allow	1000	default	Off
<input type="checkbox"/>	default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default	Off
<input type="checkbox"/>	default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default	Off
<input type="checkbox"/>	default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default	Off
<input type="checkbox"/>	default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:64295	Allow	65534	default	Off

SSH to VM

- Return to Compute Engine -> VM Instances -> Connect, SSH

SSH to VM

- Return to Compute Engine -> VM Instances -> Connect, SSH

Google Cloud Platform

HP server

Search products and resources

Compute Engine

VM instances

CREATE INSTANCE

IMPORT VM

REFRESH

START / RESUME

STOP

SUSPEND

OPERATIONS

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

TPUs

Committed use discounts

Migrate for Compute Engi...

Storage

Disks

INSTANCES

INSTANCE SCHEDULE

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. [Learn more](#)

Filter

Enter property name or value

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>		honeypot	us-west4-b			10.182.0.2 (nic0)	34.125.211.8	SSH

Related actions

DISMISS

View billing report

View and manage your Compute Engine billing

Monitor VMs

View outlier VMs across metrics like CPU and network

Explore VM logs

View, search, analyze, and download VM instance logs

Set up firewall rules

Control traffic to and from a VM instance

Patch management

Schedule patch updates and view patch compliance on VM instances

Commands

5. `sudo su #become root`
6. `apt-get update && apt-get upgrade #update and upgrade existing packages`
7. `apt-get install git -y #install git and say yes to all`
8. `git clone https://github.com/telekom-security/tpotce.git`
9. `cd tpotce #change directory to tpotce`
10. `./install.sh --type=user #execute bash script as user`



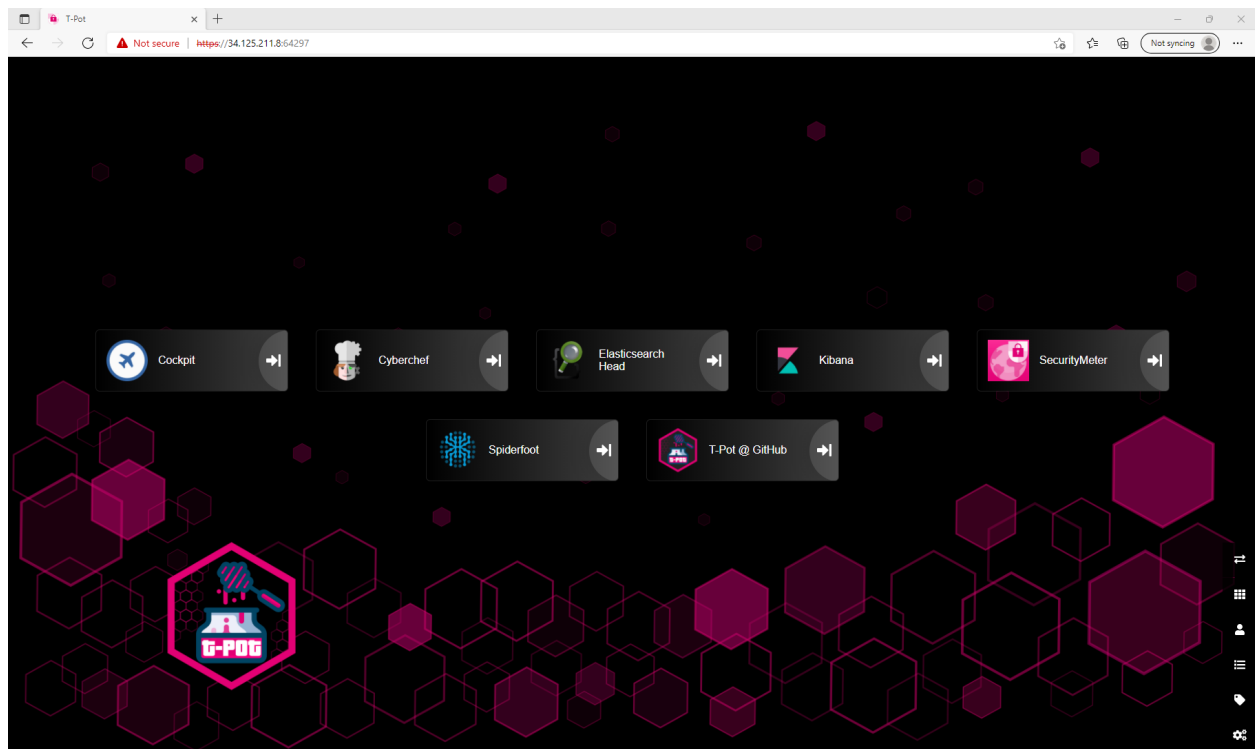
11. Select standard -> Choose username and password

