SECTRIO

MALWARE REPORT

XorDDoS

Date: 14/08/2020

Shikha Sangwan

XorDDoS is known for years now, for compromising linux systems and launching DDoS attacks. First it attacks the target host by brute-forcing against weak secure shell services password, then it acquires root privileges to execute shell scripts which infects the host system by installing malicious binaries on the system. It uses XOR encryption in the communication with C&Cs. It is built for different architectures including ARM and Intel.

OVERVIEW

- Subex Threat Intelligence detected XorDDoS attack on the HoneyPot network on 08 August 2020.
- DDoS attacks are amplifying in the IoT industry and XorDDoS is one of the main malware family which is used by attackers to launch large scale DDoS attacks and for hijacking linux machines.
- XorDDoS malware is spreading now a days with changed variants.
- The sample that we got is an ELF32 bit GNU/Linux statically compiled binary having Intel architecture. It is a linux trojan malware with bot characteristics.
- It shows characteristics of a bot and an evader too, it doesn't show rootkit characteristics.

THREAT ASSESSMENT

It starts and executes the 'sleep' command which is used to delay the execution process and to evade sandboxes. Then it deletes itself in order to hide and protect, and copies the original file in the '/tmp' directory before deleting, the name with which it creates the file is = "**bVexvNSHcD**" with a process ID = 1308. It generates the file with different names every time we execute it.

The execution continues by dropping more than 12 malicious files in the '/usr/bin' folder, so that it becomes difficult to locate the files. The files dropped will be in suspicious directories and will have random names. Due to this, so many processes will get started.

The files dropped:

/usr/bin/cccqzynply
/usr/bin/eymbnveibk
/usr/bin/rzrdxbhbdw
/usr/bin/sdpqjyakbc
/usr/bin/trmivoqxyl
/usr/bin/naukgaquuk
/usr/bin/wyslzclppe
/lib/libudev.so
/usr/bin/lywqbfmulc
/usr/bin/vvivzlfebp
/usr/bin/oaazjlzsqi
/usr/bin/mmvptggmnc
/usr/bin/cccqzynply

These files are malicious ELF32 bit files with proper ELF headers that are dropped in the suspicious directories. They start performing malicious activities and start reading the system information and then delete themselves, so that no signs or trace will be left in the system.

The files opened by **rzrdxbhbdw** :

/proc/rs_dev
/usr/bin/rzrdxbhbdw

Following are the MD5 hashes of the files:

- 9B5FC08231B221BF9619E9EEF8365157
- 70949E5B78335161666DEB0229AA80EC
- E800AAA3AFA5F4FE0BE87A542EC27D35
- C4D728FC3B7D65F56C40C655BD4811E3
- D276AB5EF6E3383E477BF1EFE2F44155
- 0B42DFB3C932D27E7A24A1B1B2AF78FA
- 8199DF8378FAB165B5DABA2CD508A81D
- 07C1CE8F76388A062B6683A7C6C52F60
- 3FE8B88003770F4049335ADDB90B41B5
- 6225F2DE266811D1754A02B115FB51EE
- C456F893282E958407667E2E58D88EAC
- 317F994AA7BE61A911582C4EF2AE32EC

The sample uses cron to persist itself (refer to the code shown below) in the host. It creates a script in the cron.hourly directory. It serves as a cron job that will be started every hour. The line "`*/3 * * * * root /etc/cron.hourly/cron.sh`" is inserted in the crontab.

To achieve the persistence behaviour and to continue the operation, it creates scripts in '**/etc/rc[1-5].d**' folder, named "S01bVexvN" , which serves as the auto start folder in linux systems.

It drops ascii text files depend.stop, depend.start and depend.boot in the '**/etc/init.d**' directory. Linux services can be started, stopped and reloaded with the use of scripts stocked in /etc/init.d/. 'update-rc.d' updates the System V style init script links /etc/rc[runlevel].d/NNname whose target is the script /etc/init.d/name. This makes the virus to autostart whenever we start the system.

