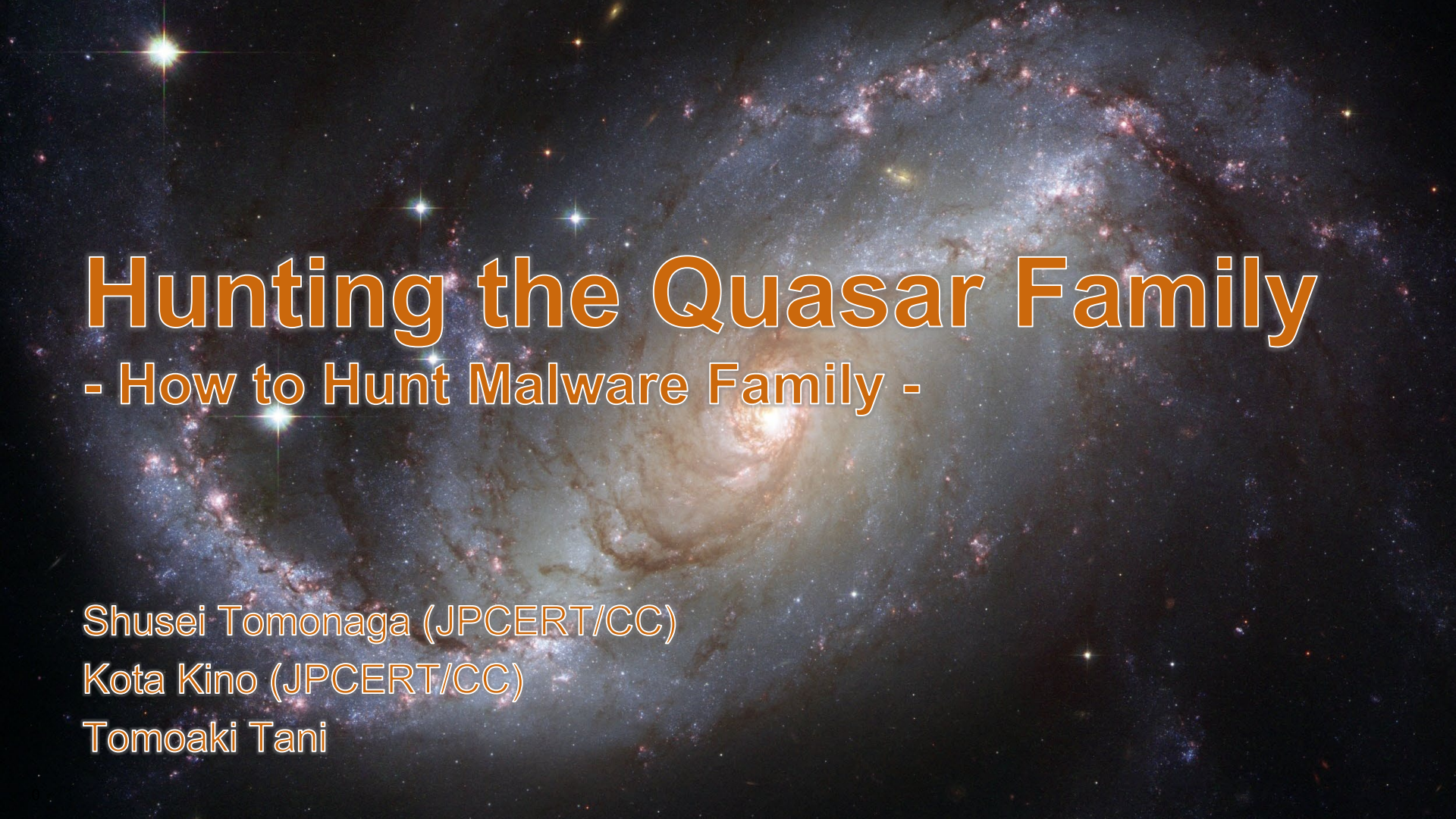


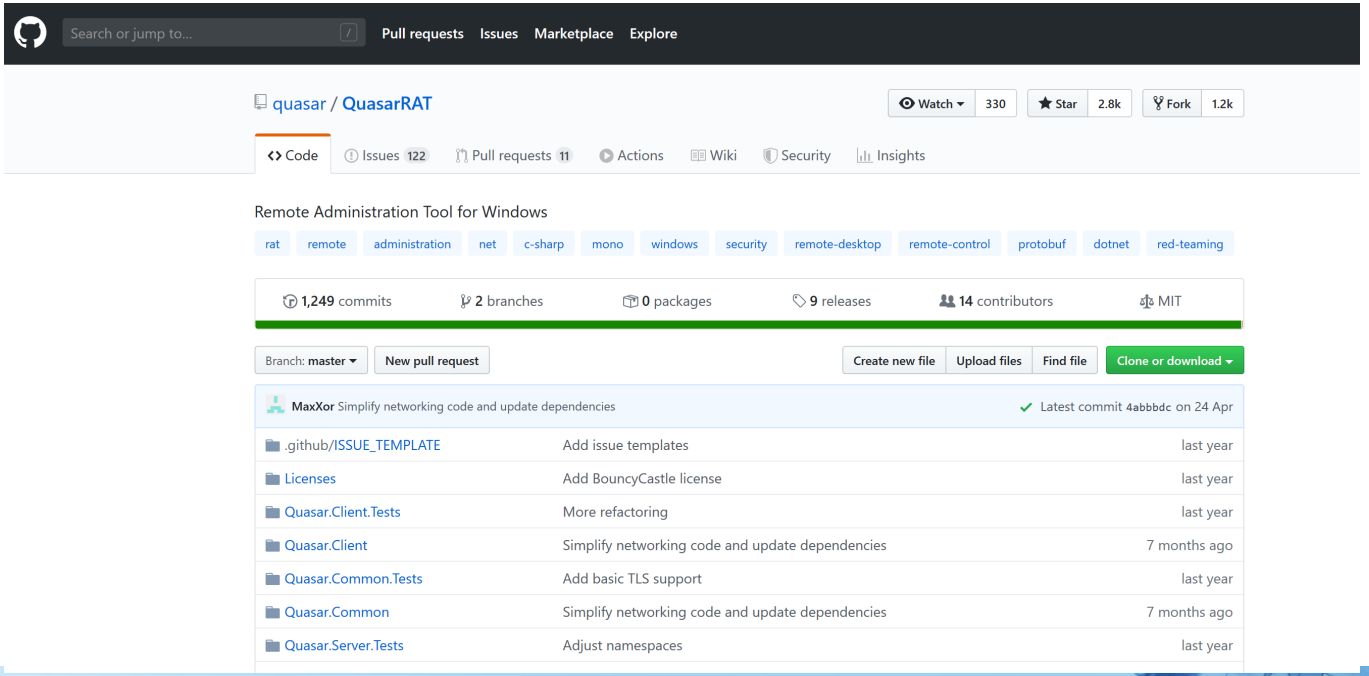
Hunting the Quasar Family

- How to Hunt Malware Family -

Shusei Tomonaga (JPCERT/CC)
Kota Kino (JPCERT/CC)
Tomoaki Tani

QuasarRAT

QuasarRAT is a famous Open-Source **RAT** project on Github.



QuasarRAT Details

Features

- TCP network stream (IPv4 & IPv6 support)
- Fast network serialization (Protocol Buffers)
- Compressed (QuickLZ) & Encrypted (TLS) communication
- UPnP Support
- Task Manager
- File Manager
- Startup Manager
- Remote Desktop
- Remote Shell
- Remote Execution
- System Information
- Registry Editor
- System Power Commands (Restart, Shutdown, Standby)
- Keylogger (Unicode Support)
- Reverse Proxy (SOCKS5)
- Password Recovery (Common Browsers and FTP Clients)
- ... and many more!

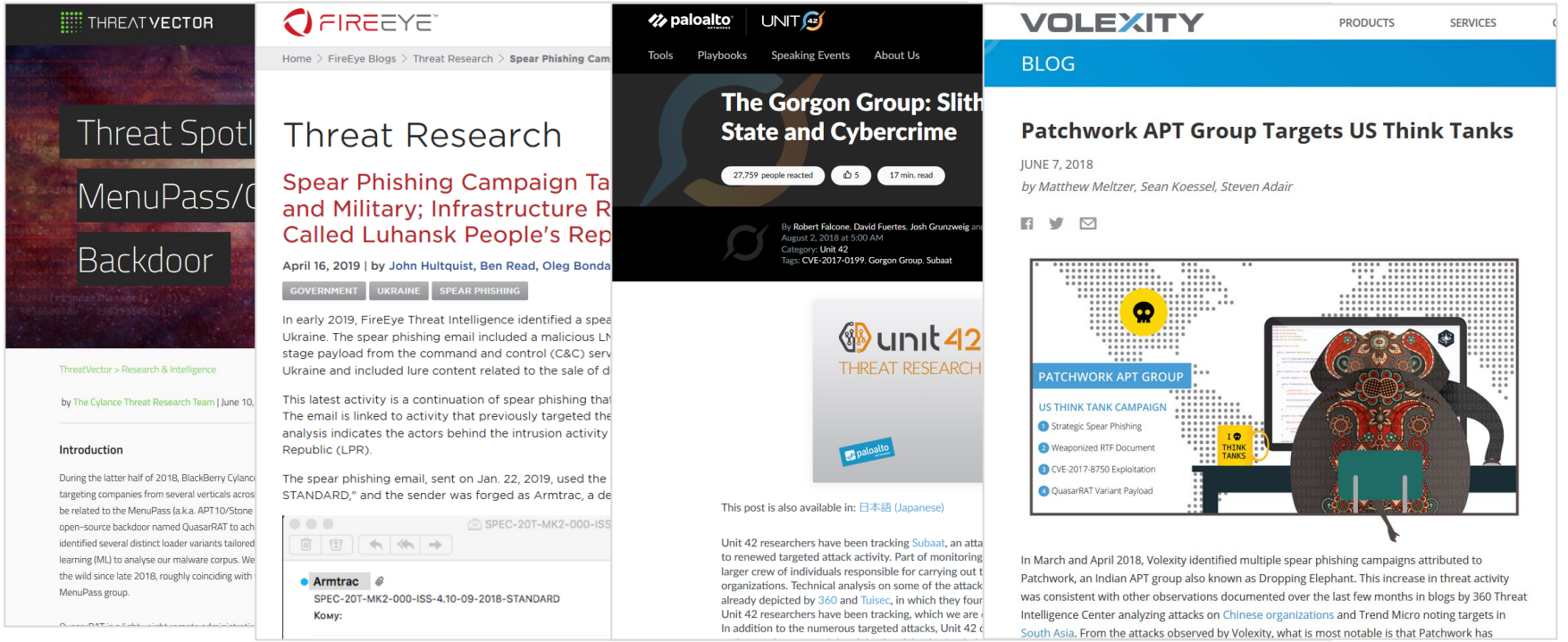
Supported runtimes and operating systems

- .NET Framework 4.5.2 or higher
- Supported operating systems (32- and 64-bit)
  - Windows 10
  - Windows Server 2019
  - Windows Server 2016
  - Windows 8/8.1
  - Windows Server 2012
  - Windows 7
  - Windows Server 2008
  - Windows Vista
- For older systems please use Quasar version 1.3.0

https://github.com/quasar/QuasarRAT

Operations using QuasarRAT

QuasarRAT is used in many attack operations.



Goal of This Presentation

This presentation shares the internals of the QuaserRAT family and C2 hunting methods.

