Sednit espionage group now using custom exploit kit

For at least five years the Sednit group has been relentlessly attacking various institutions, most notably in Eastern Europe. The group used several advanced pieces of malware for these targeted attacks, in particular the one we named Win32/Sednit, also known as Sofacy.

ESET Research 8 Oct 2014 - 08:49AM

For at least five years the Sednit group has been relentlessly attacking various institutions, most notably in Eastern Europe. The group used several advanced pieces of malware for these targeted attacks, in particular the one we named Win32/Sednit, also known as Sofacy.

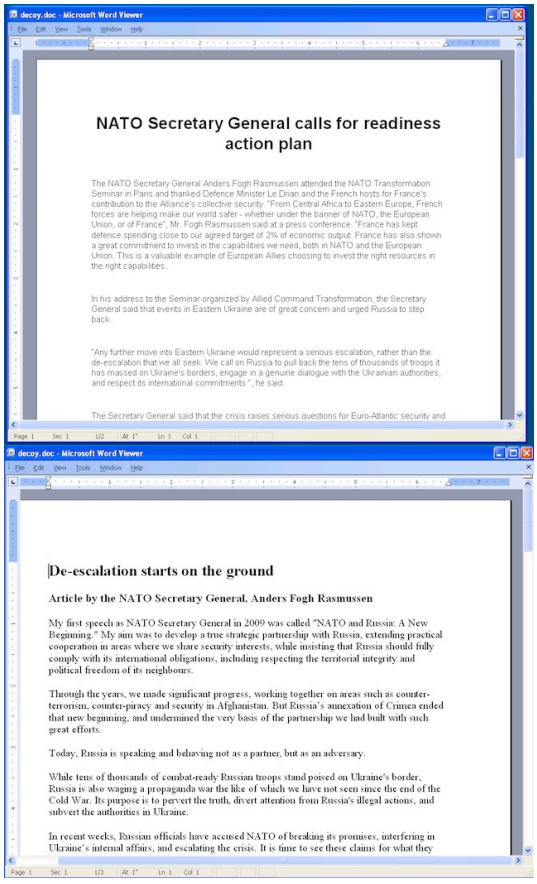
We recently came across cases of legitimate financial websites being redirected to a custom exploit kit. Based on our research and on some information provided by the Google Security Team, we were able to establish that it is used by the Sednit group. This is a new strategy for this group which has relied mostly on spear-phishing emails up until now.

In this blog, we will first examine on recent cases of spear-phishing emails using the CVE-2014-1761 Microsoft Word exploit. We will then focus on the exploit kit, which appears to still be in development and testing phase, and briefly describe the actual payload.

From Spear-Phishing Emails...

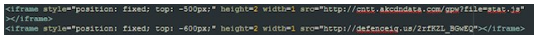
Back in April 2014, the Win32/Sednit malware was being delivered through a 0-day vulnerability in Microsoft Word RTF documents, CVE-2014-1761. It was amongst a small number of malware families being delivered through this vulnerability, like BlackEnergy and MiniDuke, which are also used for targeted attacks.

Here are two decoy documents showed to the victims while the vulnerability was silently exploited on their computer. Both of these documents present NATO views on the Ukrainian conflict.

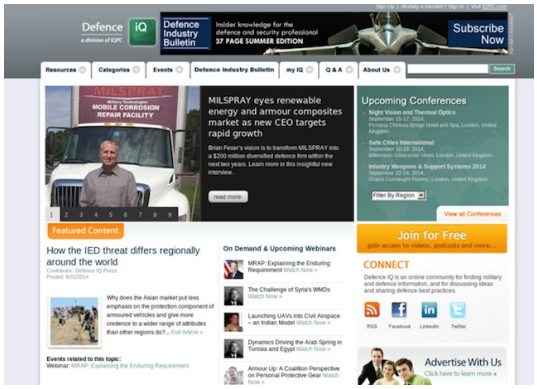


...to custom Exploit Kit

We observed redirections to the exploit kit from websites belonging to a large financial institution in Poland. The exploit kit is loaded through a simple IFRAME that is added near the end of the HTML document; for example hXXp://defenceiq.us/2rfKZL_BGwEQ in the screenshot below. We can also see a very similar looking IFRAME loading the URL hXXp://cntt.akcdndata.com/gpw?file=stat.js, whose domain name was registered on September 18th. We were not able to retrieve the content of this page but we suspect that its purpose is to collect statistics about the number of redirections.



