Nefarious Macro Malware drops “Loki Bot” to steal sensitive information across GCC countries!

cysinfo.com/nefarious-macro-malware-drops-loki-bot-across-gcc-countries/

5 years ago

Macro malware are still playing its atrocious activities in the wild, frightening all the sectors around the globe. Latest Spam campaign which flew around GCC countries created a “scary rain” across multiple entities.

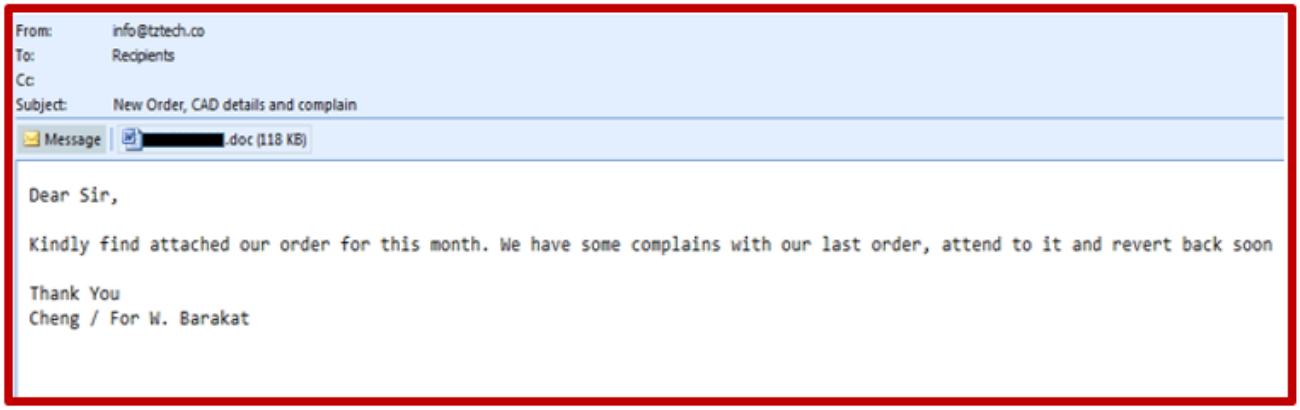
This spam mail was not targeted only for a particular entity, but extensively across multiple firms in Middle east, anticipating huge number of victims. On the other hand, the recipients in these mails (BCC) were clearly social engineered.

NB:

> *The malware and associated files were analyzed within private secured environment, without actually allowing it to communicate to its command and control*
>
> *While analyzing, we may come across with unhygienic words or phrases. Keep in mind that, malware are built by “Bad Boys”.*

Let’s Get Serious:

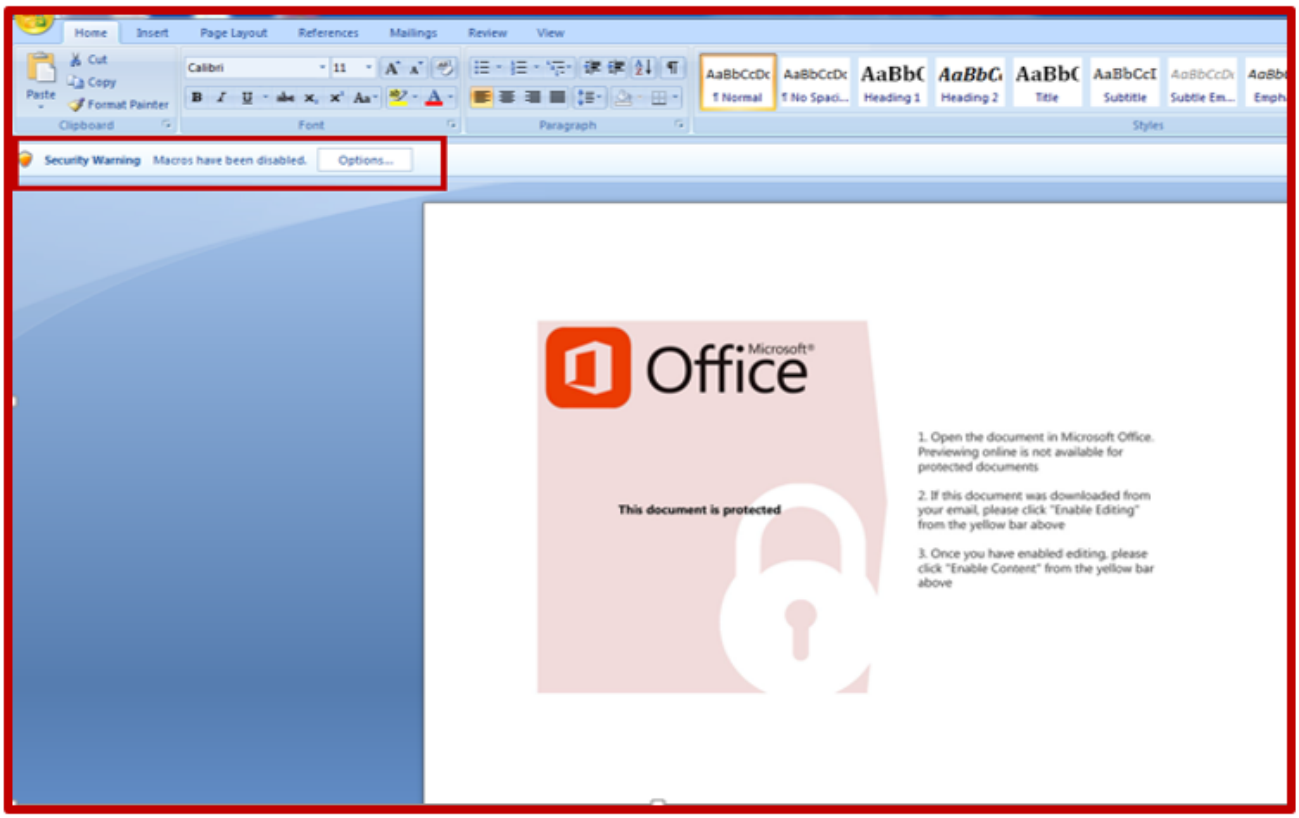
The spam mail which landed on one of the victim’s Mailbox looks like this:



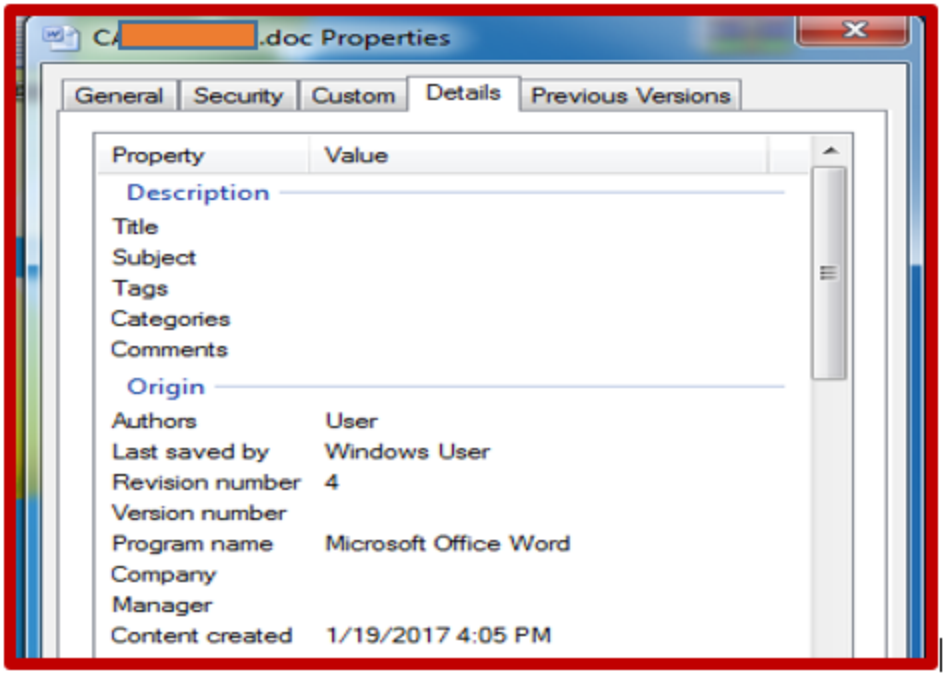
The sender Address could be spoofed, which is the contact email ID of the Cambodia based Business software provider firm **“tztechnology”**. The reputation of the sender IP address is poor:

The attachment was a document file and once it is opened, the prompt for enabling the macro starts blinking:

> Still end users are falling for these.. sad truth!!