This Presentation Topics

1. QuasarRAT Internals
2. Quasar Family
3. Campaigns using QuasarRAT
4. Hunting Quasar Family C2

1 QuasarRAT Internals

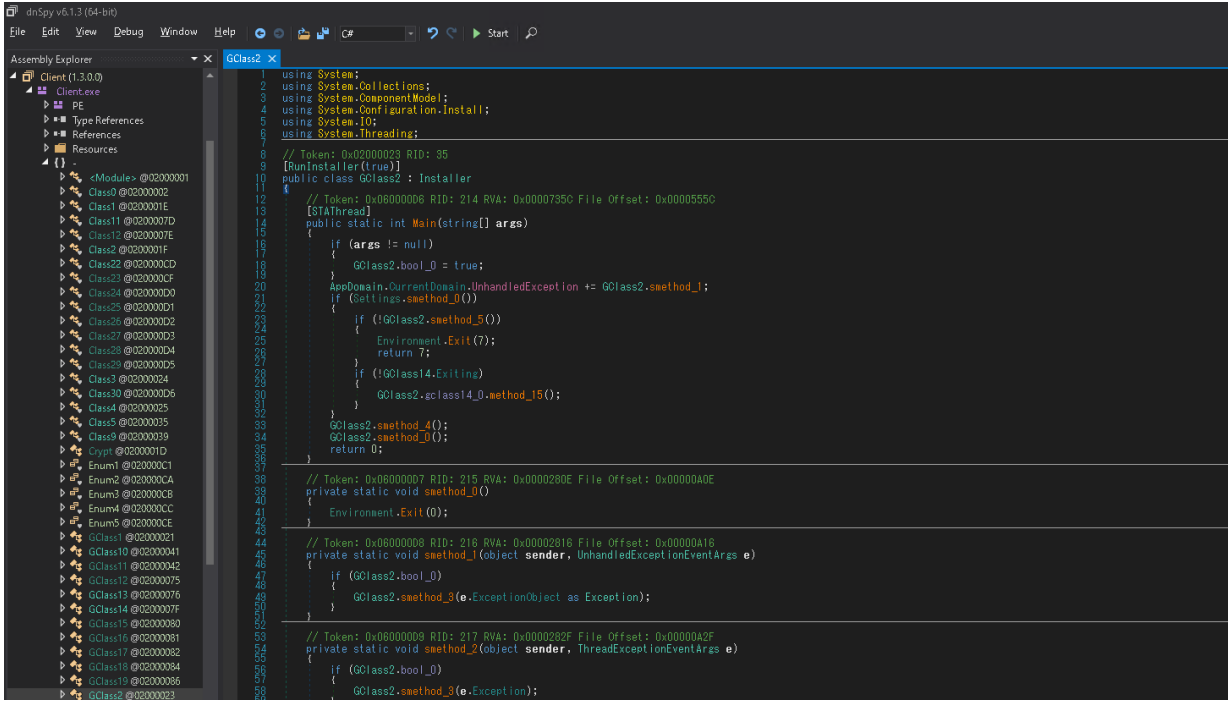
2 Quasar Family

3 Campaigns using QuasarRAT

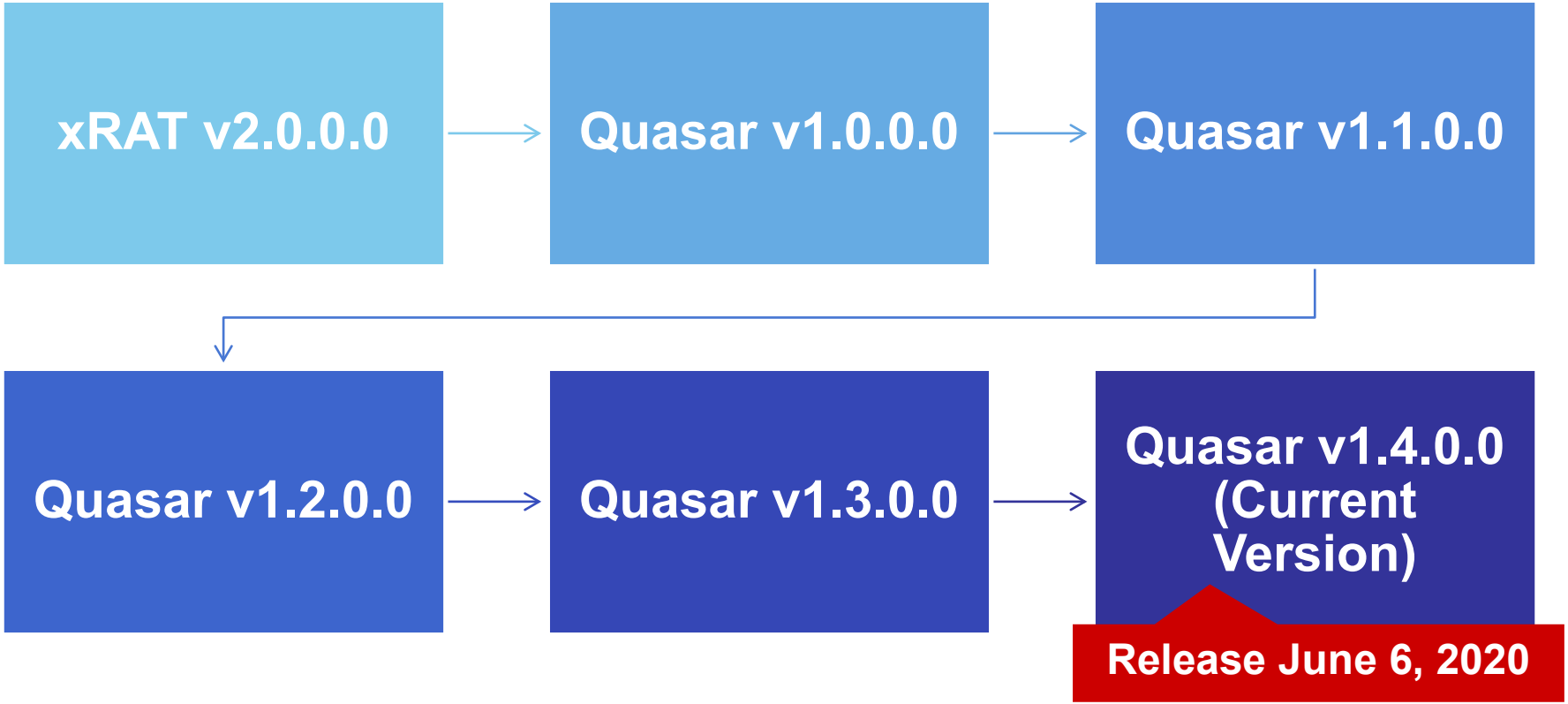
4 Hunting Quasar Family C2

QuasarRAT Internals

QuasarRAT is coded in C#.



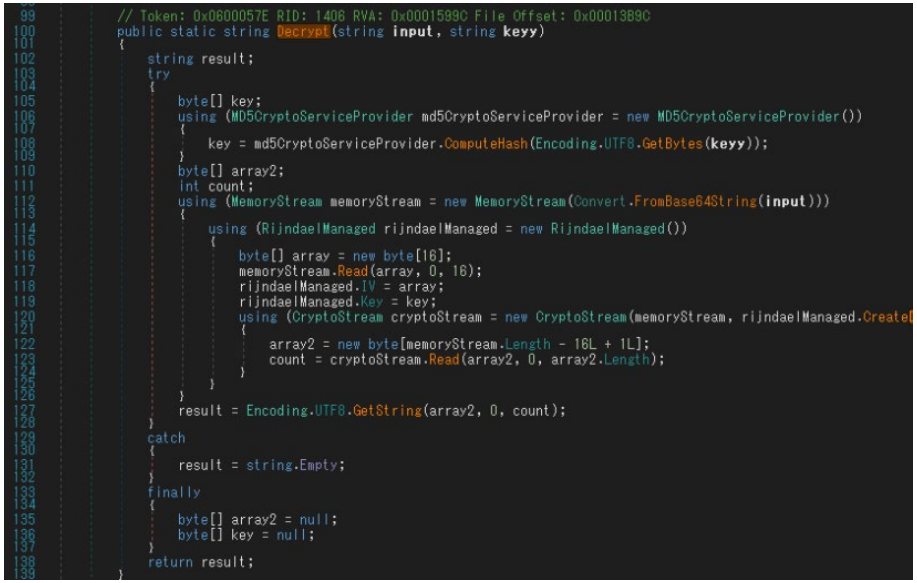
QuasarRAT Version History



Communication

xRAT 2.0.0.0

Size(4byte)	IV(16byte)	AES(mode CBC)



```
// Token: 0x0600057E RID: 1406 RVA: 0x0001599C File Offset: 0x00013B9C
public static string Decrypt(string input, string keyy)
{
    string result;
    try
    {
        byte[] key;
        using (MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider())
        {
            key = md5CryptoServiceProvider.ComputeHash(Encoding.UTF8.GetBytes(keyy));
        }
        byte[] array2;
        int count;
        using (MemoryStream memoryStream = new MemoryStream(Convert.FromBase64String(input)))
        {
            using (RijndaelManaged rijndaelManaged = new RijndaelManaged())
            {
                byte[] array = new byte[16];
                memoryStream.Read(array, 0, 16);
                rijndaelManaged.IV = array;
                rijndaelManaged.Key = key;
                using (CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateD
                {
                    array2 = new byte[memoryStream.Length - 16L + 1L];
                    count = cryptoStream.Read(array2, 0, array2.Length);
                }
            }
        }
        result = Encoding.UTF8.GetString(array2, 0, count);
    }
    catch
    {
        result = string.Empty;
    }
    finally
    {
        byte[] array2 = null;
        byte[] key = null;
    }
    return result;
}
```

Command(1byte)	argv

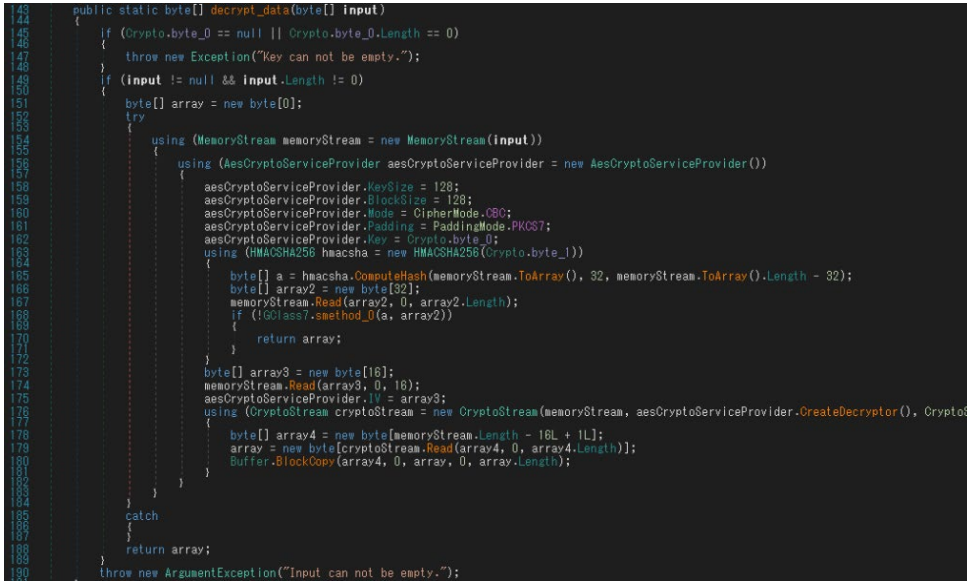
AES_KEY = md5(ENCRYPTIONKEY)

Communication Format

QuasarRAT 1.3.0.0

* QuickLZ compress level3

Size(4byte)	HMAC(32byte)	IV(16byte)	QuickLZ + AES(mode CBC)



```
public static byte[] decrypt_data(byte[] input)
{
    if (Crypto.byte_0 == null || Crypto.byte_0.Length == 0)
    {
        throw new Exception("Key can not be empty.");
    }
    if (input != null && input.Length != 0)
    {
        byte[] array = new byte[0];
        try
        {
            using (MemoryStream memoryStream = new MemoryStream(input))
            {
                using (AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider())
                {
                    aesCryptoServiceProvider.KeySize = 128;
                    aesCryptoServiceProvider.BlockSize = 128;
                    aesCryptoServiceProvider.Mode = CipherMode.CBC;
                    aesCryptoServiceProvider.Padding = PaddingMode.PKCS7;
                    aesCryptoServiceProvider.Key = Crypto.byte_0;
                    using (HMACSHA256 hmacsha = new HMACSHA256(Crypto.byte_1))
                    {
                        byte[] a = hmacsha.ComputeHash(memoryStream.ToArray(), 32, memoryStream.ToArray().Length - 32);
                        byte[] array2 = new byte[32];
                        memoryStream.Read(array2, 0, array2.Length);
                        if (!GClass7.smethod_0(a, array2))
                        {
                            return array;
                        }
                    }
                    byte[] array3 = new byte[16];
                    memoryStream.Read(array3, 0, 16);
                    aesCryptoServiceProvider.IV = array3;
                    using (CryptoStream cryptoStream = new CryptoStream(memoryStream, aesCryptoServiceProvider.CreateDecryptor(), CryptoS
                    {
                        byte[] array4 = new byte[memoryStream.Length - 16L + 1L];
                        array = new byte[cryptoStream.Read(array4, 0, array4.Length)];
                        Buffer.BlockCopy(array4, 0, array, 0, array.Length);
                    }
                }
            }
        }
        catch
        {
        }
        return array;
    }
    throw new ArgumentException("Input can not be empty.");
}
```

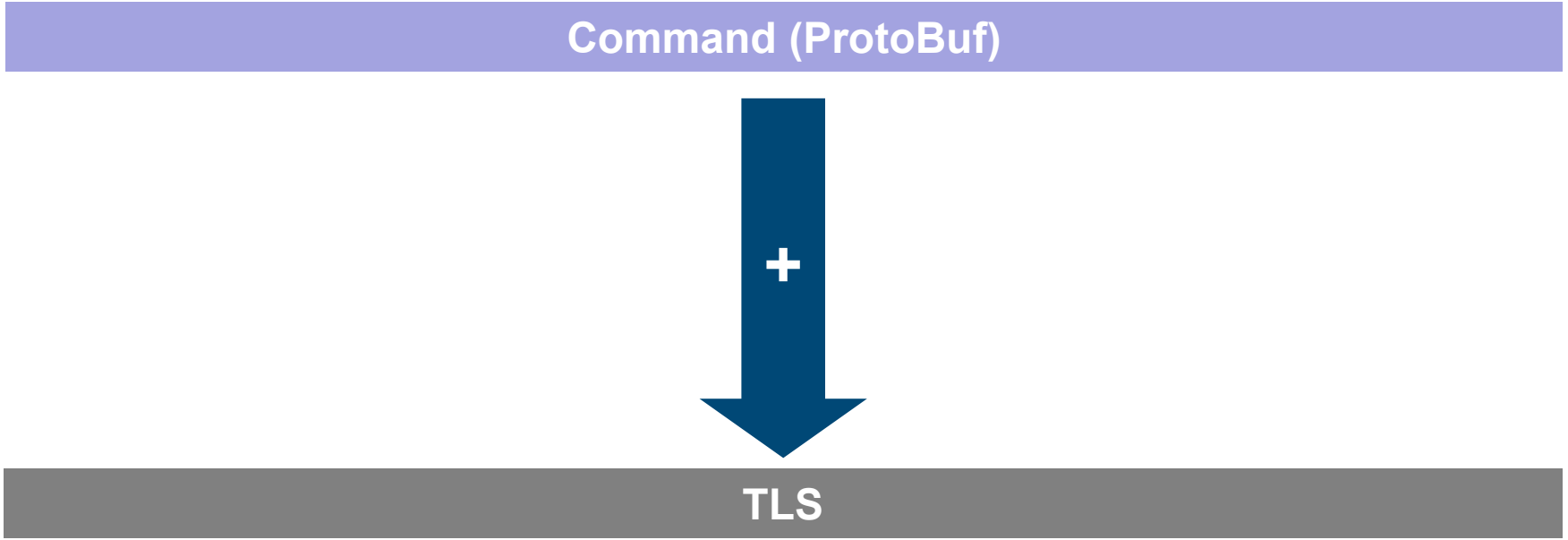
Command(1byte)	argv

AES_KEY = PBKDF2(ENCRYPTIONKEY, salt, 50000).read(16)