When directly visiting the URL hXXp://defenceiq.us, we were redirected to defenceiq.com, a legitimate website that describes itself as "an authoritative news source for high quality and exclusive commentary and analysis on global defense and military-related topics".



The domain defenceiq.us was found to resolve to 76.73.47.90. Other suspicious domains also resolved to this IP address and displayed the same redirection behavior when visited, which is a strong indication of the sectors the group is currently targeting. The redirection from Polish financial websites to a defense-related domain name seems less than optimal for a targeted attack and was probably due to the mixing of two ongoing campaigns.

Exploit kit domain	Redirects to	Website content
defenceiq.us	defenceiq.com	Military news
armypress.org	armytime.com	Military news
mfapress.org	foreignaffairs.com	Foreign Affairs magazine
mfapress.com	foreignaffairs.com	Foreign Affairs magazine
caciltd.com	caci.com	CACI International, defense & cyber security contractor

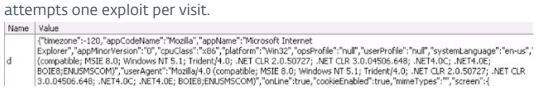
The exploit kit, which we named **Sedkit** as a reference to Sednit, behaves in a similar fashion to others commonly used today, such as the Angler or Nuclear exploit kits. A sample exploitation chain is shown below. The browser is first sent to the landing page which uses JavaScript to detect the browser and installed plugin versions.

Interestingly we can see that the call to DetectJavaForMSIE() is commented out. This follows the current trend in exploit kits of not targeting Java, because recent versions of Java and browsers warnings before loading applets. At the moment only Internet Explorer seems to be targeted: when we tested with Chrome and Firefox we were always redirected to localhost.

```
if(navigator.userAgent.indexOf("MSIE") > -1) {

        string_of_json += DetectPdfForMSIE();
        string_of_json += DetectFlashForMSIE();
        //string_of_json += DetectJavaForMSIE();
    }
    else {
        string_of_json += EnumeratePlugins();
    }
```

The browser then sends back the plugin information via a POST request. Based on this information, the exploit kit redirects the browser either to another URL containing an exploit, or to http://localhost. The kit only attempts one exploit per visit.



We recovered 3 different exploits used by the kit, all targeting Internet Explorer. They are listed below, with the specific version of IE each one targets. Interestingly, CVE-2014-1776 has not yet been seen in any popular exploit kits, and the other two have also seen only limited adoption.

CVE	Targeted IE version	Microsoft Security Bulletin
CVE-2013-1347	Internet Explorer 8	MS13-038
CVE-2013-3897	Internet Explorer 8	MS13-080
CVE-2014-1776	Internet Explorer 11	MS14-021

However other aspects of the kit lack refinement. Unlike most contemporary exploit kits, it does not use JavaScript obfuscation. We even found comments in the JavaScript code related to the exploits' ROP chains. This leads us to believe that the kit is still in its testing phase.

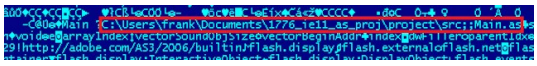
```
var ate1          = 0x77BD18D3 ;//xchg eax,esp#          77c118d3
var atz1          = 0x77BCEF5B ;//xor eax,eax#ret        77c0EF5B
var co1           = 0x77BCF519 ;//pop ecx#retn          77c0F519
var pco1          = 0x77BD3E25 ;//mov ecx,[ecx + 8]# retn    77c13e25
var jtc1          = 0x77BE746A ;//push ecx#retn          77c2746a
var vPP1          = 0x77BC1120 ;//virtualprotect         77c01120
```