The sample sets the environment variable path by as follows:

PATH:
/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin

```
.text:0804CFBD                mov     dword ptr [esp], 0Dh
.text:0804CFC4                call    ssignal
.text:0804CFC9                mov     dword ptr [esp+4], 1
.text:0804CFD1                mov     dword ptr [esp], 11h
.text:0804CFD8                call    ssignal
.text:0804CFDD                mov     dword ptr [esp+8], 1
.text:0804CFE5                mov     dword ptr [esp+4], offset aBinSbinUsrBinU ; "/bin:/sbin:/usr/bin:/usr/sbin:/usr/loca"...
.text:0804CFED                mov     dword ptr [esp], offset aPath ; "PATH"
.text:0804CFF4                call    setenv
.text:0804CFF9                mov     dword ptr [esp+4], 400h ; int
.text:0804D001                lea     eax, [ebp+filename]
.text:0804D007                mov     [esp], eax      ; buf
.text:0804D00A                call    get_self
```

It connects to the network by the following IP addresses through mysql port (3306):

45.104.44.204

204.44.104.45

98.159.110.21

It connects to different IPs every time we run the sample. In the network traffic, not much data was seen in the communication, due to unavailability of the host. Traces of network connection can be seen in the code of the malware too.

```
.rodata:080B2E04 aAcceptAcceptLa db 'Accept: */*',0Dh,0Ah
.rodata:080B2E04                                        ; DATA XREF: .data:http_data_m↓o
.rodata:080B2E04                 db 'Accept-Language: zh-cn',0Dh,0Ah
.rodata:080B2E04                 db 'User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV'
.rodata:080B2E04                 db '1; TencentTraveler ; .NET CLR 1.1.4322)',0Dh,0Ah,0
.rodata:080B2E94 aConnectionKeep db 'Connection: Keep-Alive',0Dh,0Ah
.rodata:080B2E94                                        ; DATA XREF: .data:http_data_e↓o
.rodata:080B2E94                 db 0Dh,0Ah,0
.rodata:080B2EAF aHttp           db 'http://',0         ; DATA XREF: abstract_url+23↑o
.rodata:080B2EB7                 align 4
.rodata:080B2EB8 aPostSHttp11Sho db 'POST %s HTTP/1.1',0Dh,0Ah
.rodata:080B2EB8                                        ; DATA XREF: http_post+F5↑o
.rodata:080B2EB8                 db '%sHost: %s',0Dh,0Ah
.rodata:080B2EB8                 db 'Content-Type: application/x-www-form-urlencoded',0Dh,0Ah
.rodata:080B2EB8                 db 'Content-Length: %d',0Dh,0Ah
.rodata:080B2EB8                 db '%s%s',0
.rodata:080B2F20 aGetSHttp11Shos db 'GET %s HTTP/1.1',0Dh,0Ah
.rodata:080B2F20                                        ; DATA XREF: http_download_mem+124↑o
.rodata:080B2F20                                        ; http_download+263↑o
.rodata:080B2F20                 db 'Host: %s',0Dh,0Ah
```

When we see the code, the sample collects system information like PID of the process that possess connection, cpu information, memory information, checksum of file content and process path and generates process lists and sends as data to the C&C server.

```
...b::000078bd 63 68 65        ds          "checksum"
               63 6b 73
               75 6d 00
```

```
...b::000039c9 67 65 74        ds          "getpid"
               70 69 64 00
```

```
                    s_/proc/meminfo_080b32b6                XREF[1]:     GetMemStat:0804e20d(*)
080b32b6 2f 70 72       ds          "/proc/meminfo"
         6f 63 2f
         6d 65 6d ...

                    s_/proc/cpuinfo_080b32c4                XREF[1]:     GetCpuInfo:0804e30b(*)
080b32c4 2f 70 72       ds          "/proc/cpuinfo"
         6f 63 2f
         63 70 75 ...
```

```
                    s_/proc/%d/exe_080b2f64                 XREF[1]:     read_proc_data:0804bb1d(*)
080b2f64 2f 70 72       ds          "/proc/%d/exe"
         6f 63 2f
         25 64 2f ...
```