Communication Format

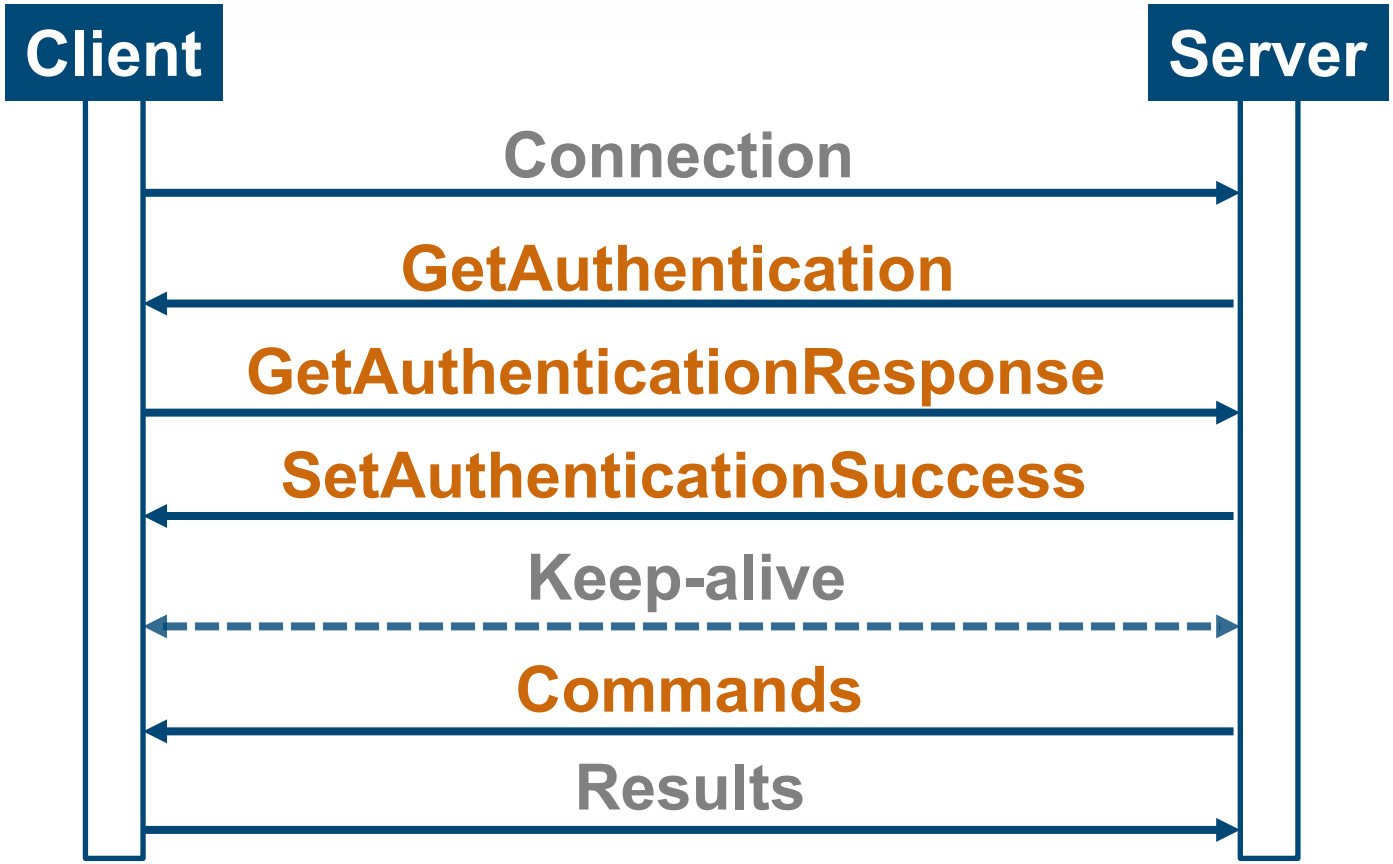
QuasarRAT 1.4.0.0

* Not use AES + QuickLZ



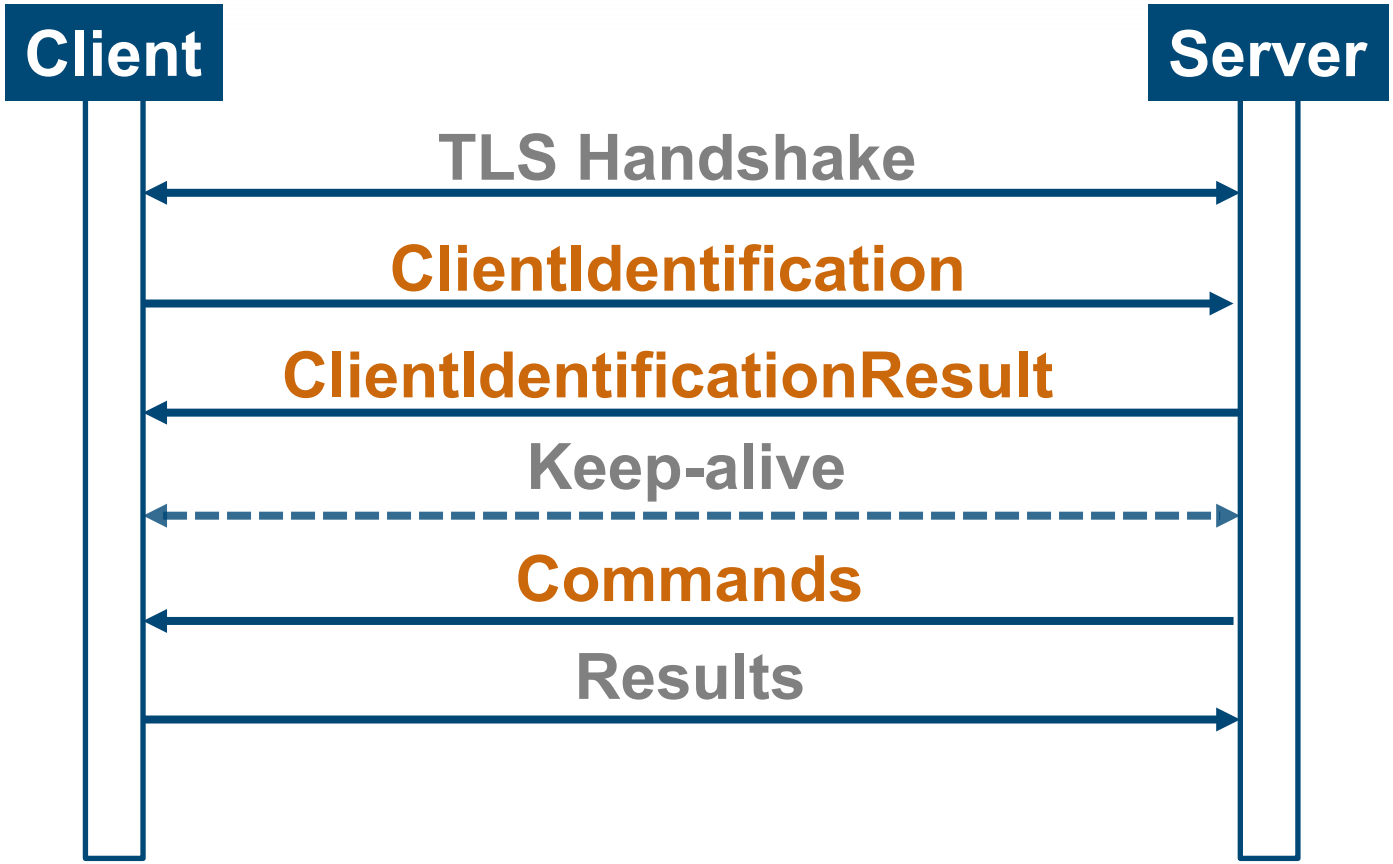
Communication Flow

QuasarRAT 1.3



Communication Flow

QuasarRAT 1.4



Configuration

xRAT

```
// Token: 0x0400019C RID: 412
public static string Version = "MuchfhtzN5zbOIiOyn8I4gEoQIQ3O6NftZEnYCSfGO+jx1tBjsOi7fIC4u/WBoun";

// Token: 0x0400019D RID: 413
public static string Host = "gtIWoOUG8NwiDphZ4DJf24Zi1kSIMvFubqcbiznA7Sw=";

// Token: 0x0400019E RID: 414
public static ushort Port = 1312;

// Token: 0x0400019F RID: 415
public static int Reconnectdelay = 5000;

// Token: 0x040001A0 RID: 416
public static string Password = "y+CxG6V8fhj3D43N5VJeFcAzu9LzsERJCPxyn+tpogNkF8I1lhWSXlpeY6veOwSR215QB5U9nvM9UySDsN

// Token: 0x040001A1 RID: 417
public static string Dir = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);

// Token: 0x040001A2 RID: 418
public static string Subfolder = "dsGnklxnhy9R5I8B2Ag+6gtyNidaCoZ2YDAKZ+QGdws=";

// Token: 0x040001A3 RID: 419
public static string Installname = "lvzieirDlDtVp+X8KOQUvmGI6bE2GQzv//7FBCu92Oc=";

// Token: 0x040001A4 RID: 420
public static bool Install = false;

// Token: 0x040001A5 RID: 421
public static bool Startup = false;

// Token: 0x040001A6 RID: 422
```

Base64

AES_KEY = md5(ENCRYPTIONKEY)

IV(16byte) | **AES(mode CBC)**

Configuration

QuasarRAT

```
// Token: 0x04000008 RID: 8
public static string VERSION = "UvADP/X6YS4Pi/TtR6RCQ49f9HntDsrnWRpHNg4Sp7M9cqQyJryNpoozrQAdD2e39pa5nVYAe1k9B5+dvQGMEw==";

// Token: 0x04000009 RID: 9
public static string HOSTS = "Tqu77dMuK/ZIsz66TL5IQUxE8/LQhIir5uxAqtiTHCs4hvRJd5xp4o035SnBtDLtmdK+sDe2ewmNBkSJWjug8HEoms8v9d6aFtI28DIw4QgEkK5X
// Token: 0x0400000A RID: 10
public static int RECONNECTDELAY = 3000;

// Token: 0x0400000B RID: 11
public static string KEY = "ebdOqS6Q89sUEWPiItdABQ==";

// Token: 0x0400000C RID: 12
public static string AUTHKEY = "SQc2h9nqnECTE5yMFKY1PRF9QHSIbZLcqE0ccuiIHMgbDR8OXD7Fv9XDKcDJAU/mouBB4/KAEpaWUM9KJEJZBQ==";

// Token: 0x0400000D RID: 13
public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;

// Token: 0x0400000E RID: 14
public static string DIRECTORY = Environment.GetFolderPath(Init_config.specialFolder_0);

// Token: 0x0400000F RID: 15
public static string SUBDIRECTORY = "di7ALkOAISIPt5o4bbnsEuESGLSgwUEYBMBMHKzTpR9SUruwW6dmfsS9Z1Xlyrt3I8xpkwaScvpuQA9aFCGCVA==";

// Token: 0x04000010 RID: 16
public static string INSTALLNAME = "/np6aAFTWsNmJIk0C428HV7MCpkYc+gwJURNOYibATKpCO0pqi9qblwFMLXptV/cDEbDjaYbN/Zfs2B8/JSBJQ==";

// Token: 0x04000011 RID: 17
public static bool INSTALL = false;

// Token: 0x04000012 RID: 18
public static bool STARTUP = false;

// Token: 0x04000013 RID: 19
public static string MUTEX = "X1T4srVtqtsrjEJxWjPtQWeVzcIi57988AaQTQ8gIRtQQgFDWr1vQ95E123ZJEEUBuUNTBLCXtkMavYPFBq9LQMVUTWE/fa5pptXFSQRxwA=";
```

Base64

AES_KEY = PBKDF2(ENCRYPTIONKEY, salt, 50000).read(16)

HMAC(32byte)	IV(16byte)	AES(mode CBC)

Configuration

Config Value	
VERSION	ENABLEUACESCALATION (Only xRAT)
HOSTS	ENABLELOGGER
PORT (Only xRAT)	ENCRYPTIONKEY
RECONNECTDELY	TAG 1.3
KEY	LOGDIRECTORY 1.3
AUTHKEY 1.3	SERVERSIGNATURE 1.4
DIRECTORY	SERVERCERTIFICATESTR 1.4
SUBDIRECTORY	SERVERCERTIFICATE 1.4
INSTALLNAME	HIDELOGDIRECTORY 1.3
INSTALL	HIDELOGSUBDIRECTORY 1.3
STARTUP	INSTALLPATH 1.4
MUTEX	LOGSPATH 1.4
STARTUPKEY	UNATTENDEDMODE 1.4
HIDEFILE	

Commands

QuasarRAT 1.3

QuasarRAT

```
public class Commands
{
    // Token: 0x06000377 RID: 887
    public static Type[] command()
    {
        return new Type[]
        {
            typeof(GetAuthentication),
            typeof(DoClientDisconnect),
            typeof(DoClientReconnect),
            typeof(DoClientUninstall),
            typeof(DoWebcamStop),
            typeof(DoAskElevate),
            typeof(DoDownloadAndExecute),
            typeof(DoUploadAndExecute),
            typeof(GetDesktop),
            typeof(GetProcesses),
            typeof(DoProcessKill),
            typeof(DoProcessStart),
            typeof(GetDrives),
            typeof(GetDirectory),
            typeof(DoDownloadFile),
            typeof(DoMouseEvent),
            typeof(DoKeyboardEvent),
            typeof(GetSystemInfo),
            typeof(DoVisitWebsite),
            typeof(DoShowMessageBox),
            typeof(DoClientUpdate),
            typeof(GetMonitors),
            typeof(GetWebcams),
            typeof(GetWebcam),
            typeof(DoShellExecute),
            typeof(DoPathRename),
            typeof(DoPathDelete),
            typeof(DoShutdownAction),
            typeof(GetStartupItems),
            typeof(DoStartupItemAdd),
            typeof(DoStartupItemRemove),
            typeof(DoDownloadFileCancel),
            typeof(GetKeyloggerLogs),
            typeof(DoUploadFile),
            typeof(GetPasswords),
            typeof(DoLoadRegistryKey),
```

Define the commands with typeof.

A structure is defined in the executable file.

```
COMMAND_SET = {
    1: "GetConnectionsResponse",
    5: "ReverseProxyDisconnect",
    7: "ReverseProxyData",
    8: "ReverseProxyConnectResponse",
    13: "AddressFamily",
    15: "ReverseProxyConnect",
    16: "GetChangeRegistryValueResponse",
    19: "GetRenameRegistryValueResponse",
    20: "GetDeleteRegistryValueResponse",
    21: "GetCreateRegistryValueResponse",
    22: "GetRenameRegistryKeyResponse",
    23: "GetDeleteRegistryKeyResponse",
    24: "GetCreateRegistryKeyResponse",
    27: "GetRegistryKeysResponse",
    29: "GetPasswordsResponse",
    31: "GetKeyloggerLogsResponse",
    32: "GetStartupItemsResponse",
    33: "DoShellExecuteResponse",
    34: "GetWebcamResponse",
    35: "GetWebcamsResponse",
    42: "GetMonitorsResponse",
    43: "GetSystemInfoResponse",
    44: "DoDownloadFileResponse",
    45: "GetDirectoryResponse",
    47: "GetDrivesResponse",
    48: "GetProcessesResponse",
    50: "GetDesktopResponse",
    51: "SetUserStatus",
    53: "SetStatusFileManager",
    54: "SetStatus",
```

Commands

QuasarRAT 1.3

Decoded QuasarRAT traffic

```
test@debian:~$ python3 quasarrat_client.py -s
coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj5ad2ROU
[+] Thread count 1.
[+] Connect port 210.144.121.100:4782.
[+] Get data size: 68
[+] Hash check OK.
[+] Command: GetAuthentication
[+] Decoded data: b'j'
[+] Send data.
[+] Send data size: 228
[+] Get data size: 68
[+] Hash check OK.
[+] Command: SetAuthenticationSuccess
[+] Decoded data: b';'
[+] Get data size: 84
[+] Hash check OK.
[+] Command: DoShellExecute
[+] Decoded data: b'P\x07\x06whoami'
```

Command number

```
127  COMMAND_SET = {
128      1: "GetConnectionsResponse",
129      5: "ReverseProxyDisconnect",
130      7: "ReverseProxyData",
131      8: "ReverseProxyConnectResponse",
132     13: "AddressFamily",
133     15: "ReverseProxyConnect",
134     16: "GetChangeRegistryValueResponse",
135     19: "GetRenameRegistryValueResponse",
136     20: "GetDeleteRegistryValueResponse",
137     21: "GetCreateRegistryValueResponse",
138     22: "GetRenameRegistryKeyResponse",
139     23: "GetDeleteRegistryKeyResponse",
140     24: "GetCreateRegistryKeyResponse",
141     27: "GetRegistryKeysResponse",
142     29: "GetPasswordsResponse",
143     31: "GetKeyloggerLogsResponse",
144     32: "GetStartupItemsResponse",
145     33: "DoShellExecuteResponse",
146     34: "GetWebcamResponse",
147     35: "GetWebcamsResponse",
148     42: "GetMonitorsResponse",
149     43: "GetSystemInfoResponse",
150     44: "DoDownloadFileResponse",
151     45: "GetDirectoryResponse",
152     47: "GetDrivesResponse",
153     48: "GetProcessesResponse",
154     50: "GetDesktopResponse",
155     51: "SetUserStatus",
156     53: "SetStatusFileManager",
157     54: "SetStatus",
```