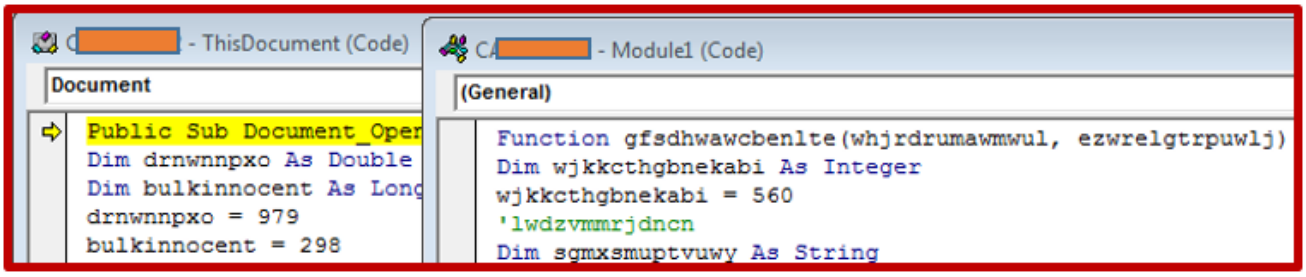
The word document properties shows, revamped or created date as “Jan 19th 2017”



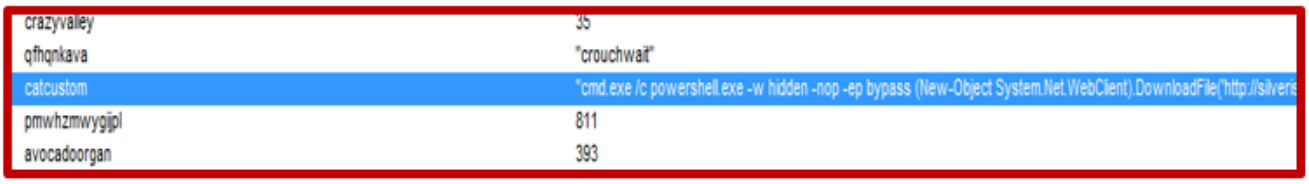
Jumping into the Document Macro, starts with “Document_Open()”, meaning , the code will be right away executed whenever the document is opened.

The VBscript contains lot of junk and unwanted parameters, which would make static analysis to choke. Also parameters inside the code seems to be encoded heavily. So at this point a mixture of static analysis and debugging needs to be done.

When we statically analyze, we can see two modules of codes present in document. Both of the module works together to build a command script and then to run this script via windows script object.

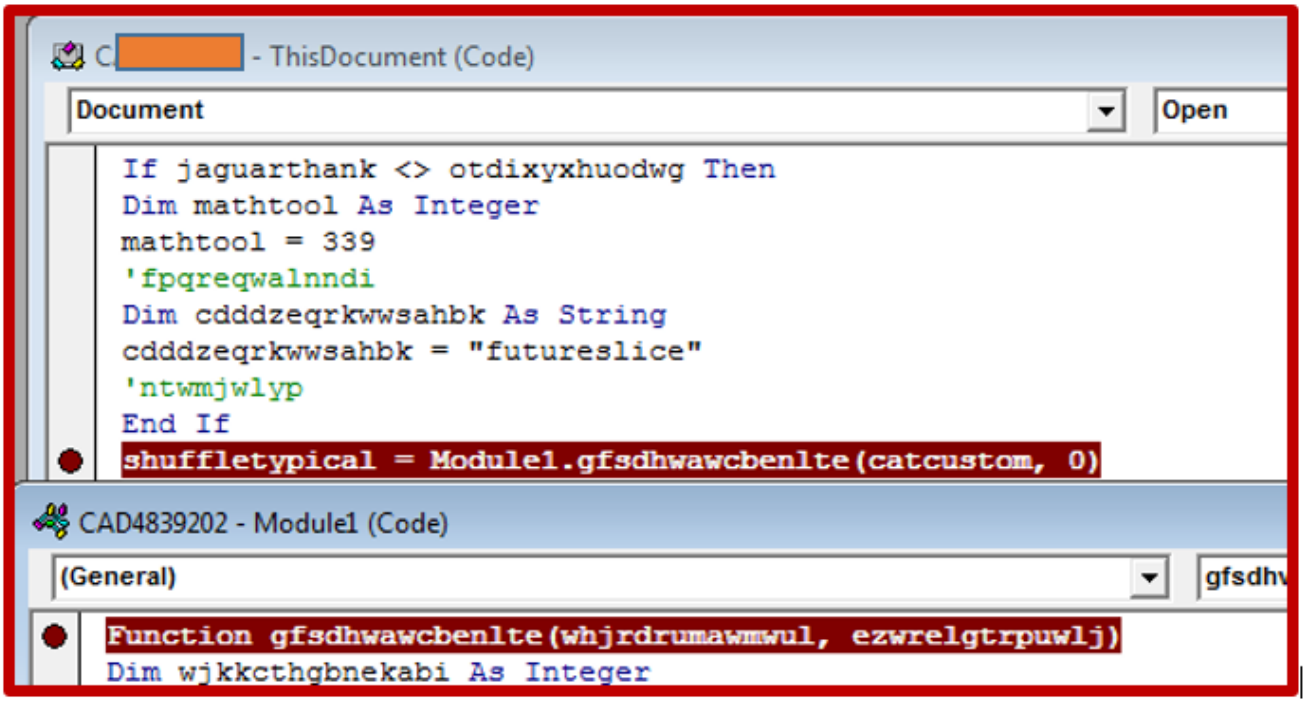


Further debugging and static analysis, found that one of the variable **“Catcustom”** stores command script which was built by the macro on the fly.

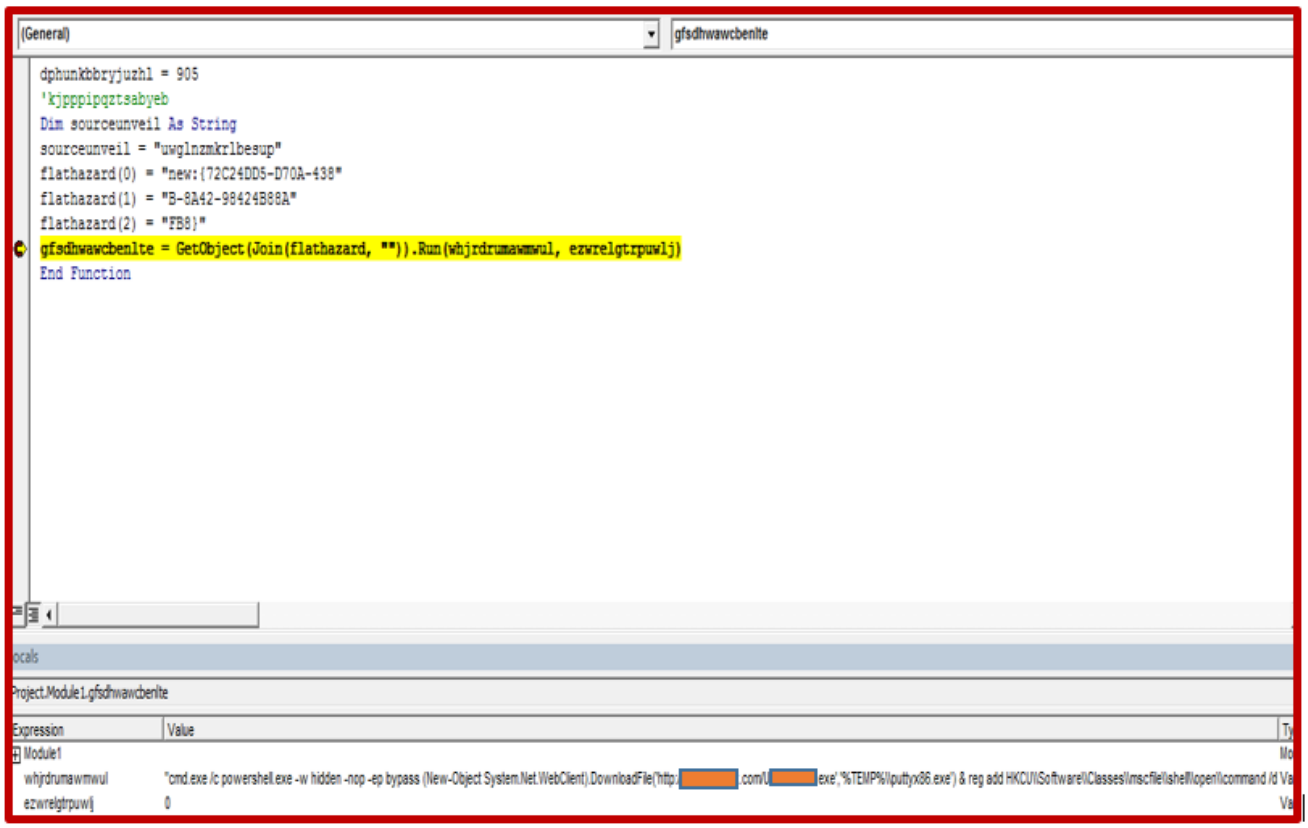


The generated code looks like this (after enumeration of temp folder):

The below snippet of code reference is the “bridge of relationship” between two modules of scripts. The earlier mentioned variable **“Catcustom”** which contained the commands where used as a parameter of another function, which is then referenced to the second module “Module1”. The referenced Function parameters **“gfsdhwawcbenlte()”**now contains the value of **“catcustom”** variable and **“0”** .



Furthermore coming down to the script at second module“Module1”, we can see malicious script was invoked by calling the windows script shell object:



> Now the question is how we understood from this code above, that it invoked windows script shell (with hidden window) to run the malicious code which earlier generated.