The sample uses the same XOR-key, that is used in other previous attacks to encrypt its strings and C&C communication i.e. **BB2FA36AAA9541F0**. (see the below code).

```
.rodata:080B30B0                db  1Ah
.rodata:080B30B1                db  43h ; C
.rodata:080B30B2                db  27h ; '
.rodata:080B30B3                db  42h ; B
.rodata:080B30B4                db  42h ; B
.rodata:080B30B5                db  42h ; B
.rodata:080B30B6                db  32h ; 2
.rodata:080B30B7                db  46h ; F
.rodata:080B30B8                db  41h ; A
.rodata:080B30B9                db  33h ; 3
.rodata:080B30BA                db  36h ; 6
.rodata:080B30BB                db  41h ; A
.rodata:080B30BC                db  41h ; A
.rodata:080B30BD                db  41h ; A
.rodata:080B30BE                db  39h ; 9
.rodata:080B30BF                db  35h ; 5
.rodata:080B30C0                db  34h ; 4
.rodata:080B30C1                db  31h ; 1
.rodata:080B30C2                db  46h ; F
.rodata:080B30C3                db  30h ; 0
```

The files that are opened and read by the original malware file are :

/etc/cron.hourly/gcc.sh	/lib/libudev.so
/proc/stat	/proc/meminfo
/var/run/gcc.pid	/lib/libudev.so
/tmp/bVexvNSHcD	/proc/meminfo
/lib/libudev.so	/lib/libudev.so
/proc/meminfo	/proc/meminfo
/proc/cpuinfo	/usr/bin/dzwzcnclqw
/etc/init.d/bVexvNSHcD	/usr/bin/xgslbkjjiu
/proc/rs_dev	/usr/bin/aedjfphajv
/proc/sys/kernel/osrelease	/usr/bin/zkgpjrfden
/proc/sys/kernel/ngroups_max	/usr/bin/zwwfhocxjy
/var/tmp	/usr/bin/uxjyrkzrhj
/proc/sys/kernel/version	/usr/bin/fbtqqebxnt
/usr/libexec/getconf	/usr/bin/twmvqrwbhl
/proc/sys/kernel/rtsig-max	/usr/bin/rzrdxbhbdw
/usr/bin/lywqbfmulc	/usr/bin/eymbnveibk
/usr/bin/mmvptggmnc	/usr/bin/naukgaquuk

IOCs:

01837ff17eaf96a37a95e2f6c5ebbbe	7c2de60fc9aa3619146a4bd458b5f1a
0190d66de838c766a844e52c1f4f047	7c96492d9922d5ca56782e1ed07f5e2
07901e0d18823b5f8909c95e0e7cde1	7e0c2f45e231b6f7b5324e2e5ca2f1f
0851395d31a889eb2fbc40576781a89	8195401195f52008f2680bd3f1d17ec
08cf2ab893d634b2a6f5d9ee47b55a3	83eea5625ca2affd3e841d3b374e88e
09489b80975a6f1a076784f0838c91b	84bcefccce52f403231a3a6cffc2683
0d6cb79acdb715f4198b6133f574e39	88ffaf0fdff2e598a915e8d60be6da8
1093d9e4b8ce510bb0f9d6ff8fb2659	8a9295908c1c4a948f3cbf553be30c2
12d5d743f029662fa0d4c7d224d2f11	8ba5afdd3b19b7a6f7c94332472d668
1c1748bdc1a73be3117f696e449e2fa	8c8da16a2b9e7c318a9544ff032bddb
1c510a878094f4fac8d2d5eca508ae8	98eaf214831e2981d834a391b94cdfe
1c6324f652df5ba49b9b7187344a6f5	9a15e92854143e58f3adf74cc995604
1cd71d3bb891df0b251249da0cdcf7e	9a79419a06f1a4ea22891b2a2ee2402
2004f9f08f281f8d4ea7c913573dd6c	a14578469fab44514dfca6c4eead755
20a842bf68912c68d403deeda82675e	a37bf50d53fb2409c16d7007d018cc8
20e858d85ce4974142e9052db597bdc	a3bd639bd3113a2e2be97b02ab9531f
232e172f7a005dd12d4aad55e0c4a33	a609138f0791ce100bd2ad8efa1f74b
24fa9dbd7ea29cbc1f549e9de4bf67a	a6d685769d19e12afde7bc83a683174
277b7d71dc3d4c1376535729a24da81	a99c10cb9713770b9e7dda376cddee3
28b4c1d34913014f2ea43298db49321	ab1fc5148c2872ea51ba5dccd1e8b7e
2eed2d8674727bbf4af4d297fe37f59	ad7a3089ab9e272987028e08c39b0d2
2ef27aea280833a7ce52643334b2fb4	b37687858779dcb5ad51c82c74b493a
3291432c0084225333ee57320404e65	b6e04d4eea2d4e044b9b5c3dde3bce0
36387ccda369530bc9a4a68e15b1f19	b9cb431c103bd716493a7b70133012d
3732b8afe4b1a11bf6484285377l5e7	ba1a59cbaeb7fa581dc1f186f198932
3765193e92c10eab1dc09a2c8985773	bc25e822c18875a4dc129ab00b9c8d9
38b27df54a2010cfe3f618623f0b10d	bcb6b83a4e6e20ffe0ce3c750360ddf
3b68ff7b7620057a246d81f646d4128	bcf80d78a918b22179c51cc68d67184
3c49b5160b981f06bd5242662f8d0a5	bf2b4a61fc7c39659a2570ed27fa155
3e34bff8e13cf6068f4a30218b55b54	c1ec720ad4e847ff37bfdcebbe5b30d
42ba80053b0e744346236592b01949d	c3b424c0978555704a2395c2664ae67
456cd7ed73b3da798fd7e4231a93594	c663827b1cf068ff2e2b1a731bbf282
4ae0d00d50a95510a4c0f8e5c65ace1	c83e6dda8083c46a5fe3e26381afbba
50b176dd2a0888bd18ff13bf7484077	c9cf8eee7586fec1e5ca863ca3d0a92
53360fc328b98bbe24ccb9e57c37d9f	caeeadeea0762565473ac39681101c2
55a111f4625348cffd6d910e49f5dbd	d1b5b4b4b5a118e384c7ff487e14ac3
569a3c552c968483445efabdb8bed9f	d1e123571b9714d054daa477992a509
5b9a58a9babe102d41c4be6aafa9169	d6a5d9bd5e6842bb595b18a9131a84a
5f2f8a05ed17c6b47a6a9655fc2f6f6	d71c80b727f73746dc0f129ad4e8cac
616b7d37976b466c6c4ca41909d35f6	d80db877543979651006f98a643e700
62321b43aef605d3713b9a2aa9717a6	e10409962c67021819eecbea5e41726
6395aafd2335a87f431bcf45adebd80	e1759be6a2317bd48d72c93be11974e
63f286aa32a4baaa8b0dd137eb4b336	e196d3c7950235ac0e08839faa0138c
64ad5b3ff3a4845e7e26d17aa0170dc	e3a4aaf829b9d16b938e8b7915e67b5
64d4f0e77b72980fb21c81e69e4bb5f	e47bb3c56a32881b7f690fcb568044d
6b2f167c56c262fa8c4b6619729c747	e555d685a5162bcef0f59a34a68ef81
757b89c6cc5a910c11a555a381684e5	e7a3aa891e550834f9af4367a564e46
79a7792955c2e7137c68bec4803ce65	f45232c67ef011ef988747e6d9bbb44
79bcd3bc33b21283428502db3e971e2	f68034c9cbd393f1b69977e3265f62e

7a891a96d6af45865e5fe6142b40eb7	fcc3be118f2bc42f8757eef69496e89

SECTRIO PROTECTION:
SECTRIO detects the XorDDoS ELF malware as ' SS_Gen_XorDDoS_ELF_A '

VULNERABILITIES TARGETED:

It mostly spreads via SSH (Secure Shell) services that are vulnerable to password cracking or mainly brute-force attacks.

- CVE-2012-5975
- CVE-2011-0766
- CVE-2005-4310
- CVE-2003-1120
- CVE-2002-1646
- CVE-2001-0553

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.