Command Set

Command Set						
DoAskElevate	DoDownloadAndExecute	DoRenameRegistryKey	DoWebcamStop	GetDesktopResponse	GetProcesses	GetWebcams
DoChangeRegistryValue	DoDownloadFile	DoRenameRegistryValue	GetAuthentication	GetDirectory	GetProcessesResponse	GetWebcamsResponse
DoClientDisconnect	DoDownloadFileCancel	DoShellExecute	GetAuthenticationResponse	GetDirectoryResponse	GetRegistryKeysResponse	ReverseProxyConnect
DoClientReconnect	DoDownloadFileResponse	DoShellExecuteResponse	GetChangeRegistryValueResponse	GetDrives	GetRenameRegistryKeyResponse	ReverseProxyConnectResponse
DoClientUninstall	DoKeyboardEvent	DoShowMessageBox	GetConnections	GetDrivesResponse	GetRenameRegistryValueResponse	ReverseProxyData
DoClientUpdate	DoLoadRegistryKey	DoShutdownAction	GetConnectionsResponse	GetKeyloggerLogs	GetStartupItems	ReverseProxyDisconnect
DoCloseConnection	DoMouseEvent	DoStartupItemAdd	GetCreateRegistryKeyResponse	GetKeyloggerLogsResponse	GetStartupItemsResponse	SetAuthenticationSuccess
DoCreateRegistryKey	DoPathDelete	DoStartupItemRemove	GetCreateRegistryValueResponse	GetMonitors	GetSystemInfo	SetStatus
DoCreateRegistryValue	DoPathRename	DoUploadAndExecute	GetDeleteRegistryKeyResponse	GetMonitorsResponse	GetSystemInfoResponse	SetStatusFileManager
DoDeleteRegistryKey	DoProcessKill	DoUploadFile	GetDeleteRegistryValueResponse	GetPasswords	GetWebcam	SetUserStatus
DoDeleteRegistryValue	DoProcessStart	DoVisitWebsite	GetDesktop	GetPasswordsResponse	GetWebcamResponse	

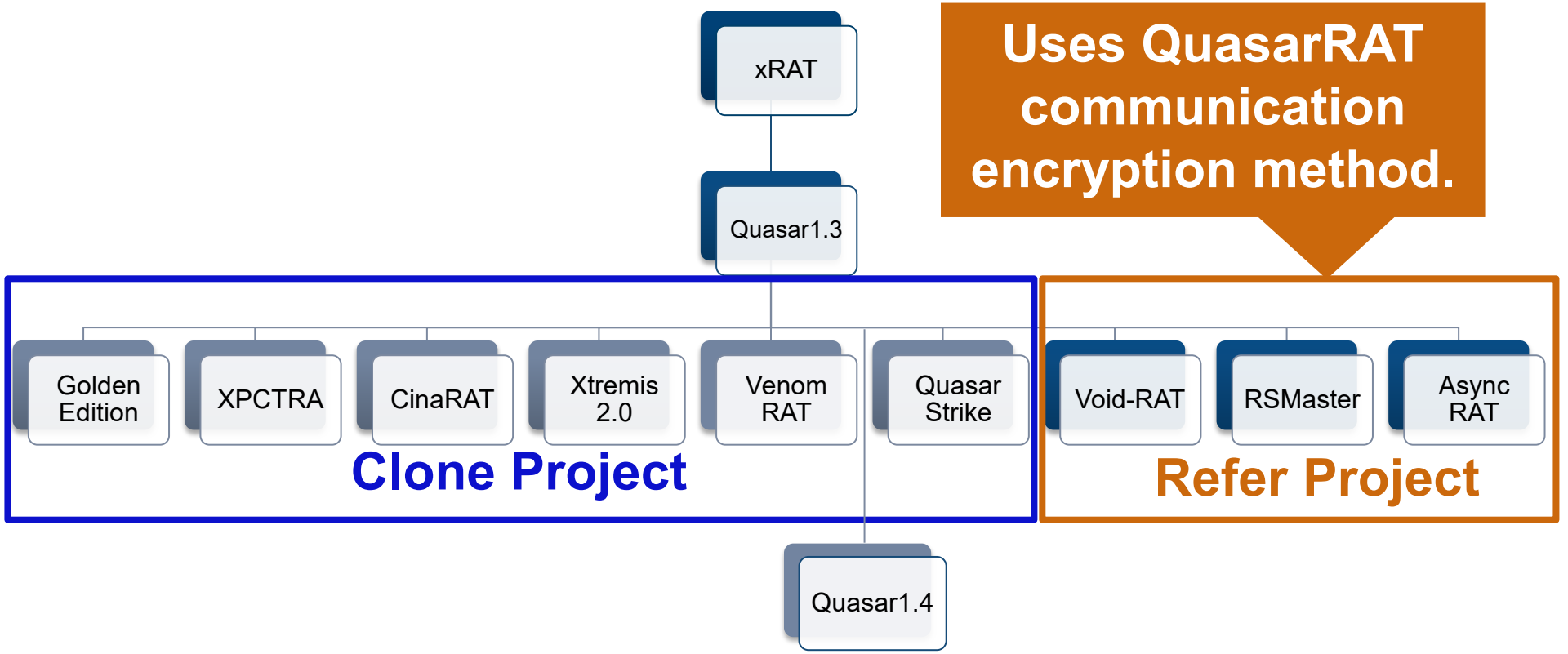
1 QuasarRAT Internals

2 Quasar Family

3 Campaigns using QuasarRAT

4 Hunting Quasar Family C2

Quasar Family



Quasar Family

Malware name	Source code	Settings	Communication	In the wild
Quasar	https://github.com/quasar/QuasarRAT	-	-	Yes
Golden Edition	not published	Original	Original	Yes
XPCTRA	not published	Custom	Original	Yes
CinaRAT	https://github.com/wearelegal/CinaRAT	Original	Original	Yes
Xtremis 2.0	https://github.com/pavitra14/Xtremis-V2.0	Original	Original	No
QuasarStrike	https://github.com/Q-Strike/QuasarStrike	Original	Original	No
VenomRAT	not published	Original	Original	No
RSMaster	https://github.com/Netskyes/rsmaster	Custom	Original	No
Void-RAT	https://github.com/KadeDev/Void-RAT	Custom	Original	Yes
AsyncRAT	https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp	Custom	Original	Yes

XPCTRA

XPCTRA is the banking malware in Brazil.

XPCTRA Config

```
public static class Settings
{
  public static string VERSION = "5.0";
  public static string HOSTS = "coca.cheddarmcmelt.top:8799;";
  public static int RECONNECTDELAY = 500;
  public static object[] HOSTS2 = new object[2]
  {
    (object) "coca.cheddarmcmelt.top",
    (object) 222
  };
  public static string PhpHttps = "http://fritas.cheddarmcmelt.top/master/PhpTrafico.php";
  public static string PhpSeParador = "http://fritas.cheddarmcmelt.top/master/Controle.php";
  public static bool ConectadoMercadobiticoin = true;
  public static bool FormPin = false;
  public static bool ConectandoBlockChain = true;
  public static bool FormBlockchain = false;
  public static bool ConectandoNeteller = true;
  public static bool FormNetller = false;
  public static bool Perfecmoney = true;
  public static string LinkHtmlSpam = "http://fritas.cheddarmcmelt.top/master/conf/Html.txt";
  public static string KEY = "1WvgEMPjdwfqIMeM9MclyQ==";
  public static string AUTHKEY = "NcFtjbDOcsw7Evd3coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj5ad2RO
  public static Environment.SpecialFolder SPECIALFOLDER = Environment.SpecialFolder.Applicatio
  public static string DIRECTORY = Environment.GetFolderPath(Settings.SPECIALFOLDER);
  public static string MUTEX = "123AKs82kA,ylAo2kAlUS2kYkala!";
  public static string TAG = "Infectado";

  public static bool Initialize()
  {
    return true;
  }
```

QuasarRAT Config

```
    public static class Settings
    {
#if DEBUG
        public static string VERSION = System.Windows.Forms.Application.ProductVersion;
        public static string HOSTS = "localhost:4782;";
        public static int RECONNECTDELAY = 500;
        public static string KEY = "1WvgEMPjdwfqIMeM9MclyQ==";
        public static string AUTHKEY = "NcFtjbDOcsw7Evd3coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj
        public static Environment.SpecialFolder SPECIALFOLDER = Environment.SpecialFolder.Appl
        public static string DIRECTORY = Environment.GetFolderPath(SPECIALFOLDER);
        public static string SUBDIRECTORY = "Test";
        public static string INSTALLNAME = "test.exe";
        public static bool INSTALL = false;
        public static bool STARTUP = false;
        public static string MUTEX = "123AKs82kA,ylAo2kAlUS2kYkala!";
        public static string STARTUPKEY = "Test key";
        public static bool HIDEFILE = false;
        public static bool ENABLELOGGER = false;
        public static string TAG = "DEBUG";
        public static string LOGDIRECTORYNAME = "Logs";
        public static bool HIDELOGDIRECTORY = false;
        public static bool HIDEINSTALLSUBDIRECTORY = false;

        public static bool Initialize()
        {
            FixDirectory();
            return true;
        }
```

XPCTRA

XPCTRA is the banking malware in Brazil.

XPCTRA Commands

```
namespace VERMELHO265MONARCA.Core.Packets
{
  public class PacketRegistery
  {
    public static Type[] GetPacketTypes()
    {
      return new Type[76]
      {
        typeof (GetAuthentication),
        typeof (DoClientDisconnect),
        typeof (DoClientReconnect),
        typeof (DoClientUninstall),
        typeof (DoWebcamStop),
        typeof (DoAskElevate),
        typeof (DoDownloadAndExecute),
        typeof (DoUploadAndExecute),
        typeof (GetDesktop),
        typeof (GetProcesses),
        typeof (DoProcessKill),
        typeof (DoProcessStart),
        typeof (GetDrives),
        typeof (GetDirectory),
        typeof (DoDownloadFile),
        typeof (DoMouseEvent),
        typeof (DoKeyboardEvent),
        typeof (GetSystemInfo),
        typeof (DoVisitWebsite),
        typeof (DoShowMessageBox),
        typeof (DoClientUpdate),
        typeof (GetMonitors),
        typeof (GetWebcams),
        typeof (GetWebcam),
        typeof (DoShellExecute),
        typeof (DoPathRename),
```

Added Commands

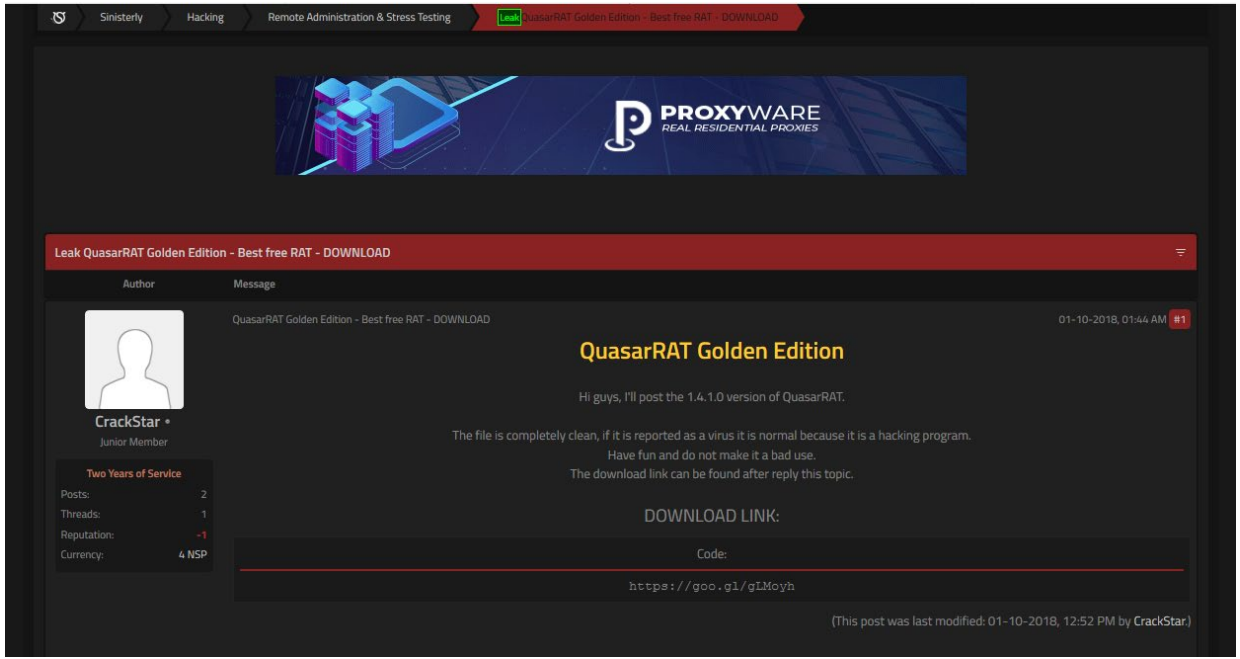
```
typeof (GetChangeRegistryValueResponse),
typeof (FAROFA785MANOBRA),
typeof (FAROFA785MANOBRAResponse),
typeof (ANGUSTIADO349MONITORIA),
typeof (ReverseProxyDisconnect),
typeof (GetConnectionsResponse)
```