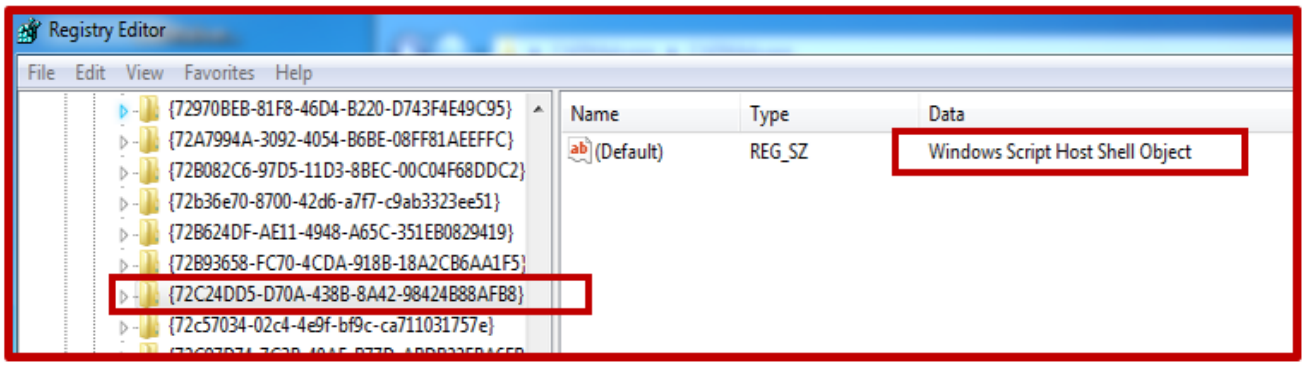
If we closely look the above snippet of code,

```
flathazard(0) = "new:{72C24DD5-D70A-438"
flathazard(1) = "B-8A42-98424B88A"
flathazard(2) = "FB8}"
gfsdhwawcbenlte = GetObject(Join(flathazard, "")).Run(whjrdrumawmwul, ezwrelgtrpuwlj)
```

The function is getting the “new object” by joining **flathazard(0), flathazard(1) and flathazard(2)** to get :

new:{72C24DD5-D70A-438 B-8A42-98424B88A FB8}, Now if we go to the registry **“HKEY_CLASSES_ROOT\CLSID\{72C24DD5-D70A-438B-8A42-98424B88AFB8} “**, this ID refers to the windows script shell object.

Meaning, the function is calling a new windows script shell object instance to run the malicious commands in **“whjrdrumawmwul”.**



We can also see **“whjrdrumawmwul”** contains the value of generated script. The **“ezwrelgtrpuwlj”** contains the value **“0”**.

> That said, Let’s see the syntax for **.Run** command in **VB**:

Objshell.Run (strCommand, [intWindoStyle], [bWaitOnReturn])

“Objshell”, We already found how shell object was invoked and we saw ***“strCommand”*** value in variable **“whjrdrumawmwul”.** Now **“ezwrelgtrpuwlj”** holds the value **“0”** which means the **“hide window”**. The ***“bWaitOnReturn”*** if left blank immediately returns to script execution.

Hence we found that the below code was executed by invoking windows script shell object and being executed in hidden window:

We can also see that the PowerShell is invoked in hidden mode, bypassing execution policy to download a malicious executable from a remote host, which is then renamed to "**puttyx86"**. The addition of this temp path of malicious executable to the above registry and then invoking the "eventvwr.exe" is a technique to bypass the UAC feature inorder to acquire highest integrity for executing the malware.

The above fileless technique of bypassing UAC has already been explained in my post of a real-life scenario::

https://www.linkedin.com/pulse/newborn-macro-malware-generates-powershell-script-winston?trk=pulse_spock-articles

Real-life usage of the technique and similar code generated by Macro is drafted in below article:

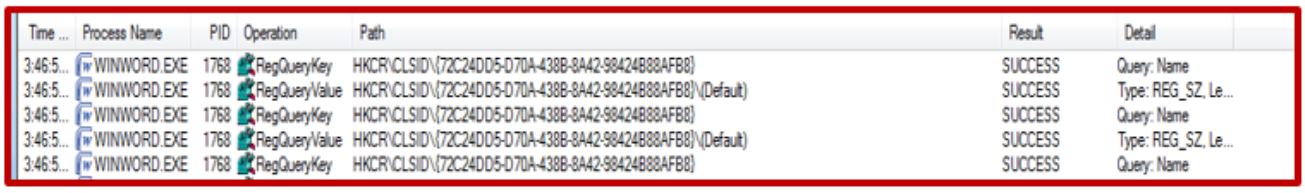
https://cysinfo.com/cyber-attack-targeting-indian-navys-submarine-warship-manufacturer/

And the mechanism of UAC bypass technique drafted in the blog:

https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/

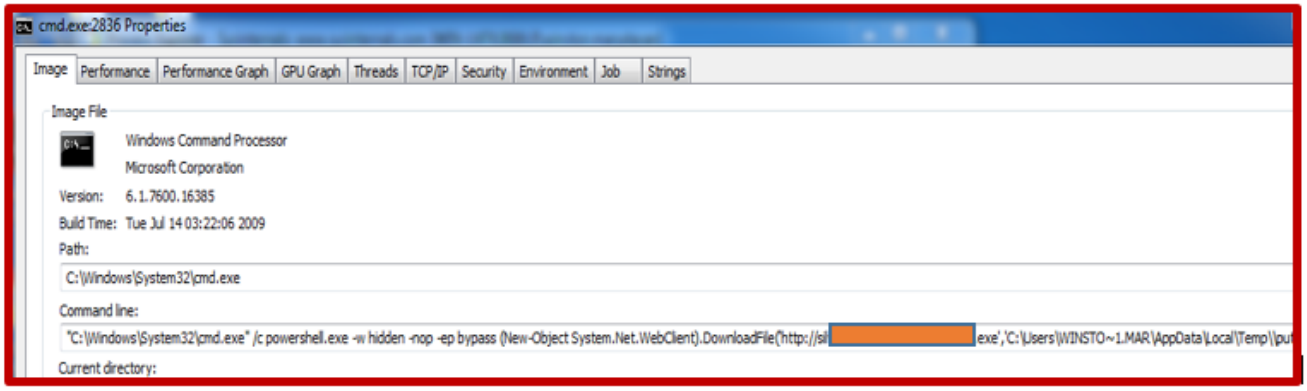
Let's find whether the above findings are true by doing a dynamic analysis:

As we discussed earlier the windows script shell object is invoked via registry with Class ID



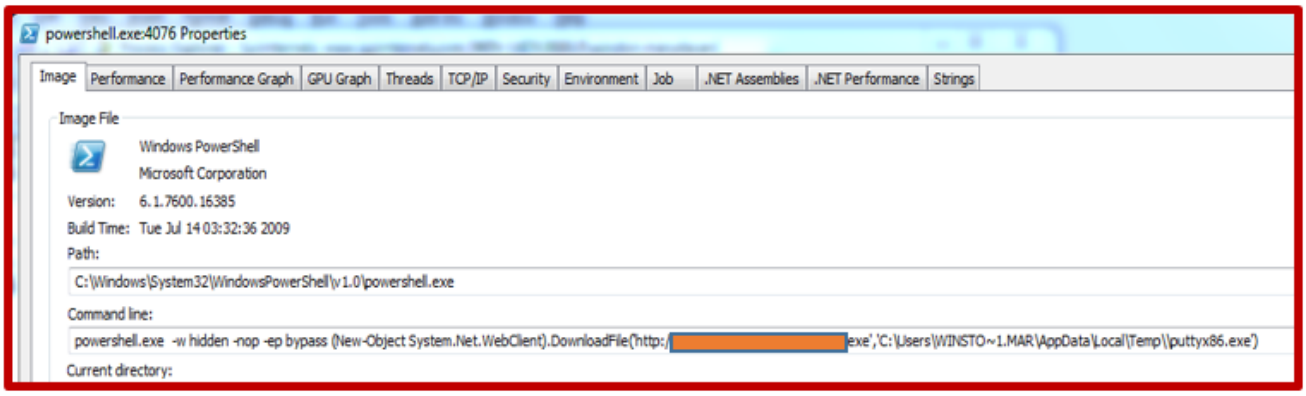
Time ...	Process Name	PID	Operation	Path	Result	Detail
3:46:5...	WINWORD.EXE	1768	RegQueryKey	HKCR\CLSID\{72C24DD5-D70A-438B-8A42-98424B88AFB8}	SUCCESS	Query: Name
3:46:5...	WINWORD.EXE	1768	RegQueryValue	HKCR\CLSID\{72C24DD5-D70A-438B-8A42-98424B88AFB8}\(Default)	SUCCESS	Type: REG_SZ, Le...
3:46:5...	WINWORD.EXE	1768	RegQueryKey	HKCR\CLSID\{72C24DD5-D70A-438B-8A42-98424B88AFB8}	SUCCESS	Query: Name
3:46:5...	WINWORD.EXE	1768	RegQueryValue	HKCR\CLSID\{72C24DD5-D70A-438B-8A42-98424B88AFB8}\(Default)	SUCCESS	Type: REG_SZ, Le...
3:46:5...	WINWORD.EXE	1768	RegQueryKey	HKCR\CLSID\{72C24DD5-D70A-438B-8A42-98424B88AFB8}	SUCCESS	Query: Name

Next the "**cmd.exe"** has the entire script running under it.