Monitor

12. Return to Compute Engine -> VM Instances -> Copy your external ip address
13. Open up a browser -> Enter `https://<external ip>:64297`
14. Enter the username and password you have set for the user

Landing Page

(please note that it takes about 5 mins for Kibana and Elasticsearch to initialize)

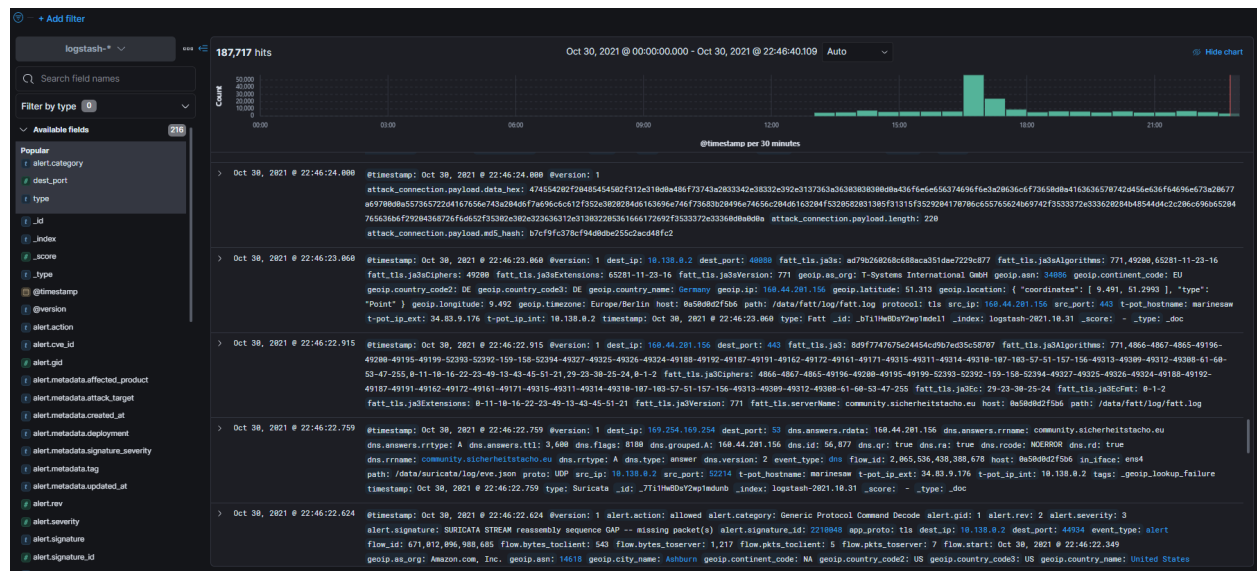


Kibana after 5 mins

Attacks are happening already.



Over 10,000 attacks were recorded.