QuasarRAT Commands

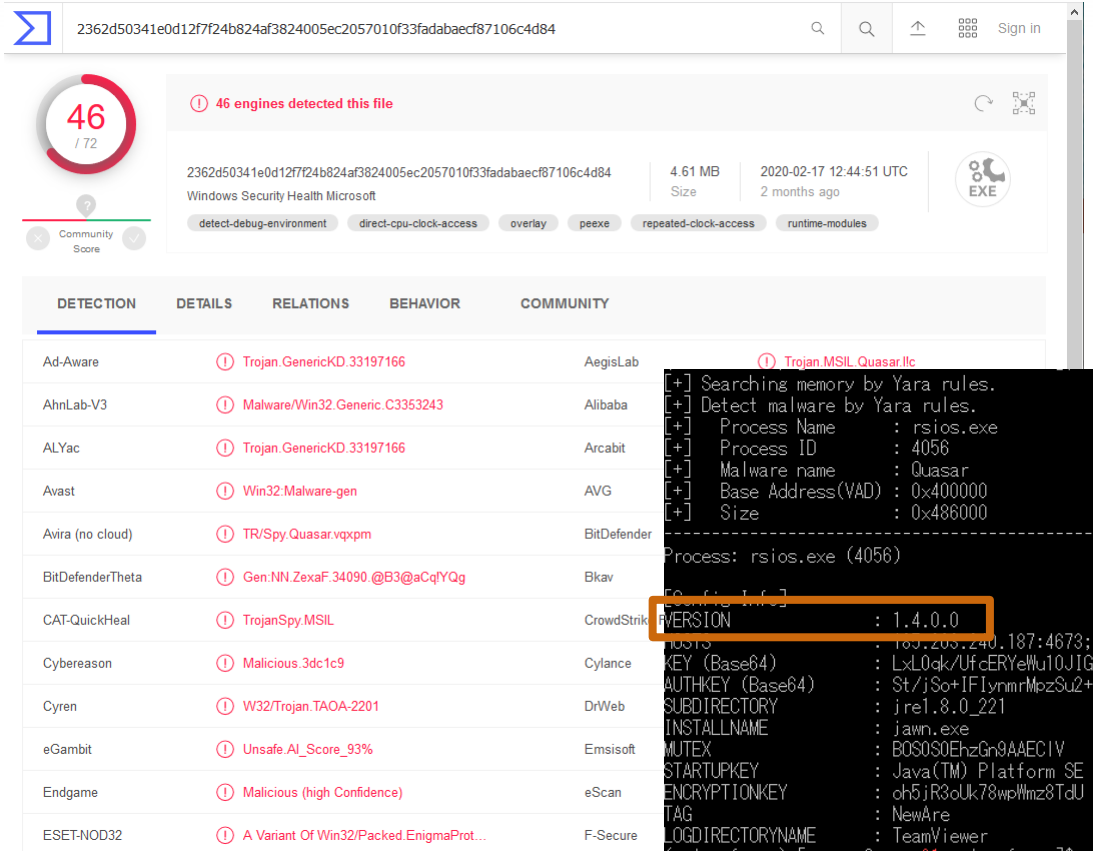
```
namespace xClient.Core.Packets
{
    public class PacketRegistery
    {
        public static Type[] GetPacketTypes()
        {
            return new Type[]
            {
                typeof (Packets.ServerPackets.DoPathRename),
                typeof (Packets.ServerPackets.DoPathDelete),
                typeof (Packets.ServerPackets.DoShutdownAction),
```

Golden Edition

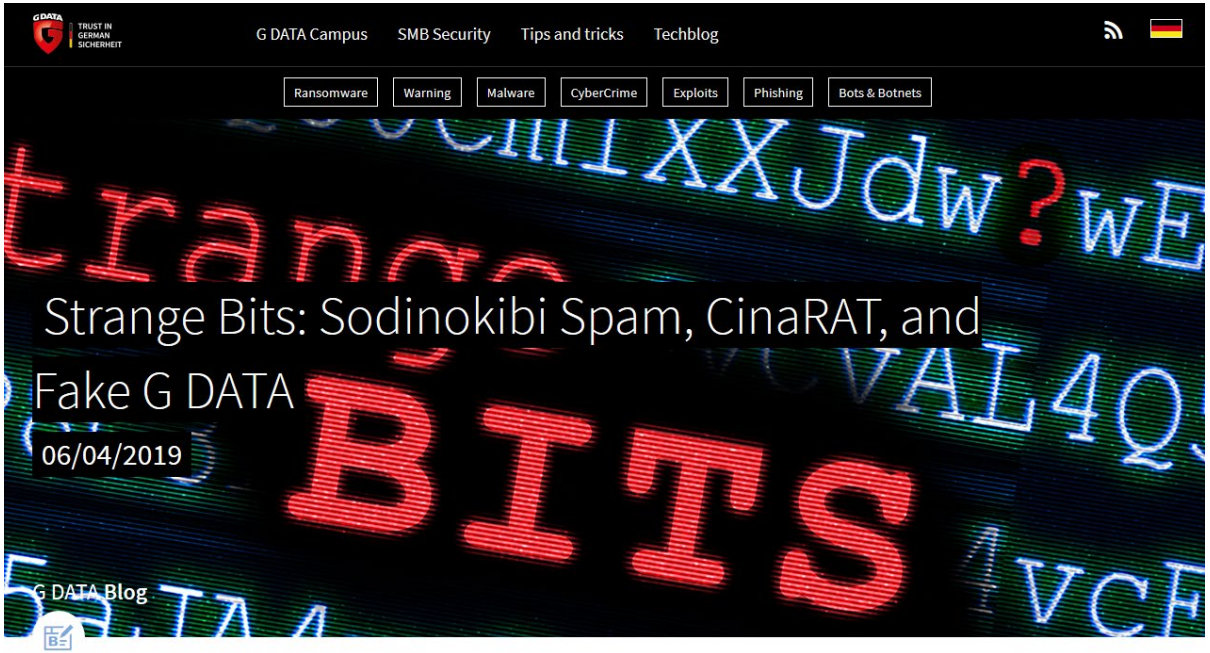
Golden Edition is a version that uses the original source code as is.



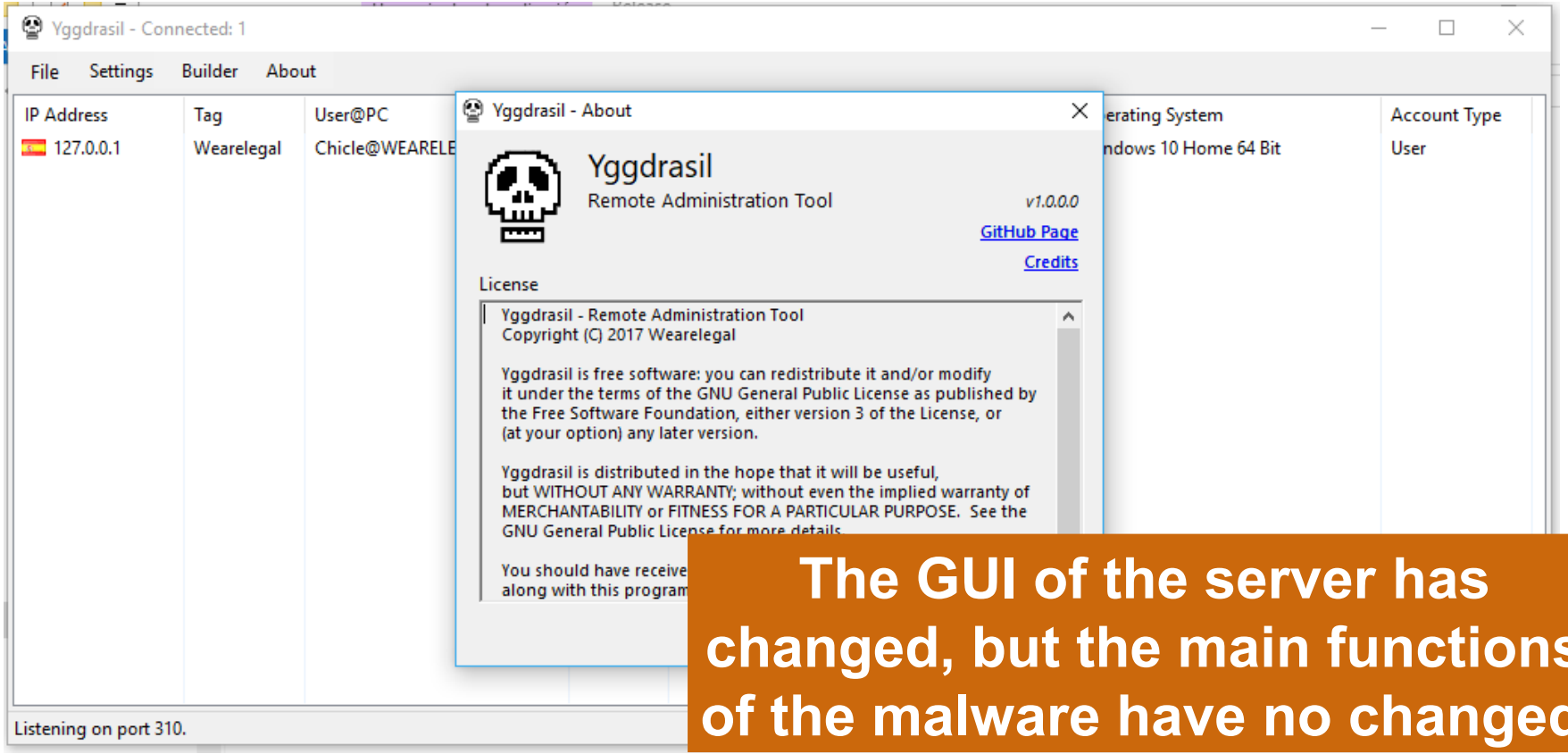
Golden Edition



CinaRAT



CinaRAT



The GUI of the server has changed, but the main functions of the malware have no changed.

VenomRAT

VenomRAT is a RAT sold for $150 per month.

VenomRAT

VenomRAT has some additional unique features

VenomRAT Commands

```
public static Type[] GetPacketTypes()
{
    return new Type[]
    {
        typeof(GetAuthentication),
        typeof(DoClientDisconnect),
        typeof(DoClientReconnect),
        typeof(DoClientUninstall),
        typeof(DoWebcamStop),
        typeof(DoAskElevate),
        typeof(DoRemoveRdp),
        typeof(DoRemoveVnc),
        typeof(DoDownloadAndE
        typeof(DoInstallVNC),
        typeof(DoInstallRDP),
        typeof(DoStealer),
        typeof(DoUploadAndExe
        typeof(GetDesktop),
        typeof(GetProcesses),
        typeof(DoProcessKill),
        typeof(DoProcessStart
        typeof(GetDrives),
        typeof(GetDirectory),
        typeof(DoDownloadFile
        typeof(DoMouseEvent),
        typeof(DoKeyboardEven
        typeof(GetSystemInfo),
        typeof(GetVncInfo),
        typeof(GetRdpInfo),
        typeof(DoVisitWebsite
        typeof(DoShowMessageBox),
        typeof(DoClientUpdate),
        typeof(GetMonitors),
        typeof(GetWebcams),
        typeof(GetWebcam),
        typeof(DoShellExecute),
```

Added Commands

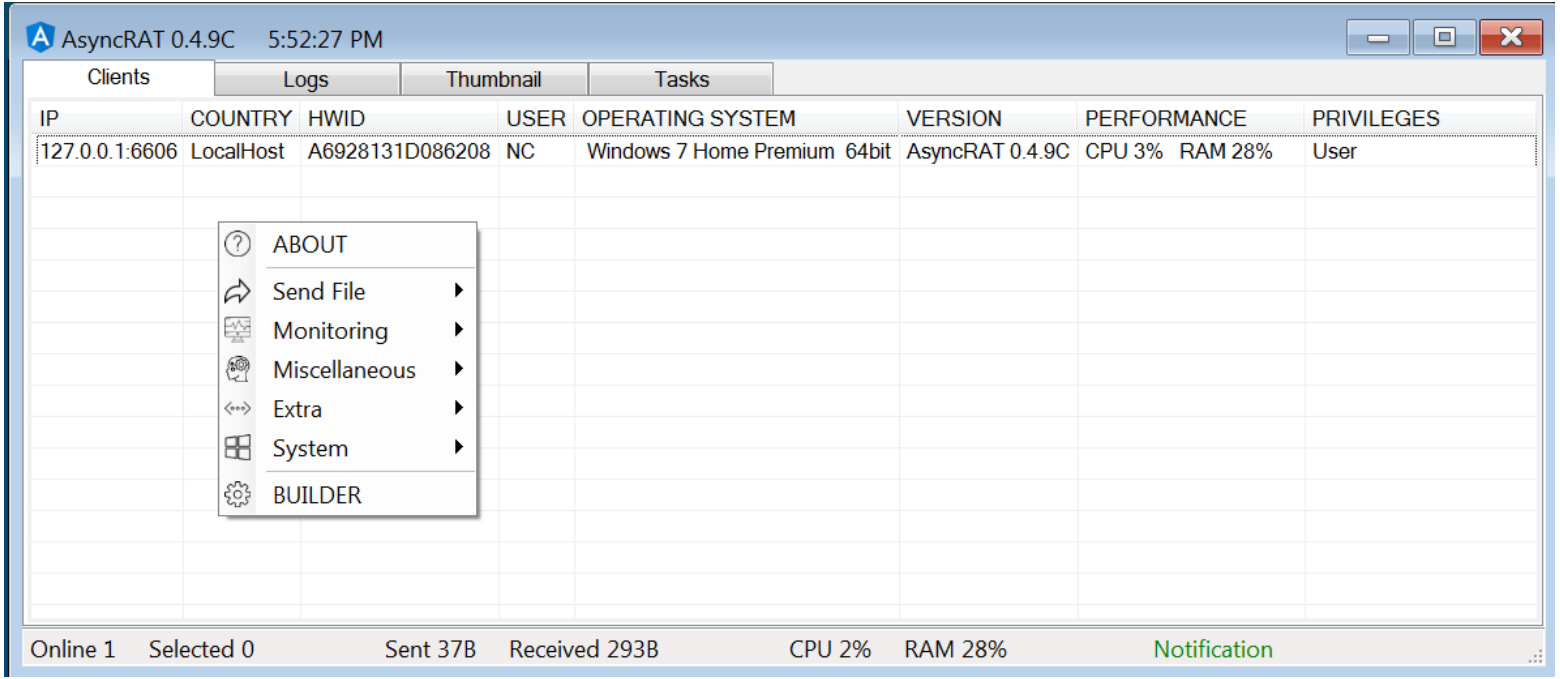
```
typeof(DoRemoveRdp),
typeof(DoRemoveVnc),
typeof(DoInstallVNC),
typeof(DoInstallRDP),
typeof(DoStealer),
typeof(GetVncInfo),
typeof(GetRdpInfo),
typeof(GetAllPasswords),
```

QuasarRAT Commands

```
namespace xClient.Core.Packets
{
    public class PacketRegistery
    {
        public static Type[] GetPacketTypes()
        {
            return new Type[]
            {
                typeof (Packets.ServerPackets.GetAuthentication),
                typeof (Packets.ServerPackets.DoClientDisconnect),
                                                  ientReconnect),
                                                  ientUninstall),
                                                  bcamStop),
                                                  kElevate),
                                                  wnloadAndExecute),
                                                  loadAndExecute),
                                                  esktop),
                                                  rocesses),
                                                  ocessKill),
                                                  ocessStart),
                                                  rives),
                                                  irectory),
                                                  wnloadFile),
                                                  useEvent),
                                                  yboardEvent),
                                                  ystemInfo),
                                                  sitWebsite),
                                                  owMessageBox),
                                                  ientUpdate),
                                                  onitors),
                typeof (Packets.ServerPackets.GetWebcams),
                typeof (Packets.ServerPackets.GetWebcam),
                typeof (Packets.ServerPackets.DoShellExecute),
                typeof (Packets.ServerPackets.DoPathRename),
                typeof (Packets.ServerPackets.DoPathDelete),
                typeof (Packets.ServerPackets.DoShutdownAction),
```

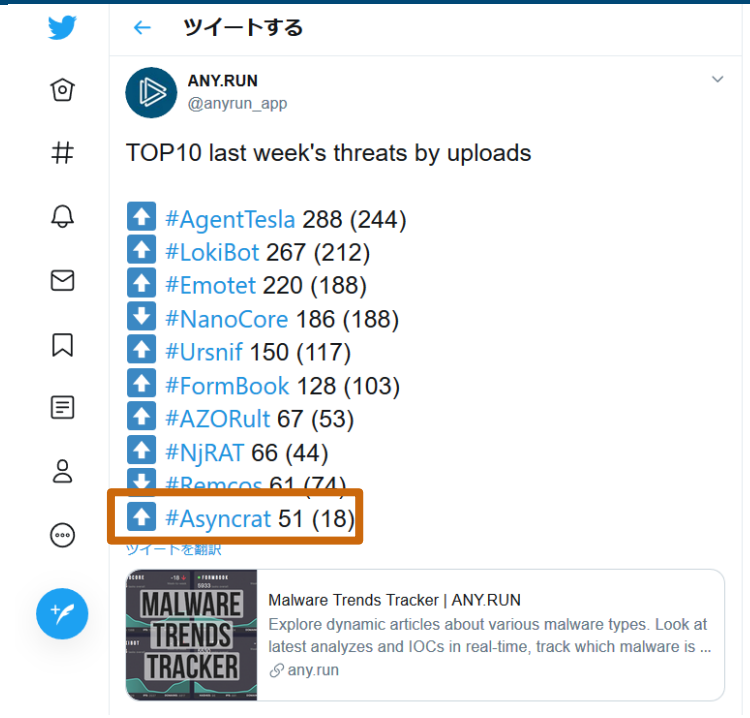
AsyncRAT

AsyncRAT is a plugin-based RAT.