Figure 1 From CVE-2013-3897 exploit code

When uncompressing the Flash file used for the CVE-2014-1776, a path is visible. This information is not found in previous samples of this exploit in our collection.

```
for (i = 1; i <= (0x42c-4) / 4; i++) {
    if (i == 0x29) {
        overwrite_str.push(0x18182O2C); // offset was 0x94 now 0xA4
    }
    else if (i == 0x2B) {
        // TODO not needed?
        overwrite_str.push(0x12010019);
    }
    else if (i == 0x42) {
        overwrite_str.push(0x18182214);
    }
    else if (i == 0x66) {
        overwrite_str.push(-1);
    }
    else if (i == 0x46) {
        overwrite_str.push(0x18182214); // mov     eax, [esi+198h]  # shr     eax, 0Ch # test    al, 1
    }
    else {
        overwrite_str.push(i);
    }
}
```

Figure 2 From CVE-2014-1776 exploit code



Upon successful exploitation the payload is downloaded; whether it is encrypted depends on the exploit.

Payload

The binary deployed on the infected machine is named "runrun.exe". Its sole purpose is to deploy a second program – initially encrypted and compressed — on the machine and ensure its persistence on the system. The second program is a Windows library named "splm.dll". According to our data, this malware has been employed in targeted attacks since at least 2009.

Roughly summarized, this payload has been created with a C++ framework. Thanks to the Run-Time Type Information (RTTI), a part of the program architecture can be reconstructed with the names chosen by the programmer. The malware contains *agent modules* implementing malicious activities, and *channels* for communications between modules and remote controllers. In this sample, we found the following agent modules, identified by a 16-bit ID:

Module Name	ID	Purpose
AgentKernal	0x0002	Execution manager
ModuleRemoteKeyLogger	0x1002	Keystroke logging
ModuleFileSystem	0x1102	File system accesses
ProcessRetranslatorModule	0x1302	Provides communication means

It also instantiates one external communication channel named **WinHttp**, which decrypts three domain names used as command and control: msonlinelive.com, windows-updater.com and azureon-line.com.

Conclusion

In recent years, exploit kits have become a major method employed to spread crimeware, malware intended for mass-scale distribution to facilitate financial fraud and abuse of computing resources for purposes such as sending spam, bitcoin mining, credentials harvesting etc.

Since 2012, we observed this strategy being used for espionage purposes in what has become known as "*watering-hole attacks*" or "*strategic web compromises*". A Watering-hole attack can be described is redirecting traffic from websites likely to be visited by members of a specific organization or industry being targeted. In ESET's retrospective on Windows exploitation in 2013, Artem Baranov wrote *"the past year can rightly be called the year of targeted attacks and watering hole attacks"*.

While many instances of watering-hole attacks have been documented for related actors in cases such as noted by Symantec in their Elderwood Project report, we are now seeing this strategy being adopted by another

group and it seems likely that others will follow them.

Indicators of compromise

Here are some indicators that could help to identify the payload sample dropped by the exploit kit described in this blog post:

- Presence of the CLSID {d702b440-b130-47f7-a94c-c1fae33d2820} under the registry key HKLM\Software\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad
- Mutex named "XSQWERSystemCriticalSection_for_1232321"
- Mailslot named "\\.\mailslot\check_mes_v5555"
- Files in temporary folder (as returned by GetTempPath()) named "edg6EF885E2.tmp" and "edg6E85F98675.tmp"
- Network communications with the domains named msonlinelive.com, windows-updater.com, or azureon-line.com

Hashes

SHA-1	Role	Detection name
86092636E7FFA22481CA89AC1B023C32C56B24CF	Word exploit	Win32/Exploit.CVE-2014-1761.D
12223F098BA3088379EC1DC59440C662752DDABD	Word exploit	Win32/Exploit.CVE-2014-1761.D
D61EE0B0D4ED95F3300735C81740A21B8BEEF337	Dropper	Win32/Agent.WLF
D0DB619A7A160949528D46D20FC0151BF9775C32	Payload	Win32/Agent.WLF

ESET Research 8 Oct 2014 - 08:49AM

Similar Articles

Tracking Turla: New backdoor delivered via Armenian watering holes

Guildma: The Devil drives electric

Up close and personal with Linux malware

Linux and malware: Should you worry?

Discussion