As it step by step runs the commands in cmd.exe,

PowerShell is invoked with the script to download the malware from remote host and save to temp folder as “puttyx86.exe”

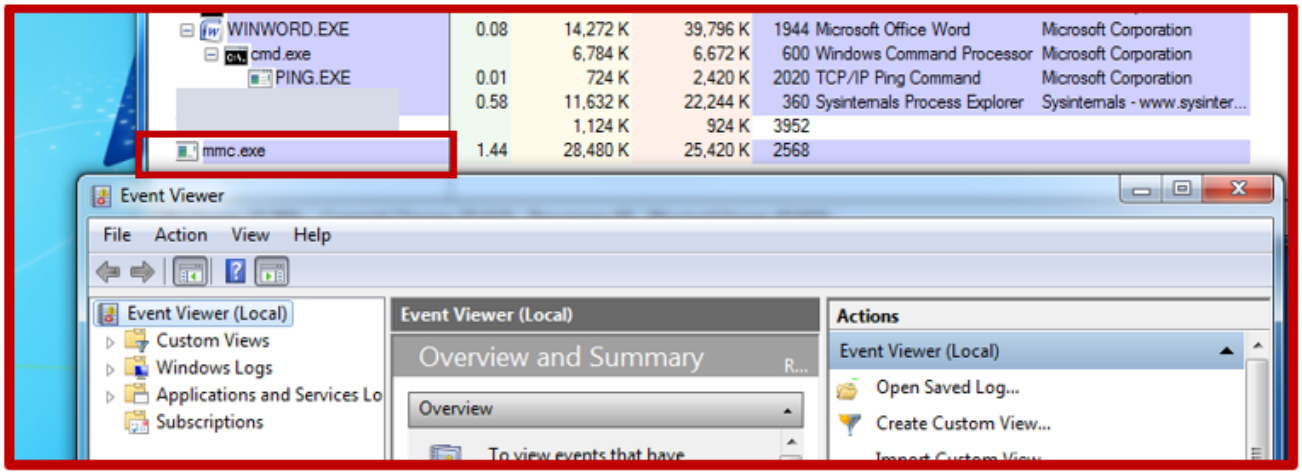


Glitch while Acquiring Highest Integrity via Eventvwr.exe

In our sample, there happened a small glitch while script was trying to write the malware path to **“HKCU\\Software\\Classes\\mscfile\\shell\\open\\command”** registry to be executed via **eventvwr.exe**. It may be due to extra slashes, because when I tweaked commands from **“\\”** to **“\”** the Registry write was successful.

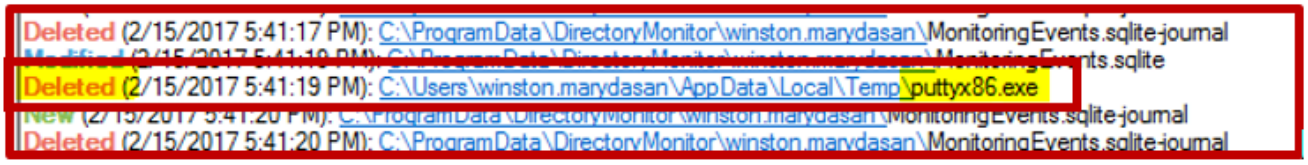
Due to the glitch, original eventvwr.msc popped up instead of malware when the macro was executed, quiet unlucky.

If you see in the below picture the **“mmc.exe”** initiated the **eventvwr.msc** normally.

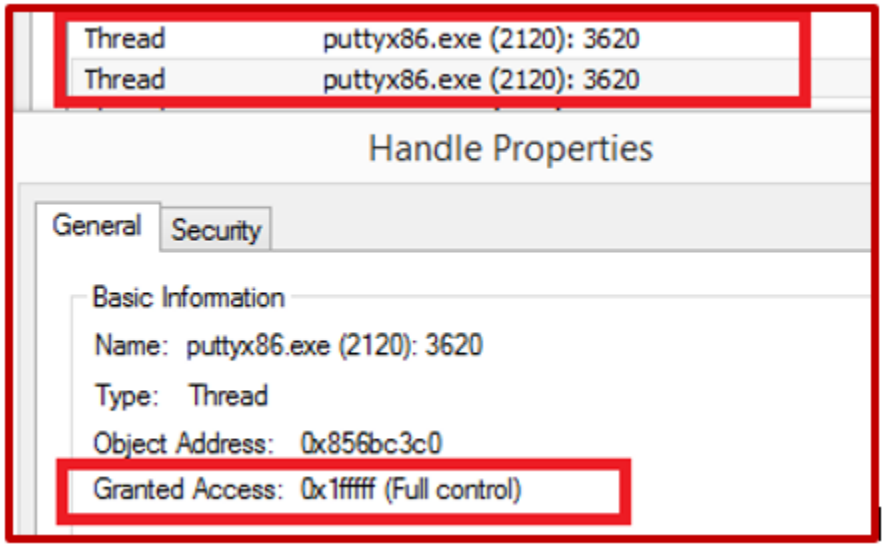


However, even though the script couldn't invoke malware via **"eventvwr"** technique, after a 15 PING-sleep (Using Ping command 15 times redirecting to nul), the malware at temp folder was directly invoked. Which made the malware to run with medium integrity.

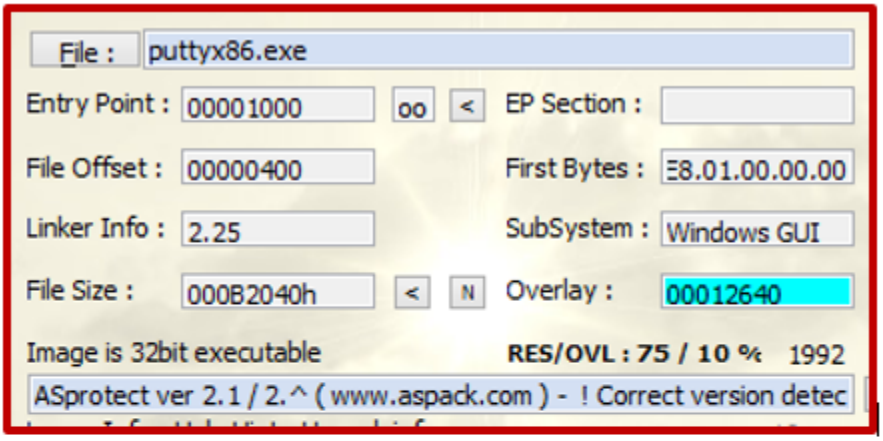
Once the **"puttyx86.exe"** is executed normally, it spawns a child of its own and kills the parent process. Also managed to delete the executable from the path.

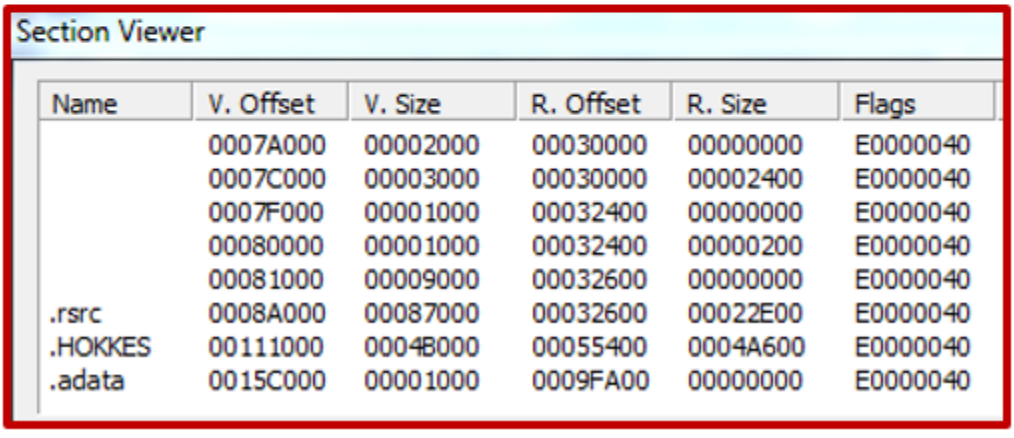


Then if we see the handles for the child process, it acquired full access for each thread. The Malware must have its own elevation feature.



The dropped malware seems to be protected by the infamous “ASprotect” executable protection, header of the file also throws the acknowledgment with bogus section names.