AsyncRAT

AsyncRAT is a plugin-based RAT.



AsyncRAT

AsyncRAT uses the same communication method as QuasarRAT.

AsyncRAT

```
namespace Client.Algorithm
{
        public class Aes256
        {
            private const int KeyLength = 32;
            private const int AuthKeyLength = 64;
            private const int IvLength = 16;
            private const int HmacSha256Length = 32;
            private readonly byte[] _key;
            private readonly byte[] _authKey;

            private static readonly byte[] Salt =
            {
            0xBF, 0xEB, 0x1E, 0x56, 0xFB, 0xCD, 0x97, 0x3B, 0xB2, 0x19, 0x2, 0x24, 0x30, 0xA5, 0x78, 0x43, 0x0, 0x3D, 0x56,
            0x44, 0xD2, 0x1E, 0x62, 0xB9, 0xD4, 0xF1, 0x80, 0xE7, 0xE6, 0xC3, 0x39, 0x41
        };
```

QuasarRAT

```
namespace xClient.Core.Cryptography
{
    public static class AES
    {
        private const int IvLength = 16;
        private const int HmacSha256Length = 32;
        private static byte[] _defaultKey;
        private static byte[] _defaultAuthKey;

        public static readonly byte[] Salt =
        {
            0xBF, 0xEB, 0x1E, 0x56, 0xFB, 0xCD, 0x97, 0x3B, 0xB2, 0x19, 0x2, 0x24, 0x30, 0xA5, 0x78, 0x43, 0x0, 0x3D, 0x56,
            0x44, 0xD2, 0x1E, 0x62, 0xB9, 0xD4, 0xF1, 0x80, 0xE7, 0xE6, 0xC3, 0x39, 0x41
        };
```

Same AES256 salt value and communication format.

Void-RAT

Void-RAT is a simple RAT that uses the same communication method as Quasar RAT.

Void-RAT

Void-RAT

Void-RAT is based on the **decompiled source code** of QuasarRAT.

Void-RAT

```
int HmacSha256Length = 32;

byte[] defaultKey;

private static byte[] defaultAuthKey;

public static readonly byte[] Salt = new byte[]
{
    191,
    235,
    30,
    86,
    251,
    205,
    151,
    59,
    178,
    25,
    2,
    36,
    48,
    165,
    120,
    67,
```

QuasarRAT decompile

```
// Token: 0x040001BB RID: 443
private static byte[] byte_0;

// Token: 0x040001BC RID: 444
private static byte[] byte_1;

// Token: 0x040001BD RID: 445
public static readonly byte[] byte_2 = new byte[]
{
    191,
    235,
    30,
    86,
    251,
    205,
    151,
    59,
    178,
    25,
    2,
    36,
    48,
    165,
    120,
    67,
```

Same AES256 salt value.

1 QuasarRAT Internals

2 Quasar Family

3 Campaigns using QuasarRAT

4 Hunting Quasar Family C2

Campaigns using QuasarRAT

Campaign Name	QuasarRAT Version	Custom	Obfuscation
APT33	1.3.0.0	No	ConfuserEx v1.0.0
Gorgon Group		No	
APT-C-09	2.0.0.0 RELEASE3	No	
DustySky	1.1.0.0	No	
APT10	2.0.0.0 **(Custom Version)**	Yes	ConfuserEx v1.0.0
Spear Phishing Campaign Targets Ukraine *	1.1.0.0 2.0.0.0 RELEASE3	No	

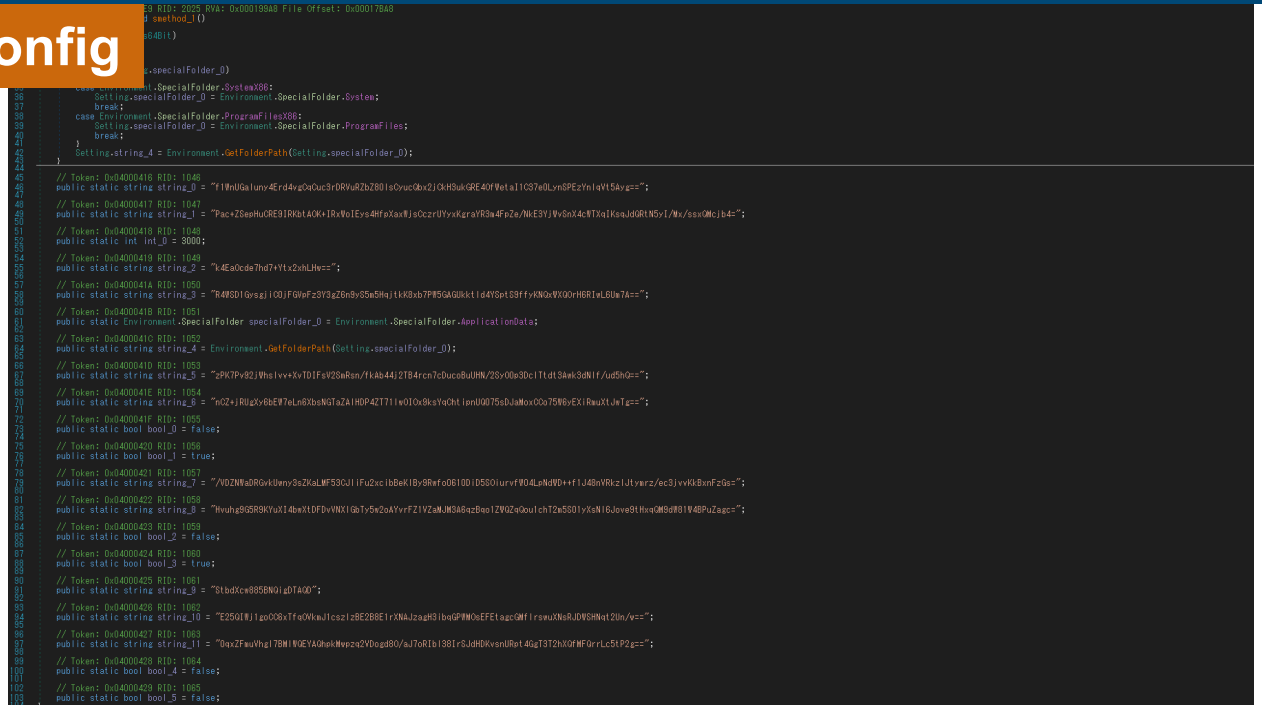
Most campaigns use the original RAT for attacks.

* https://www.fireeye.com/blog/threat-research/2019/04/spear-phishing-campaign-targets-ukraine-government.html
https://www.welivesecurity.com/wp-content/uploads/2018/07/ESET_Quasar_Sobaken_Vermin.pdf

Case of APT33

APT33 uses the original RAT (1.3.0.0) for attacks.

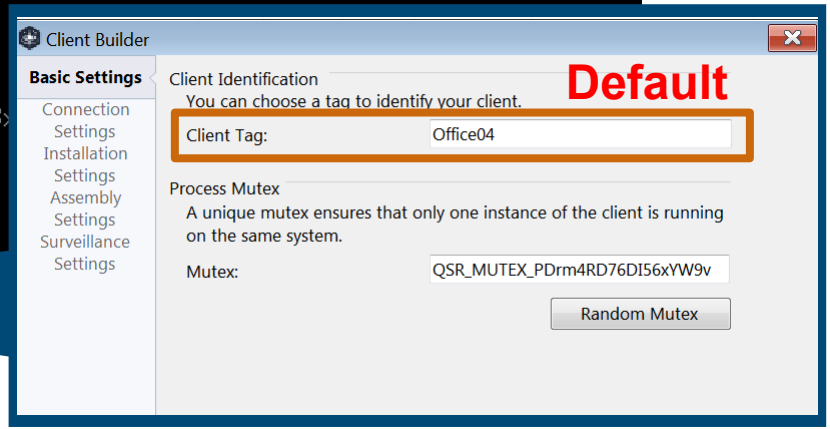
APT33 config



```
9 RID: 2025 RVA: 0x000199A8 File Offset: 0x00017BA8
f smethod_1()
64Bit)
.specialFolder_0)
.SpecialFolder.SystemX86:
    Setting.specialFolder_0 = Environment.SpecialFolder.System;
    break;
case Environment.SpecialFolder.ProgramFilesX86:
    Setting.specialFolder_0 = Environment.SpecialFolder.ProgramFiles;
    break;
}
Setting.string_4 = Environment.GetFolderPath(Setting.specialFolder_0);
}

// Token: 0x04000416 RID: 1046
public static string string_0 = "f1VnUGaluny4Erd4vgOxCuc3rDRVuRZbZ80lsCvuc0bx2jCkH3ukGRE4OfVetaI1CS7e0LvnSPEzYnlaVt5Ayg==";

// Token: 0x04000417 RID: 1047
public static string string_1 = "Pac+ZSeeHuCRE9IRKbtAOK+IRxVoIEvs4HfpXaxWjsCczrUYyxKgraYR3w4FnZe/NkE9YjVv5nX4cWTXqIKsqJdGRtN5yI/Wx/ssxOMcjb4=";

// Token: 0x04000418 RID: 1048
public static int int_0 = 3000;

// Token: 0x04000419 RID: 1049
public static string string_2 = "k4Ea0cde7hd7+Ytx2xhLHw==";

// Token: 0x0400041A RID: 1050
public static string string_3 = "R4WSDIGvsgjiOJFGVeFz3Y3gZGn8yS5m5HqjtkK8xb7PM5GAGUkktId4YSptS9ffyKNQxVXQOrH6RImLGUm7A==";

// Token: 0x0400041B RID: 1051
public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;

// Token: 0x0400041C RID: 1052
public static string string_4 = Environment.GetFolderPath(Setting.specialFolder_0);

// Token: 0x0400041D RID: 1053
public static string string_5 = "zPK7Pv92jVhsIvv+XvTOIFsV2SmRsn/fkAb44j2T84rcn7cDuceBuUHN/2Sy0Qe3DcITdt3Awk3dNIf/ud5hQ==";

// Token: 0x0400041E RID: 1054
public static string string_6 = "nCZ+jRUgXy6bEW7eLn6XbsNGTaZAlHDP4ZT7IIwOIOx8ksYqChtipnUQO75aDJaWoxCCo75N6yEXiRwuXtJwTg==";

// Token: 0x0400041F RID: 1055
public static bool bool_0 = false;

// Token: 0x04000420 RID: 1056
public static bool bool_1 = true;

// Token: 0x04000421 RID: 1057
public static string string_7 = "/VDZNMaDRGvkUwny3sZKaLMF53CJiFu2xcibBeKlBy9RefoO61ODiD5SOiurvf9O4LeNdVD+fiJ4BnVRkzlJtymrz/ec3jvvkkBxnFzGs=";

// Token: 0x04000422 RID: 1058
public static string string_8 = "Hvuhg8G5R9KYuXI4bmXtDFDvVNXIGbTy5w2oAYvrFZ1VZaWJM3A8qzBaoIZVQZeQouIchTZm5SOIyXsNI6Jove9tHxaQM9df01V4BPuZsgc=";

// Token: 0x04000423 RID: 1059
public static bool bool_2 = false;

// Token: 0x04000424 RID: 1060
public static bool bool_3 = true;

// Token: 0x04000425 RID: 1061
public static string string_9 = "StbdXcm885BNQiqDTAQO";

// Token: 0x04000426 RID: 1062
public static string string_10 = "E25GIMj1goOC0xTfaOVkwJlcszlzBE2BBE1rXNAJzagH3ibqGPMMOsEFEtagcGMflrswuXNaRJDWSHNat2Un/w==";

// Token: 0x04000427 RID: 1063
public static string string_11 = "QxxZFwuVhgl7BMlNQEYAQhekNvpzq2VDogd8O/aJToRIbl38IrSJdHDKvsnURpt4GgT3T2hXOfMFQrrLc5tP2g==";

// Token: 0x04000428 RID: 1064
public static bool bool_4 = false;

// Token: 0x04000429 RID: 1065
public static bool bool_5 = false;
```

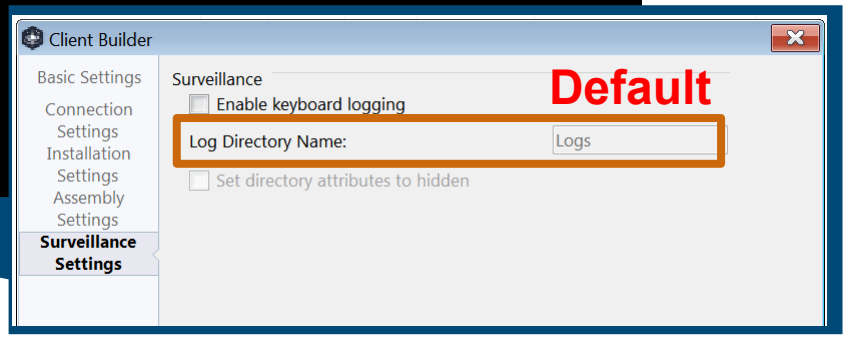
APT33 QuasarRAT Configuration

```
[+] Detect malware by Yara rules.
[+]   Process Name      : a23c18.exe
[+]   Process ID        : 6660
[+]   Malware name      : Quasar
[+]   Base Address(VAD) : 0x1F80000
[+]   Size              : 0x57000
----------------------------------------------------------------------
Process: a23c18.exe (6660)

[Config Info]
VERSION             : 1.3.0.0
HOSTS               : mywinnetwork.ddns.net:80;
KEY (Base64)        : k4EaOcde7hd7+Ytx2xhLHw==
AUTHKEY (Base64)    : R4WSD1GysgjiCOjFGVpFz3Y3gZ6n9yS5m5HqjtkK8
SUBDIRECTORY        : SubDir
INSTALLNAME         : Client.exe
MUTEX               : QSR_MUTEX_RjrwbI1Icgf7POFqQF
STARTUPKEY          : Windows Session Manager
ENCRYPTIONKEY       : SthdYcw88FBNQigDTAQD
TAG                 : Office04
LOGDIRECTORYNAME    : Logs
```