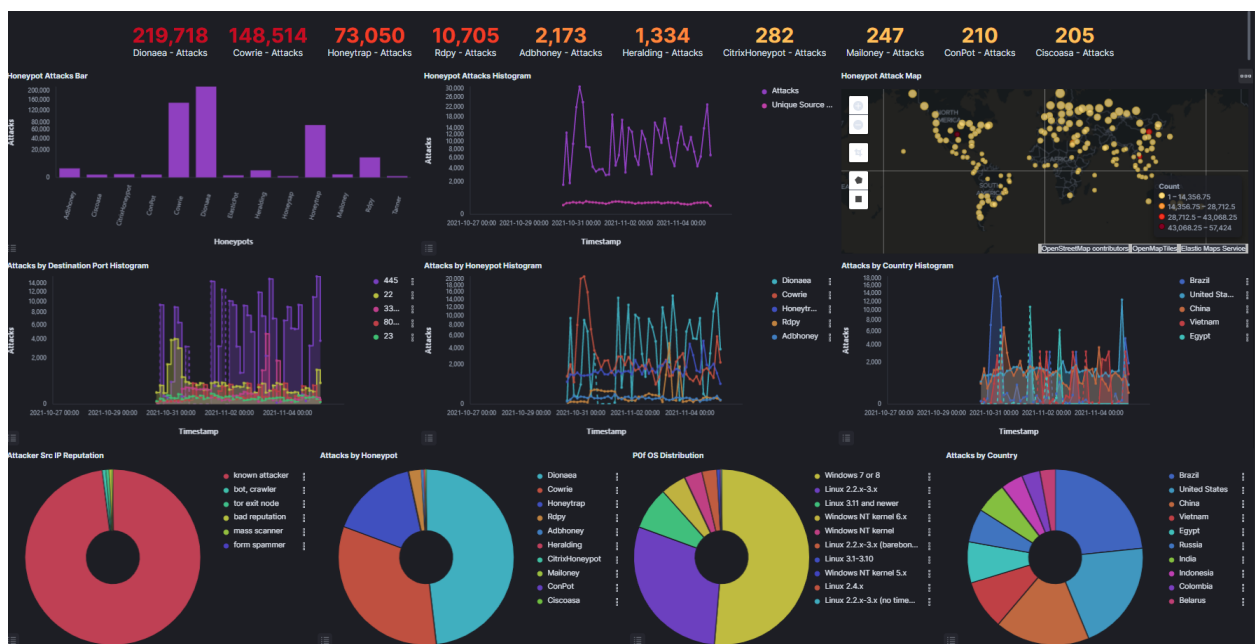


Thoughts

Considering the fact that I was attacked within the first five minutes of deployment, hackers do scan addresses for vulnerable ports on a daily basis, even minutely basis. Honeypot is a powerful tool for companies to learn what type of attacks hackers are performing these days. Some information will need to be taken with a grain of salt such as location and IP addresses. But most importantly, the logs generated from this can be used to do further analysis in preventing future attacks.

After Thoughts

I came back to see how many attacks were performed ten days later.

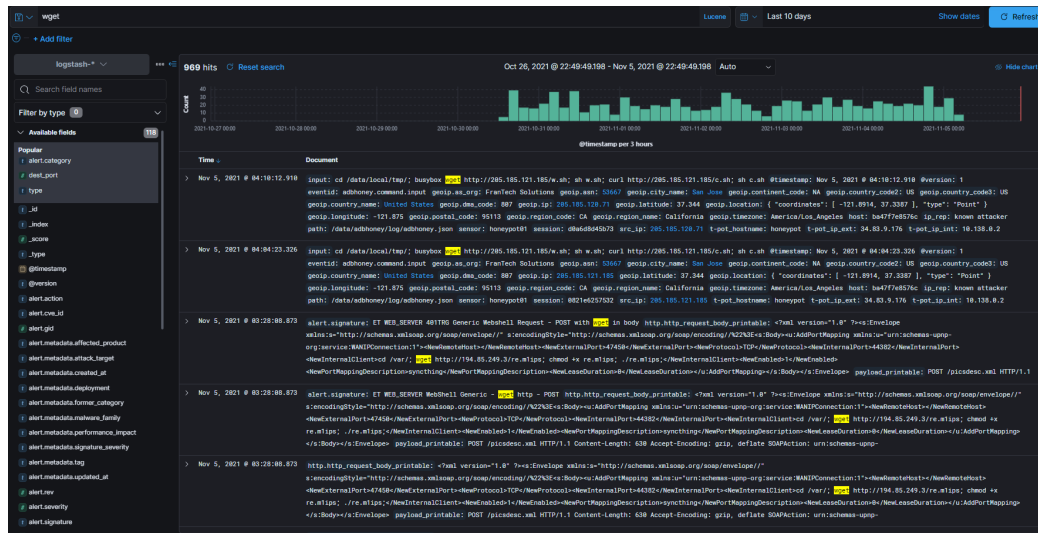


Here are some of the interesting things that I found:

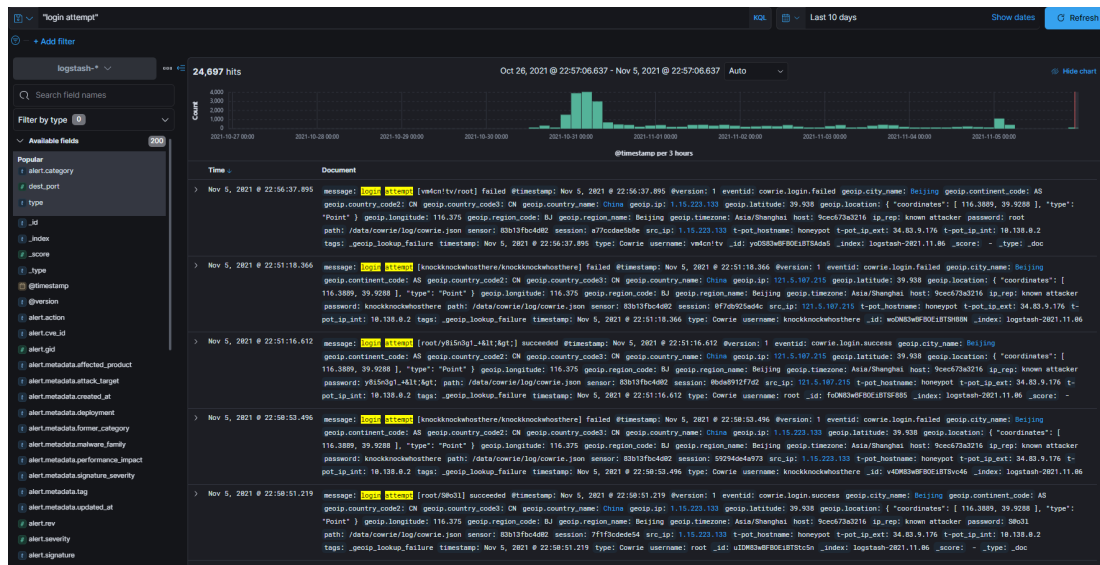
1. Someone downloaded different versions of malware named dota*.tar.gz

Filename	T-Pot Path (/data/cowrie/downloads)	Count
dota3.tar.gz	d1/a67ee2d7d0517224b8c12406d597e4b63041ab75ea519b8...	2
dota3.tar.gz	d1/0d27d54ece114e7dcc27ef80482d45f1e8929ce05245e576...	1
dota3.tar.gz	d1/ac00871ebbbb76ca1591cb175521b634ca1ced9d132525c...	1
dota3.tar.gz	d1/cbcb2331843b4369e36c4a38013c6cac234ddfd0f097aa5...	1
dota3.tar.gz	d1/e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca...	1
dota.tar.gz	d1/4a61b67a96a8943a5c114ea306dbec15d057446720a5d0...	1
dota.tar.gz	d1/5fd40131a6647d1e627787796a164f38609f06df69c13ee...	1
dota.tar.gz	d1/86cac62d8ac5c8d26069b0d37bb569aa1d8e0c2e2a4166bf...	1
dota.tar.gz	d1/f2fb42f593c652f621fd39ef5364a128921c8646912f218...	1

2. There have been over 900+ attempts to retrieve files via HTTP/HTTPS using wget



3. Over 24000+ failed login attempts



4. Username and password tagcloud along with top ten attack source/IP