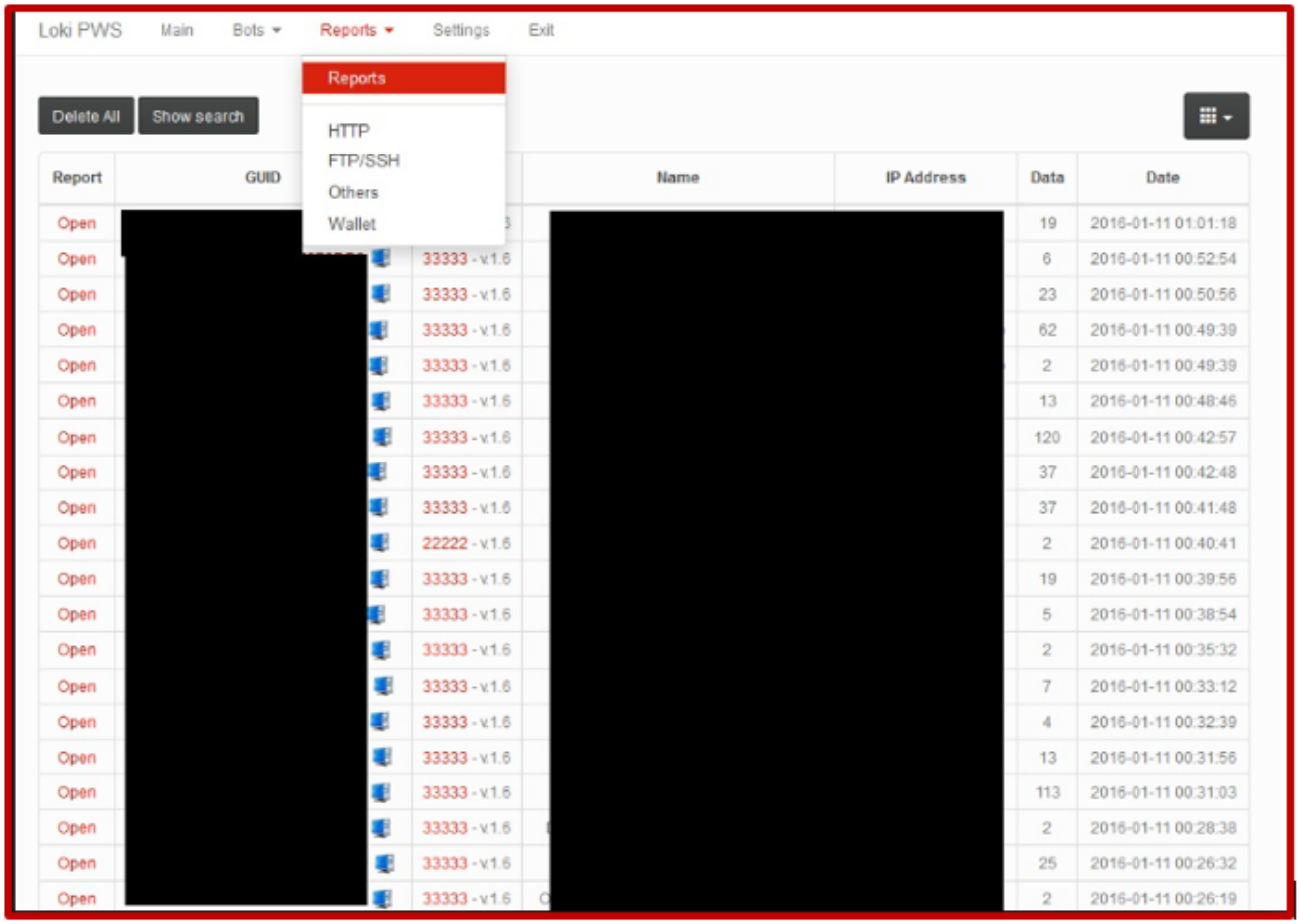


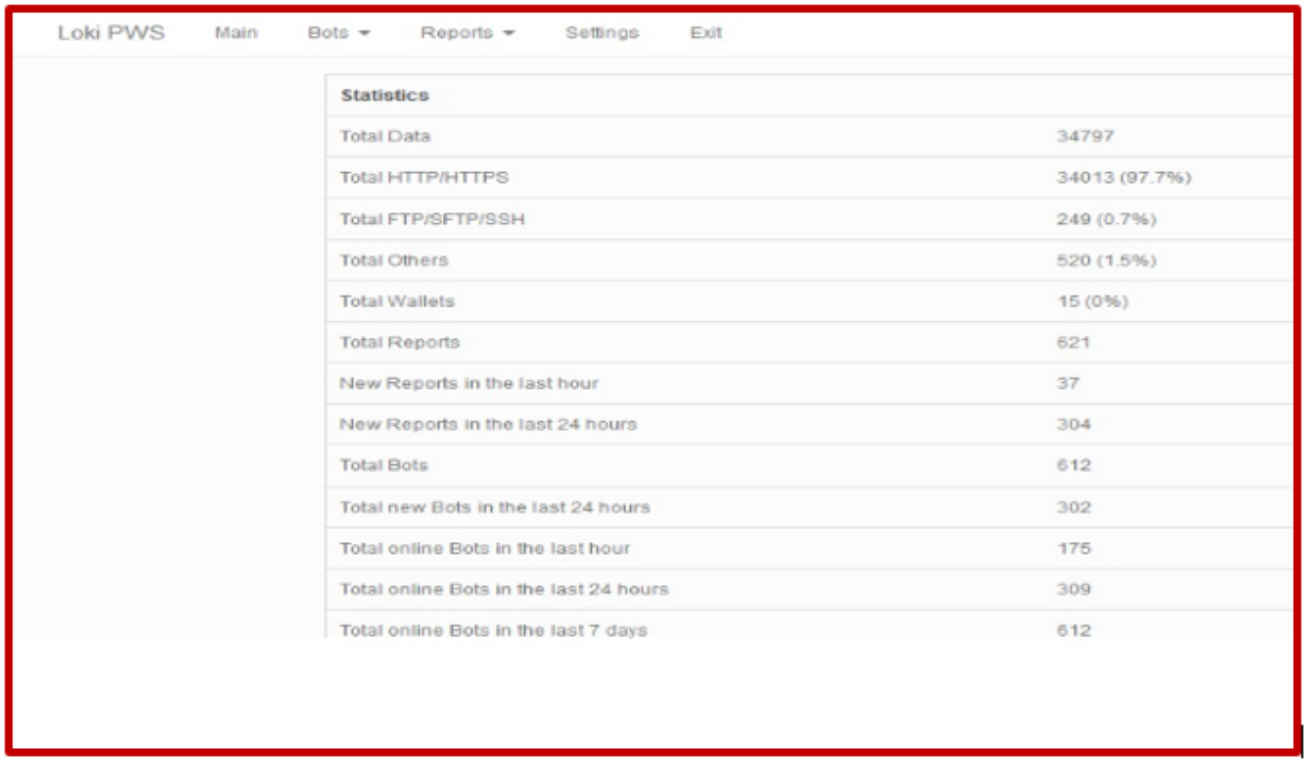


After a tug of war between the malware using static code analysis and debugging, found that the piece of malware was piece of infamous **“Loki Bot”.**

> *"Loki Bot is resident loader and password and cryptocoin-wallet stealer. It comes with wallet checker (coin inspector, read below). It can steal passwords from browsers, ftp/ssh, e-mail and poker clients. Written in C++. Works on Windows XP, Vista, 7, 8, 8.1. and Linux. UAC Bypass"*

Below shown pictures are snips from the actual interface of main Loki Bot and





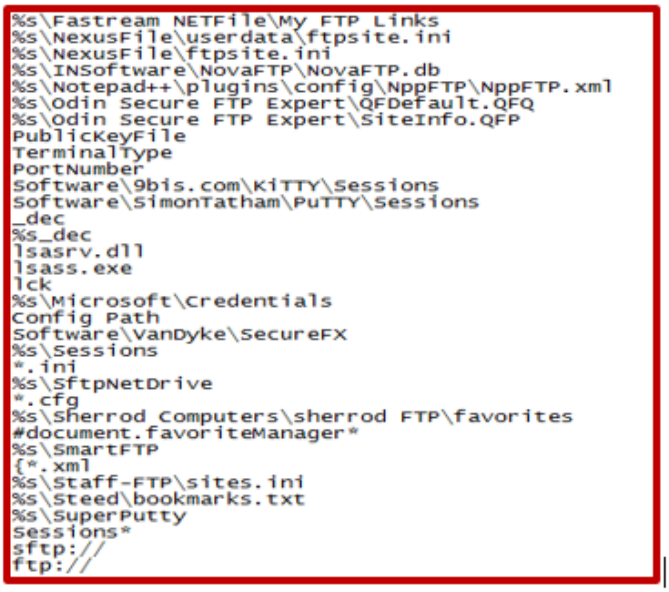
The dropped Malware had most of the Anti-analysis capabilities like VMawareness, Debugger detection, System time check and more. Carefully tweaking these will make malware into running as if in a physical machine.

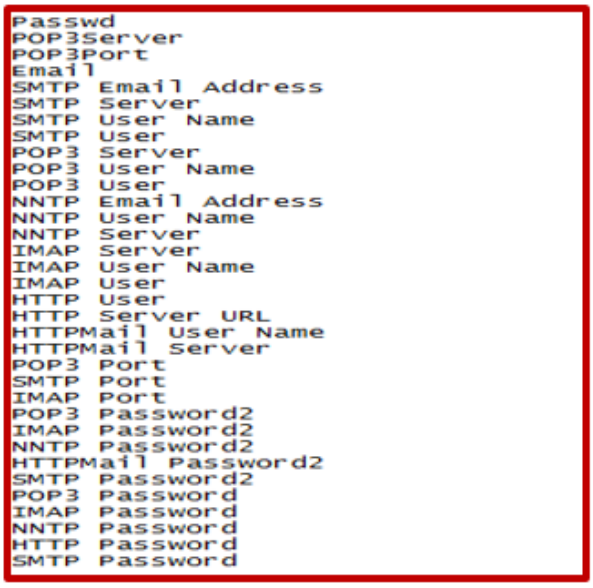
If we try to see the strings of malware without unpacking from the ASProtect protection mechanism, we will not get any “sweet fruit”. But after debugging and disassembling, we will get good amount of data about of the malware which is obviously fruitful.