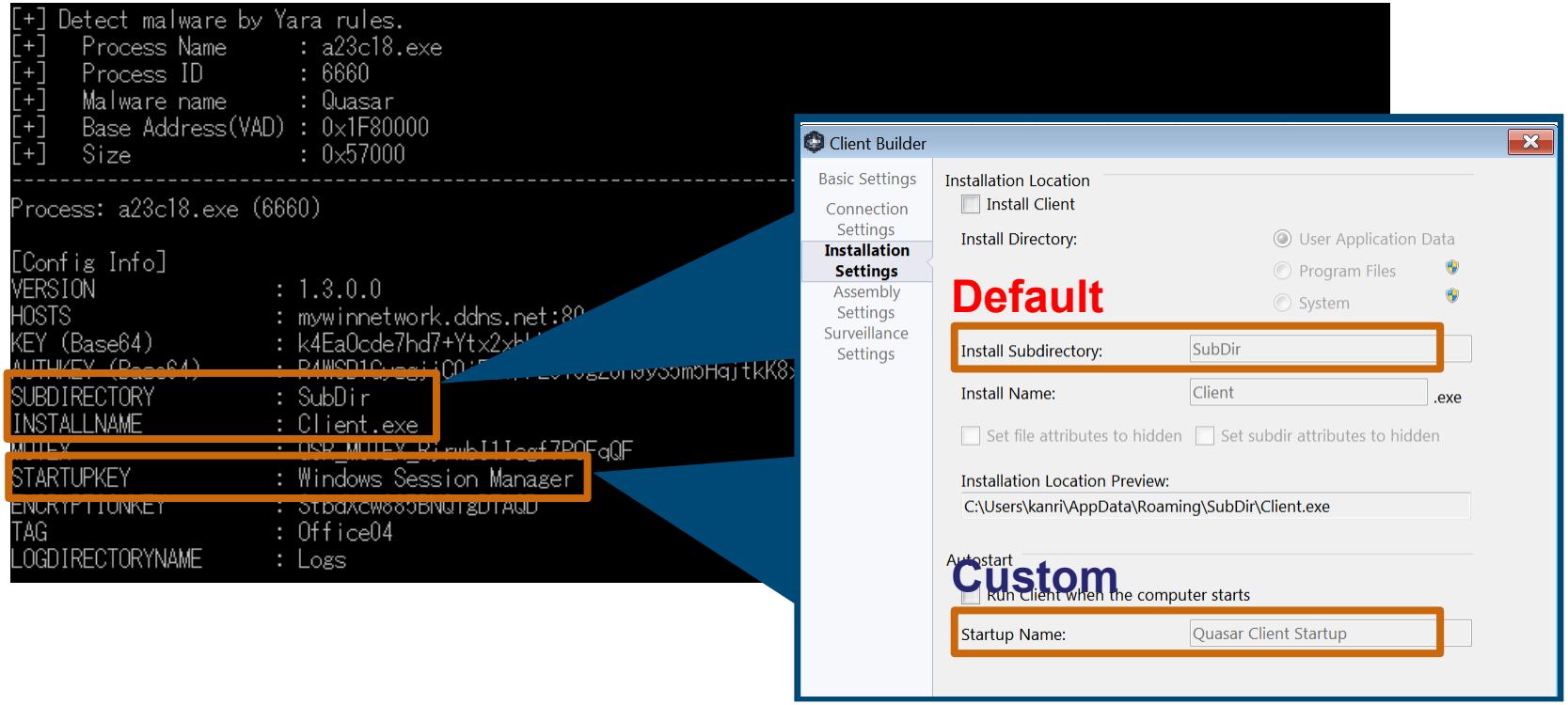


APT33 QuasarRAT Configuration

```
[+] Detect malware by Yara rules.
[+]   Process Name      : a23c18.exe
[+]   Process ID        : 6660
[+]   Malware name      : Quasar
[+]   Base Address(VAD) : 0x1F80000
[+]   Size              : 0x57000
----------------------------------------------------------------------
Process: a23c18.exe (6660)

[Config Info]
VERSION              : 1.3.0.0
HOSTS                : mywinnetwork.ddns.net:80;
KEY (Base64)         : k4Ea0cde7hd7+Ytx2xhLHw==
AUTHKEY (Base64)     : R4WSD1GysgjiC0jFGVpFz3Y3gZ6n9yS5m5HqjtkK8xb7PW5GAGUkktld4YSptS9ffyKNQxWXQOrH6RIwL6Um7A==
SUBDIRECTORY         : SubDir
INSTALLNAME          : Client.exe
MUTEX                : QSR_MUTEX_RjrwbI1Icgf7POFqQF
STARTUPKEY           : Windows Session Manager
ENCRYPTIONKEY        : StbdXcw885BNQigDTAQD
TAG                  : Office04
LOGDIRECTORYNAME     : Logs
```



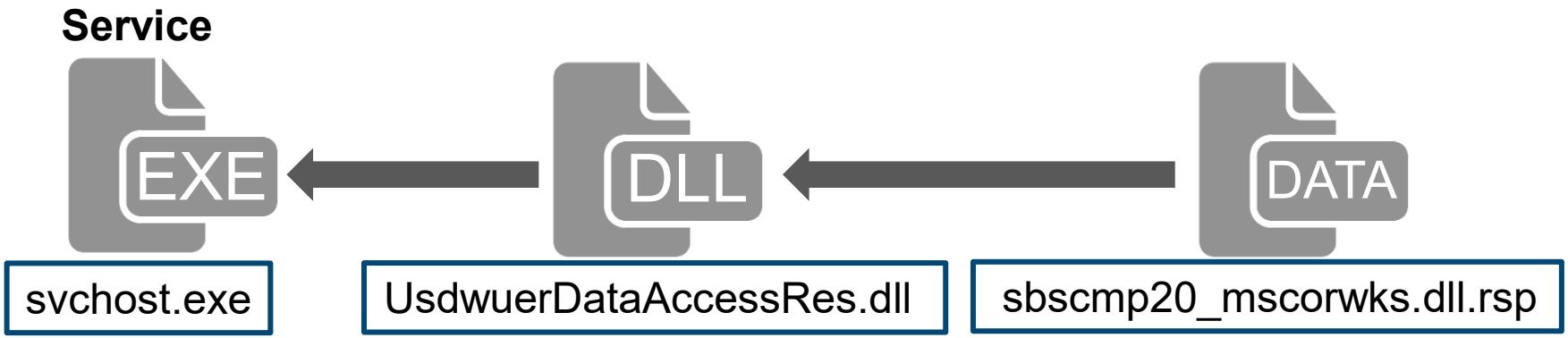
APT33 QuasarRAT Configuration



QuasarRAT Configuration Pattern

QuasarRAT used in APT case has default configuration,
so it is difficult to determine attribution using **"configuration"** only.

"Custom" Case of APT10



Service	Value
EntryName	PimIndexMaintenanceSvc_hhdx9
DisplayName	Contact Data_hhdx9
ImagePath	C:¥Windows¥system32¥svchost.exe -k codazisvc
ServiceDLL	C:¥Windows¥System32¥UsdwuerDataAccessRes.dll

UsdwuerDataAccessRes.dll

QuasarRAT Loader

Export Function

- FuckYouAnti

Loading File

- **Folder**: C:¥Windows¥Microsoft.NET¥
- **Size**: 100KB < size < 500KB

Decrypt

- **Mode**: AES-CBC
- **Pass**: tVihjTlrfrFMmZkRALGM
- **Salt**: 3,1,4,1,5,9,2,6,5,3,5,8,9,7,2,3,8,4,6,2,6,4,3,3

PDB

- C:¥Users¥user¥Desktop¥template¥libogg¥x64¥Release¥libogg.pdb

```
; Exported entry   1. FuckYouAnti

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

; Attributes: library function static bp-based fram

                public FuckYouAnti
FuckYouAnti     proc near               ; DATA XREF
                                        ; .rdata:of

var_90          = qword ptr -90h
Block           = qword ptr -88h
var_78          = qword ptr -78h
var_70          = qword ptr -70h
var_60          = qword ptr -60h
var_50          = qword ptr -50h
var_48          = qword ptr -48h
var_38          = qword ptr -38h
var_20          = qword ptr -20h
var_10          = qword ptr -10h
var_s0          = byte ptr  0

; FUNCTION CHUNK AT .text:000000018000FFA0 SIZE 000
; FUNCTION CHUNK AT .text:000000018000FFB0 SIZE 000
; FUNCTION CHUNK AT .text:000000018000FFC0 SIZE 000

; __unwind { // __GSHandlerCheck_EH
                mov     rax, rsp
                push    rbp
                lea     rbp, [rax-5Fh]
                sub     rsp, 0B0h
```

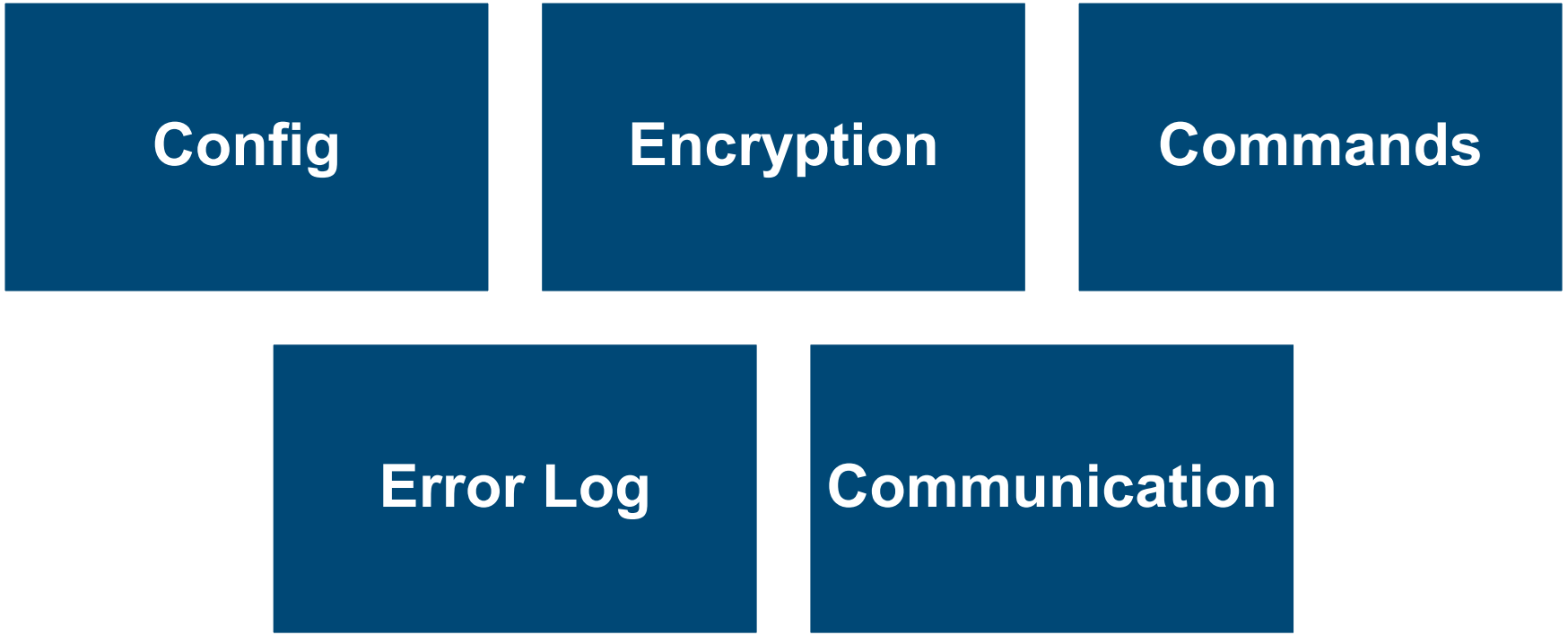
sbscmp20_mscorwks.dll.rsp

QuasarRAT

File Path

- **C:¥Windows¥Microsoft.NET¥**Framework64¥sbscmp20_mscorwks.dll.rsp

Customized APT10 QuasarRAT



APT10 QuasarRAT Configuration

APT10 QuasarRAT

```
    // Token: 0x04000056 RID: 86
    public static string VERSION = "q06kN0+IHvjn9lm0vEc40eyrUSORpHIQj3PD/rxSECWjyInSrDnBIdGc9iVSz7JgM7kXxulcfGLxQ6YTp/Zfsg==";

    // Token: 0x04000057 RID: 87
    public static string HOSTS = "DmIP21/zkSCJvK3qijV/9JB7OiFjbt5ZgY091Devksz3bnrJX3XnNTOuVsejIYJBFgXTOIh3OTImcEnvQIk0iNwNJRExqXXE4k1

    // Token: 0x04000058 RID: 88
    public static int RECONNECTDELAY = 34109;

    // Token: 0x04000059 RID: 89
    public static string KEY = "+PrSiHnNK8Mpq4nK+spXzw==";

    // Token: 0x0400005A RID: 90
    public static string AUTHKEY = "Gr7icnpxp4j1oWFhmLnblfe2fnmzIMEiuuCNQ20SsJ24cE3b0dT/zVObPVIgENbELTQQM3hm50qk0CckZ2cn0g==";

    // Token: 0x0400005B RID: 91
    public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;

    // Token: 0x0400005C RID: 92
    public static string DIRECTORY = Environment.GetFolderPath(init_config.specialFolder_0);

    // Token: 0x0400005D RID: 93
    public static string SUBDIRECTORY = "U0+XcT7VIxB01c0jBQ+75wnOGkqwINckNwkRLAfiLg1i0RyKpXCUMSi1ArCMwZmVMvJ5pUdpCJaec0gDCnlpmw==";

    // Token: 0x0400005E RID: 94
    public static string INSTALLNAME = "tpW0ttiTfTALm/y1nqePR0SZ3Lcv3hrOVrgxjYctKgmqTdHJINTbSmwwiyKL9cH5943cEz2Auzk4rBLBP7mAhw==";

    // Token: 0x0400005F RID: 95
    public static bool INSTALL = false;

    // Token: 0x04000060 RID: 96
    public static bool STARTUP = false;

    // Token: 0x04000061 RID: 97
    public static string MUTEX = "yaXi6Hf4cuVmHwZCOM1s5A2J9JJpTKdao4ufUOISQ17swcOVyyiQJ4maUCLttsJlk3BJXX1HpK8IxJ9sWZFPQX0LmpfoORBz0FZ

    // Token: 0x04000062 RID: 98
    public static string STARTUPKEY = "17CimMdhwiGVfzkT5DYzI3dMdhjNS0x+534Etf37R96xwZOOKaHvCMpb3Yh5cOz0rte2eBu4OUJ3OR3IWCJRrEgZIMJF8r

    // Token: 0x04000063 RID: 99
    public static bool HIDEFILE = false;

    // Token: 0x04000064 RID: 100
    public static bool ENABLELOGGER = false;

    // Token: 0x04000065 RID: 101
    public static string ENCRYPTIONKEY = "mAAxGFFqcQAMw8HMU3Iq";

    // Token: 0x04000066 RID: 102
    public static string TAG = "MOP87R4Ni4bnTawBeUXavHm2arf9I5mdTyIQzcv4GOHfc9L4rXxYducEhmEc32ORgjTCc/SNp6nri2MSDh8SoA==";

    // Token: 0x04000067 RID: 103
    public static string LOGDIRECTORYNAME = "EqqsurqmvQbtYMWWN8cPHIpJ/s67IB7GeZ5xo28J+ZOCbxbZKaBB98aGGBSYINraFuvIcspu9toOwwfY5Z8uyg==

    // Token: 0x04000068 RID: 104
    public static bool HIDELOGDIRECTORY = false;

    // Token: 0x04000069 RID: 105
    public static bool HIDEINSTALLSUBDIRECTORY = false;

    // Token: 0x0400006A RID: 106
    public static string Download_url = "Q9Df9v8dzGX4E/07n/DYUxSkKzxrGJ2hbs8GUjpG8og07ZUNmNIFyPe7u2XbdrecaR0wg+jnwAZs/Imv60zuMg==";

    // Token: 0x0400006B RID: 107
    public static string Proxy = "tINxfsfXhIgRcBKu299BkHh8ftOXrC3SBiLAYaPEyjCPr/HjNA2eKg0/a2gZCC0JnfooYqwEu7UksRe02L7KvA==";
}
```