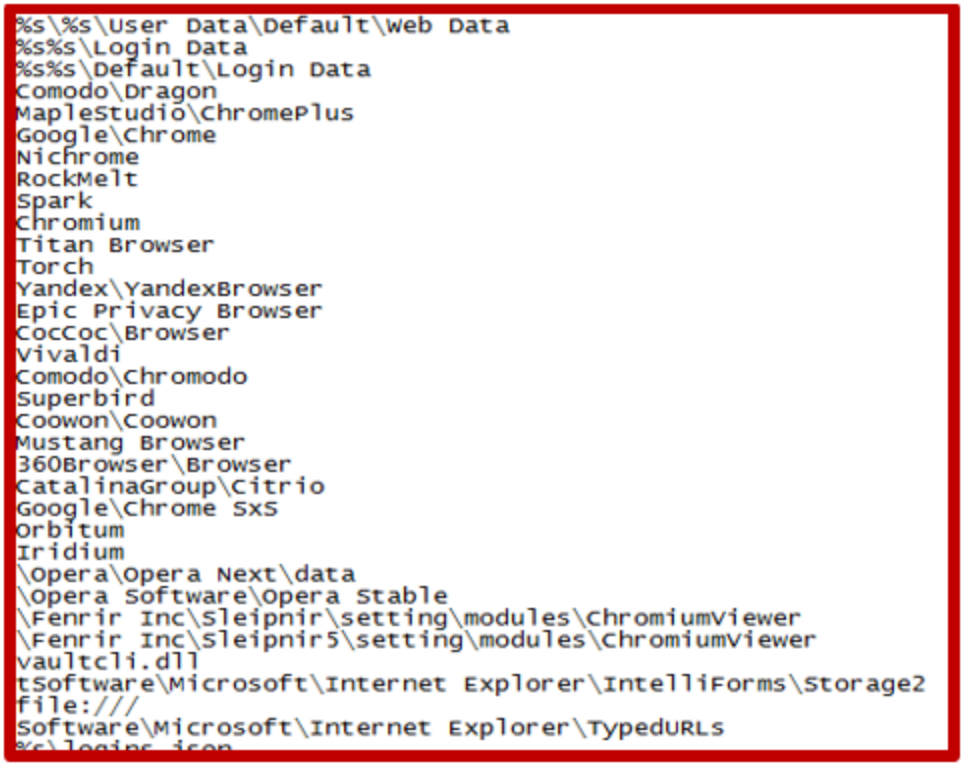
That said, I was able to retrieve and filter very useful data about the malware which gives enough evidence about the above said malware.

The capability of this malware is enormous and even have capability of receiving the Bot commands from “BOT Boss”.

The malware have capabilities for luring all the FTP flavored credentials, SMTP, Browser data, DBs information, have inbuilt Key logger features and much more. The portions of retrieved strings are below:







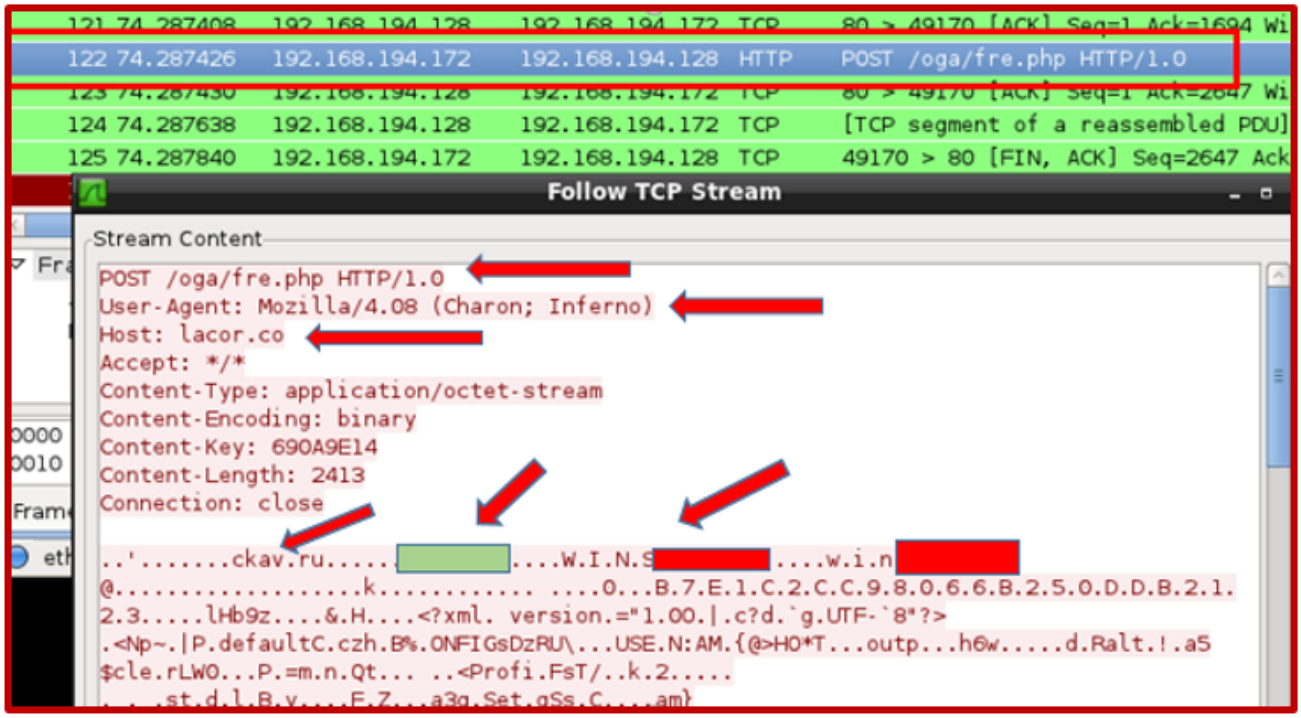
In addition to that, malware gets the details about the current user, Machine name, FQDN, MachineGuid and so on

A hardcoded URL was very promising though, suspecting the above collected details and this URL must have some connection.

```
getaddrinfo
freeaddrinfo
WS2_32.dll
GetLastError
SetLastError
HeapAlloc
HeapFree
GetProcessHeap
KERNEL32.dll
CoInitialize
CoUninitialize
CoCreateInstance
ole32.dll
OLEAUT32.dll
http://lacor.co/oga/fre.php
```

If we see the network traffic generated by the malware, we can see a promising “Post” traffic to the above found hardcoded URL:

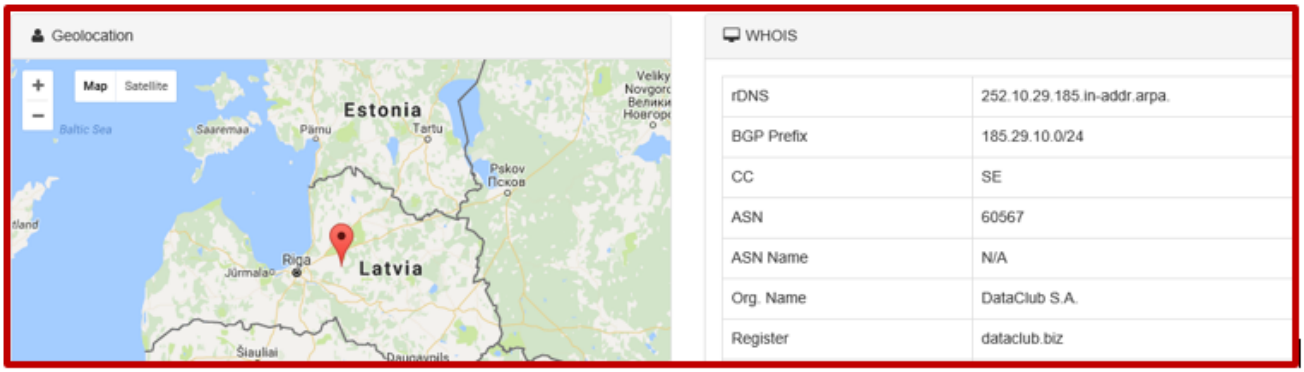
All the communication and analysis were done completely isolated environment without actually allowing malware to communicate actual CNC servers and DNS.



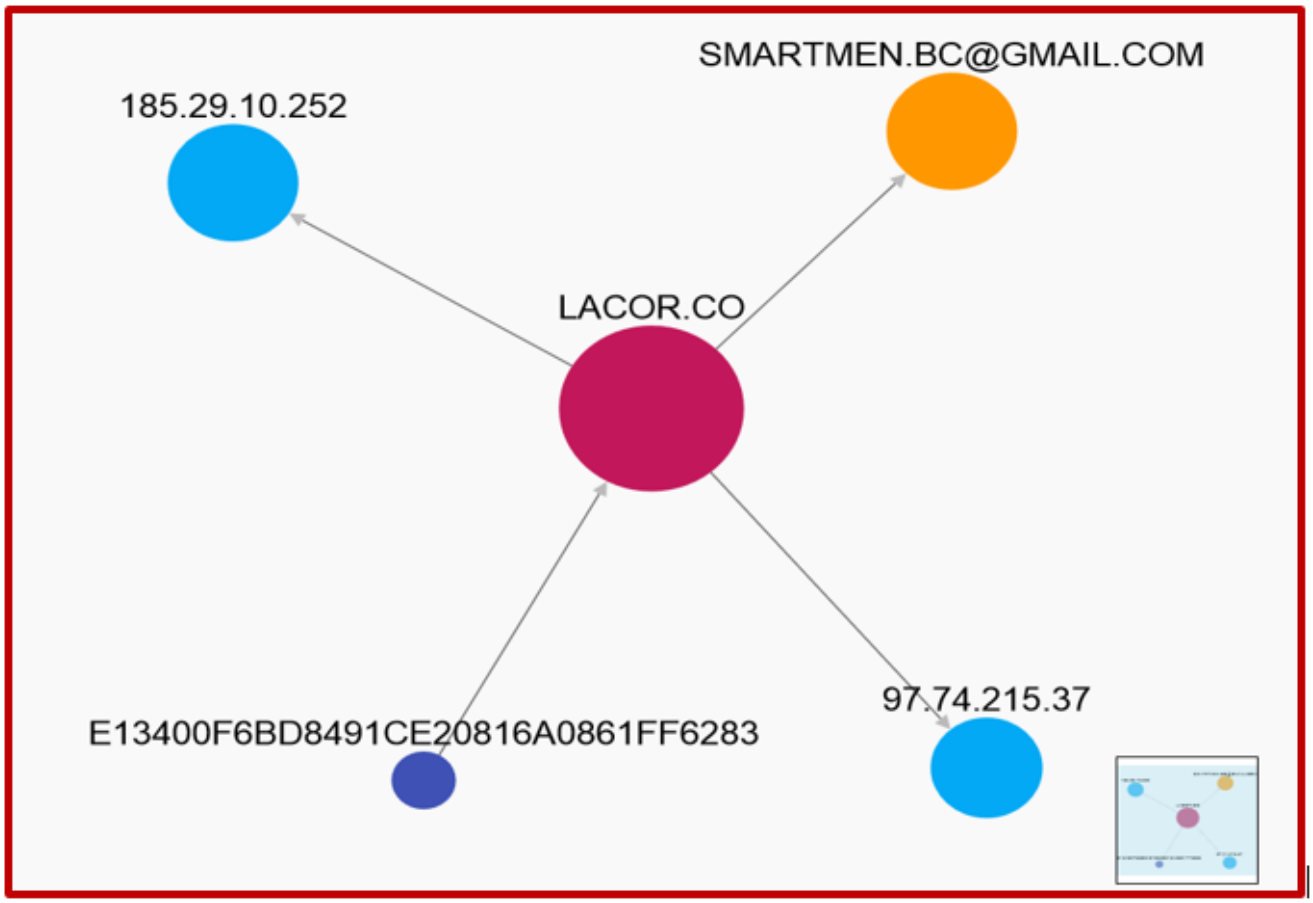
The malware after acquiring enough details such as Username, Machinename, FQDN, and lots of stolen data from the victim machine would then try communicate with the command and control server as we can see in the above stream of packet.

The user agent **“Mozilla/4.08 (Charon; Inferno)”** used has been infamous as it was used in other **Fareit Trojan or PonyLoader**. At this point the Loki exhibits similar kind of behavior though.

The host name seems to be parked at **“185.29.10.252”** which is a Latvia based IP which is malicious.



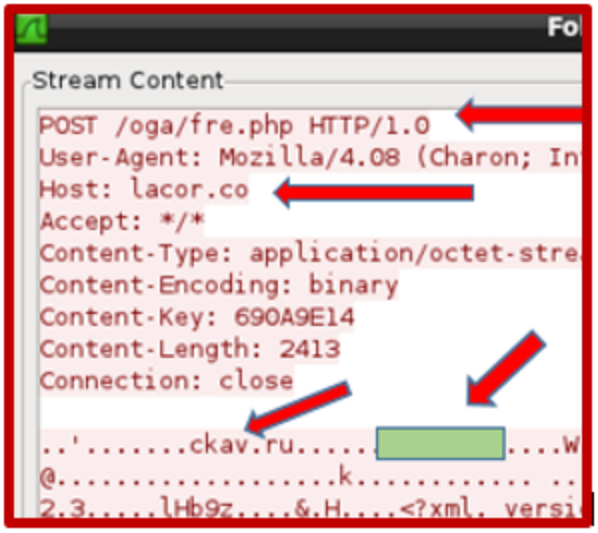
The relation between the IP address, host with hash can be seen below:



Emerging threats have already written rule comprising the malicious user agent:

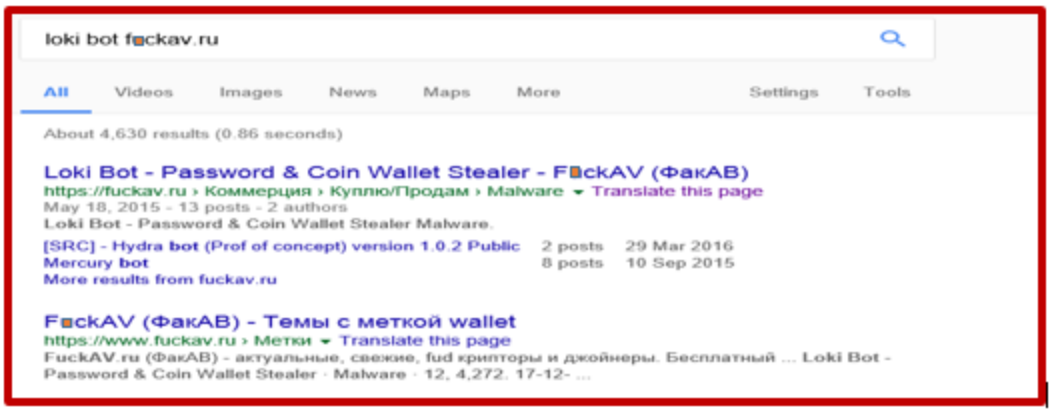
http://doc.emergingthreats.net/bin/view/Main/2021641

Let’s move the spot light to the string **“ckav.ru”** in the stream above shown. From initial glance, we can suspect it might be Russian based malicious website. Even though the domain exists privately, could not find any clear context with the sample we are analyzing.



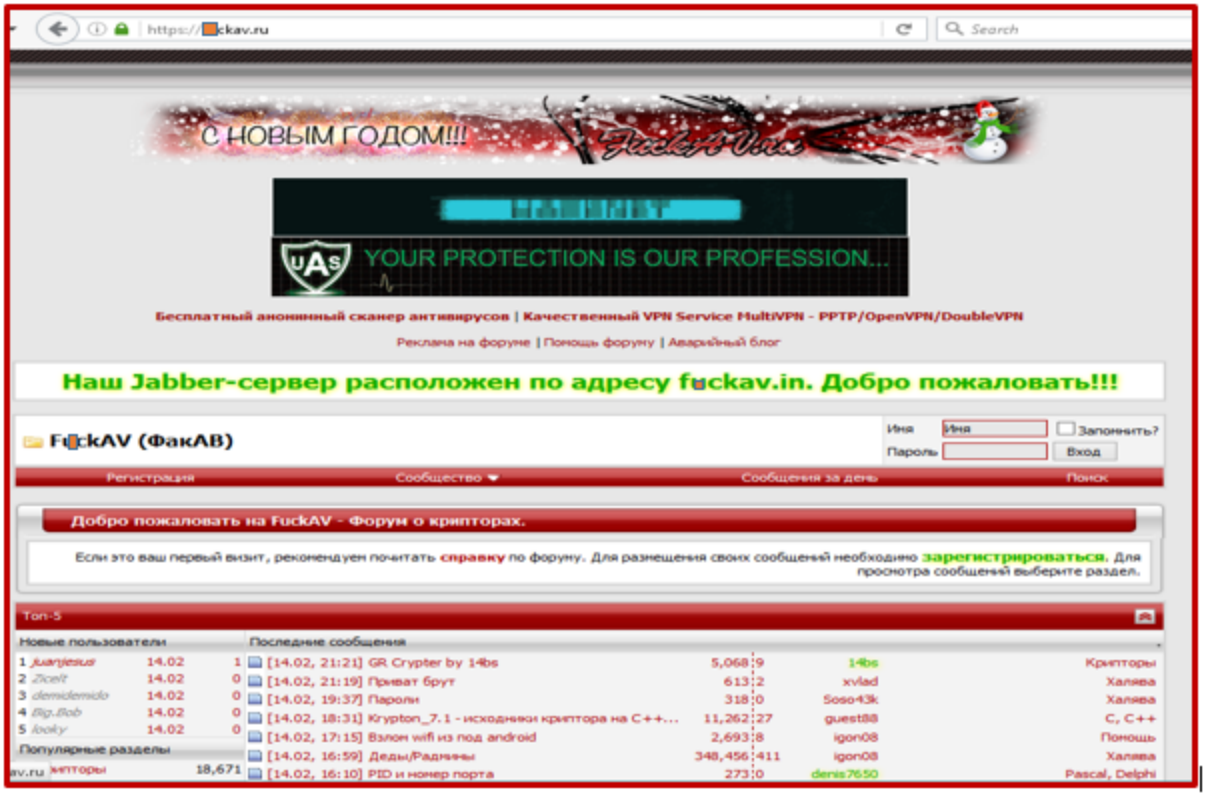
Anticipating if I can get any clue from the unpacked sample strings, I was able to find the missing characters and confirmed it was the URL of a **Russian underground forum**:

When we do a blind search with this URL and suspecting Loki Bot, we will get very promising result:

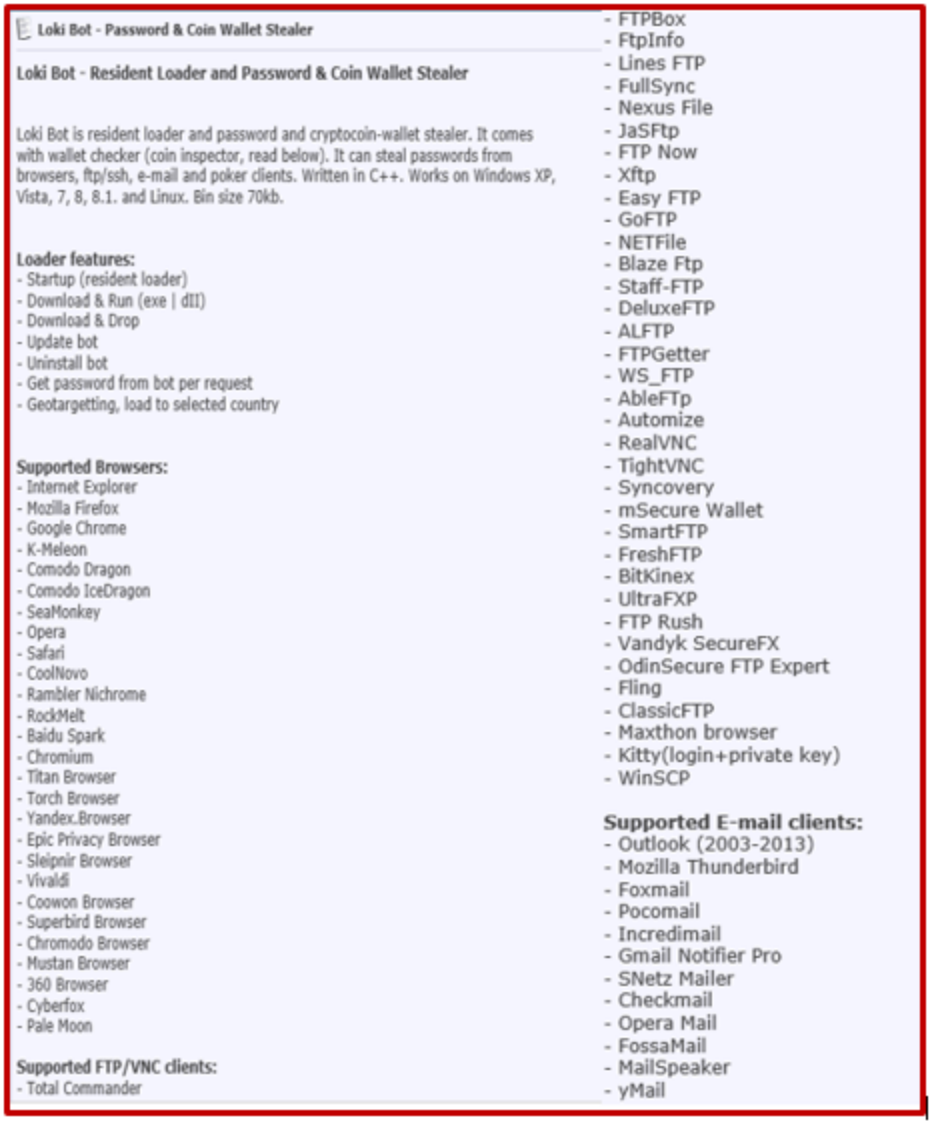


This Bot is being sold in a **Russia underground Forum**. If we see into this website, there are lot other tools which one can register and join the group. After successfully registering we should connect with an already registered account with Jabber and then have to link. Once this is completed anyone can download or share any tools or techniques

> Really Scary!!



Some more deep search gives more result about the Bot. Even advertisement about the same. The features described in this Russian forum matches with our finding earlier:



Loki Bot - Password & Coin Wallet Stealer

Loki Bot - Resident Loader and Password & Coin Wallet Stealer

Loki Bot is resident loader and password and cryptocoin-wallet stealer. It comes with wallet checker (coin inspector, read below). It can steal passwords from browsers, ftp/ssh, e-mail and poker clients. Written in C++. Works on Windows XP, Vista, 7, 8, 8.1. and Linux. Bin size 70kb.

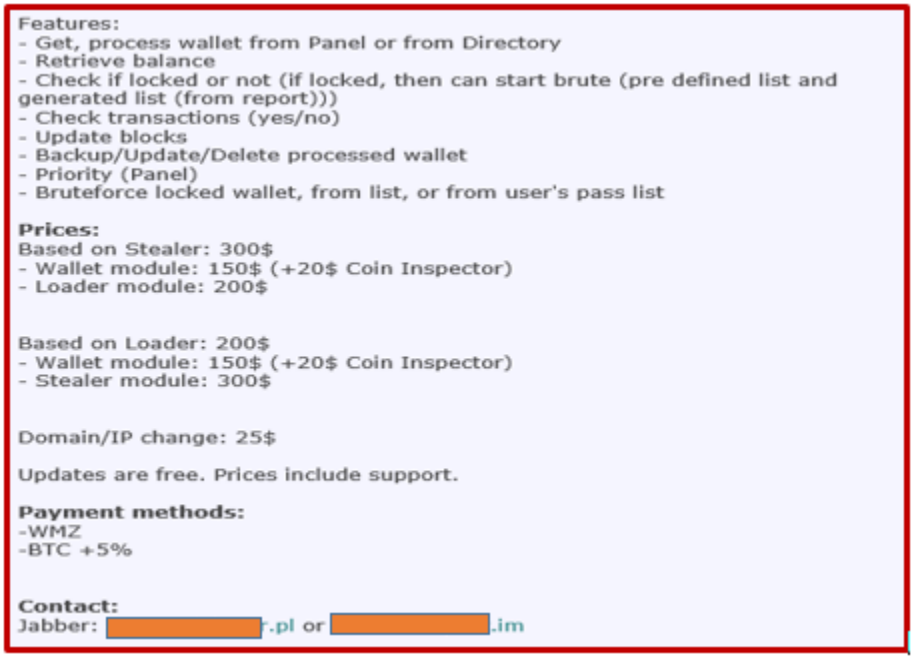
Loader features:
- Startup (resident loader)
- Download & Run (exe | dll)
- Download & Drop
- Update bot
- Uninstall bot
- Get password from bot per request
- Geotargetting, load to selected country

Supported Browsers:
- Internet Explorer
- Mozilla Firefox
- Google Chrome
- K-Meleon
- Comodo Dragon
- Comodo IceDragon
- SeaMonkey
- Opera
- Safari
- CoolNovo
- Rambler Nichrome
- RockMelt
- Baidu Spark
- Chromium
- Titan Browser
- Torch Browser
- Yandex.Browser
- Epic Privacy Browser
- Sleipnir Browser
- Vivaldi
- Coowon Browser
- Superbird Browser
- Chromodo Browser
- Mustan Browser
- 360 Browser
- Cyberfox
- Pale Moon

Supported FTP/VNC clients:
- Total Commander
- FTPBox
- FtpInfo
- Lines FTP
- FullSync
- Nexus File
- JaSFtp
- FTP Now
- Xftp
- Easy FTP
- GoFTP
- NETFile
- Blaze Ftp
- Staff-FTP
- DeluxeFTP
- ALFTP
- FTPGetter
- WS_FTP
- AbleFTp
- Automize
- RealVNC
- TightVNC
- Syncovery
- mSecure Wallet
- SmartFTP
- FreshFTP
- BitKinex
- UltraFXP
- FTP Rush
- Vandyk SecureFX
- OdinSecure FTP Expert
- Fling
- ClassicFTP
- Maxthon browser
- Kitty(login+private key)
- WinSCP

Supported E-mail clients:
- Outlook (2003-2013)
- Mozilla Thunderbird
- Foxmail
- Pocomail
- Incredimail
- Gmail Notifier Pro
- SNetz Mailer
- Checkmail
- Opera Mail
- FossaMail
- MailSpeaker
- yMail

> **Even the features, payment details and contact details are published with it!**



With an embarrassed mind let me conclude..

Are we in a digitally connected world? If the answer is yes, then Obviously Malware is the biggest nightmare for all the entities, irrespective of its geographical location or nature of business. In this Era of Cyber War, Phishing e-mails with targeted macro malware are exponentially circulated by the Offenders across the Globe. Of course, the easiest weakness spotted by offenders is “Human Weakness”. Anyways offenders will stay fingers crossed, whether the end user “allows” himself to respond these malicious attachments or simply “drops” the plan.

As a cautionary note, as we saw in this article, hack advises, hire a hacker, malware, hack tools and anything is now easily available everywhere in the Internet and abundant in the deepest corners of the web. This is very scary right? , so a rigid security posture should be maintained by all the entities to defend these types of threats.

We should be in a position to tell boldly,

> **“If the Offenders are finding new techniques and tactics, so are we”!!!**

Funny Note 🙂

The malware author of the above malware must be a fan of cartoon characters from the below file properties comments:

```
Version info - File Type : Unknown
File Version Info Size=1548 -> 060Ch
Translations : 040904b0   Language : English (U.S.)

CompanyName = Old McDonald's Farm
FileDescription = Billy The Goat
FileVersion = 2.6.0.0
InternalName = ***
LegalCopyright = ©2007-2012 Old McDonald's Farm
LegalTradeMarks = ***
OriginalFilename = ***
ProductVersion = ***
Comments = Billy the goat ate all the autorun.inf files...because Old McDonald was sick of all the viruses and worms on
```

> **Comments =** ***"Billy the goat ate all the autorun.inf files…because Old McDonald was sick of all the viruses and worms on his farm"***

Billy the Goat, who is always hungry ;)

Old McDonald's Farm

References

https://blog.sensecy.com/tag/loki-bot/

https://hackforums.net/showthread.php?tid=5456831

https://www.scmagazine.com/floki-bot–a-zeus-wannabe-with-delusions-of-grandeur/article/569329/

https://digital-forensics.sans.org/blog/2009/11/23/extracting-vb-macros-from-malicious-documents/