QuasarRAT

```
        // Token: 0x04000008 RID: 8
        public static string VERSION = "UvADP/X6YS4Pi/TtR6RC049f9HntDsrnWRpHNg4Sc7M9caQvJryNboozrQAdD2e39pa5nVYAe1k9B5+dvQGMEw==";

        // Token: 0x04000009 RID: 9
        public static string HOSTS = "Tqu77dMuK/ZIsz66TL5IQUxE8/LQhIir5uxAqtiTHCs4hvRJd5xp4c035SnBtDLtmdK+sDe2ewmNBkSJWjug8HEoms8v9d8aFtI28DIw4QgEkK5XI

        // Token: 0x0400000A RID: 10
        public static int RECONNECTDELAY = 3000;

        // Token: 0x0400000B RID: 11
        public static string KEY = "ebdOoS8G89sUEWPiItdABQ==";

        // Token: 0x0400000C RID: 12
        public static string AUTHKEY = "SOc2k9nqnECTE5yMFKY1PRF90HSlbZLcqE0ccuiIHMgbOR8OXD7Fv9XDKcDJAU/mouBB4/KAEpaWUM9KJEJZBQ==";

        // Token: 0x0400000D RID: 13
        public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;

        // Token: 0x0400000E RID: 14
        public static string DIRECTORY = Environment.GetFolderPath(Init_config.specialFolder_0);

        // Token: 0x0400000F RID: 15
        public static string SUBDIRECTORY = "di7ALkQAISIPt5o4bbnsEuESGLSgwUEYBMEMHKzTpR9SUruwW6dmfsS9Z1XIyrt3I8xpkwaScvpuQA9aFCGCVA==";

        // Token: 0x04000010 RID: 16
        public static string INSTALLNAME = "/np6aAFTWsNwJIk0C428HV7MCckYc+gwJURNOYibATKpCO0pqi9qblwFMLXptV/cDEbDjaYbN/Zfs2B8/JSBJQ==";

        // Token: 0x04000011 RID: 17
        public static bool INSTALL = false;

        // Token: 0x04000012 RID: 18
        public static bool STARTUP = false;

        // Token: 0x04000013 RID: 19
        public static string MUTEX = "X1T4grV+qtsriFJxWjPtOMeVzcIj57988AaOT08glRtO0gFDWr1yQ95F1237JEEUbJuNTBICXtkMavYPEBq9LQMVUTWE/fa5pptXFSQRxwA=";

        // Token: 0x04000014 RID: 20
        public static string STARTUPKEY = "Ip3JtNTns9moIMzfUvUddz0VOMkvQizpAV8RFTtK9ARtWdNPATjCQHqKCnJx+OTBrTQyEjHNPuSK3zhV0V/ZkNa0qLL7er9QH+HEHtOYbuO

        // Token: 0x04000015 RID: 21
        public static bool HIDEFILE = false;

        // Token: 0x04000016 RID: 22
        public static bool ENABLELOGGER = true;

        // Token: 0x04000017 RID: 23
        public static string ENCRYPTIONKEY = "2XmmEP5ycdaJwbloz781";

        // Token: 0x04000018 RID: 24
        public static string TAG = "KtAqIrxyxoIoMbHNYnJjH+p+y2N4dMmtIiM1IICP8QfBFBiPSMOqzI8ysA16VC88eicwzIZitM+nDIN4A4ZGug==";

        // Token: 0x04000019 RID: 25
        public static string LOGDIRECTORYNAME = "aX4+OTWn0952T790CiTfmXX/vuFGG5SSu4YjWRzZf6QJdte3LJ2US0X6J3pyqfsq9EeN8Ynhhihoy84DKNFRgw==";

        // Token: 0x0400001A RID: 26
        public static bool HIDELOGDIRECTORY = true;

        // Token: 0x0400001B RID: 27
        public static bool HIDEINSTALLSUBDIRECTORY = false;
    }
}
```

APT10 QuasarRAT Configuration

APT10 QuasarRAT

Config Value	
VERSION	STARTUPKEY
HOSTS	HIDEFILE
RECONNECTDELY	ENABLELOGGER
KEY	ENCRYPTIONKEY
AUTHKEY	TAG
DIRECTORY	LOGDIRECTORY
SUBDIRECTORY	HIDELOGDIRECTORY
INSTALLNAME	HIDELOGSUBDIRECTORY
INSTALL	DOWNLOAD_URL NEW
SETUP	PROXY NEW
MUTEX	

APT10 QuasarRAT Configuration

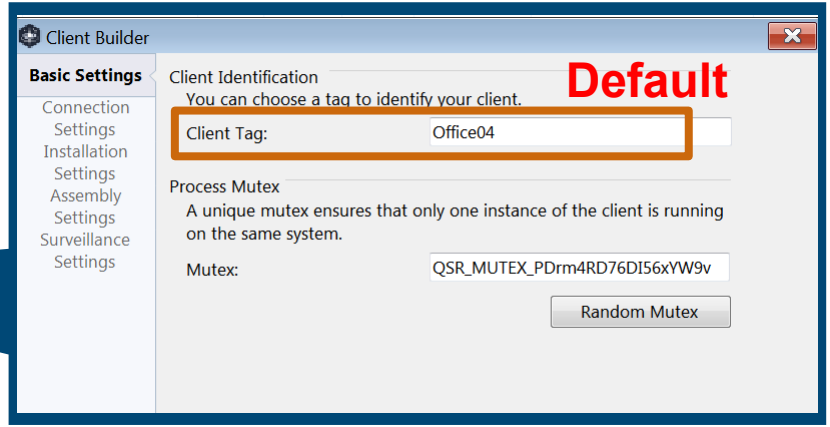
Config

```
VERSION                 : 2.0.0.0
HOSTS                   : 178.32.24.180:443;
RECONNECTDELAY          : 34109
KEY (Base64)            : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64)        : Gr7icnpxp4j1oWFhmLnbIfe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY            : SubDir
INSTALLNAME             : Client.exe
INSTALL                 : FALSE
STARTUP                 : FALSE
MUTEX                   : cOKPyGqEqejOVh0SaOwHlXa5
STARTUPKEY              : Quasar Client Startup
HIDEFILE                : FALSE
ENABLELOGGER            : FALSE
ENCRYPTIONKEY           : mAAxGFFqcOAMw8HMJ3Iq
TAG                     : Office04
LOGDIRECTORYNAME        : Logs
HIDEDIRECTORY           : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url            : none
Proxy                   :
```

APT10 QuasarRAT Configuration

Config

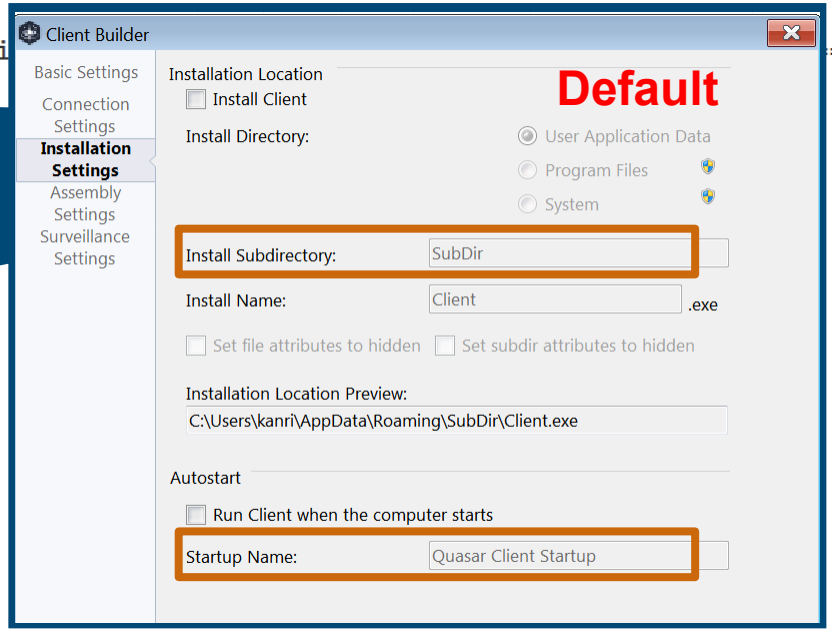
```
VERSION                 : 2.0.0.0
HOSTS                   : 178.32.24.180:443;
RECONNECTDELAY          : 34109
KEY (Base64)            : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64)        : Gr7icnpxp4j1oWFhmLnbIfe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY            : SubDir
INSTALLNAME             : Client.exe
INSTALL                 : FALSE
STARTUP                 : FALSE
MUTEX                   : cOKPyGqEqejOVh0SaOwHlXa5
STARTUPKEY              : Quasar Client Startup
HIDEFILE                : FALSE
ENABLELOGGER            : FALSE
ENCRYPTIONKEY           : mAAxGFFqcOAMw8HMJ3Iq
TAG                     : Office04
LOGDIRECTORYNAME        : Logs
HIDEDIRECTORY           : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url            : none
Proxy                   :
```



APT10 QuasarRAT Configuration

Config

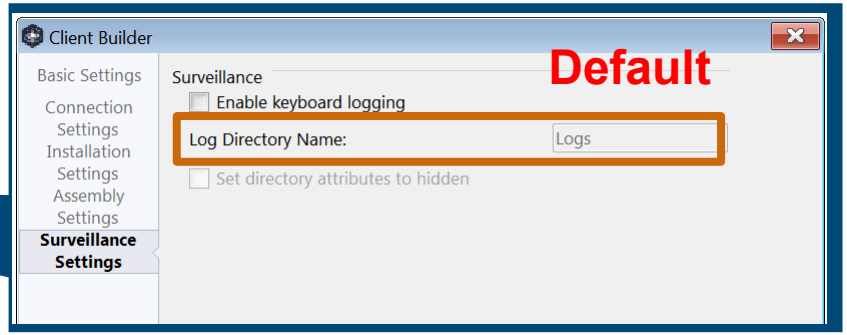
```
VERSION                 : 2.0.0.0
HOSTS                   : 178.32.24.180:443;
RECONNECTDELAY          : 34109
KEY (Base64)            : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64)        : Gr7icnpxp4j1oWFhmLnbIfe2fnmzIMEi
SUBDIRECTORY            : SubDir
INSTALLNAME             : Client.exe
INSTALL                 : FALSE
STARTUP                 : FALSE
MUTEX                   : cOKPyGqEqejOVh0SaOwHlXa5
STARTUPKEY              : Quasar Client Startup
HIDEFILE                : FALSE
ENABLELOGGER            : FALSE
ENCRYPTIONKEY           : mAAxGFFqcOAMw8HMJ3Iq
TAG                     : Office04
LOGDIRECTORYNAME        : Logs
HIDEDIRECTORY           : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url            : none
Proxy                   :
```



APT10 QuasarRAT Configuration

Config

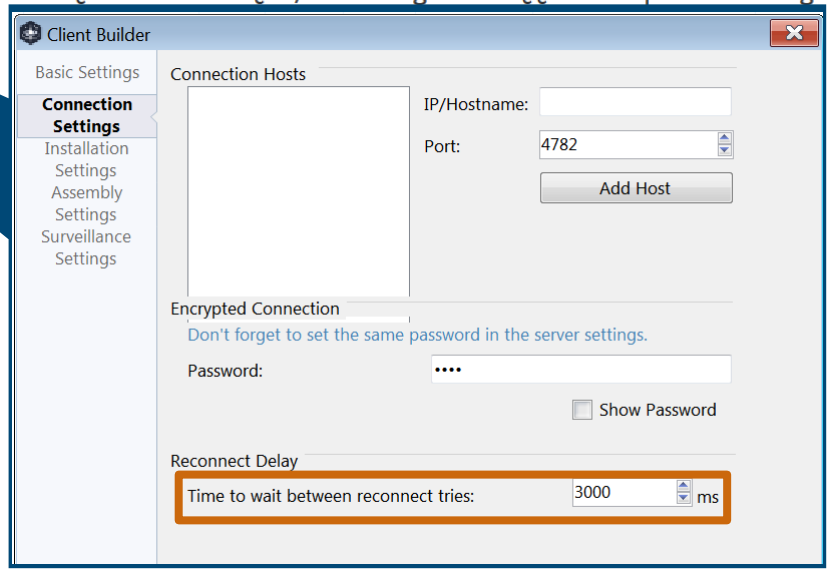
```
VERSION                 : 2.0.0.0
HOSTS                   : 178.32.24.180:443;
RECONNECTDELAY          : 34109
KEY (Base64)            : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64)        : Gr7icnpxp4j1oWFhmLnbIfe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY            : SubDir
INSTALLNAME             : Client.exe
INSTALL                 : FALSE
STARTUP                 : FALSE
MUTEX                   : cOKPyGqEqejOVh0SaOwHlXa5
STARTUPKEY              : Quasar Client Startup
HIDEFILE                : FALSE
ENABLELOGGER            : FALSE
ENCRYPTIONKEY           : mAAxGFFqcOAMw8HMJ3Iq
TAG                     : Office04
LOGDIRECTORYNAME        : Logs
HIDEDIRECTORY           : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url            : none
Proxy                   :
```



APT10 QuasarRAT Configuration

Config

```
VERSION                 : 2.0.0.0
HOSTS                   : 178.32.24.180:443;
RECONNECTDELAY          : 34109
KEY (Base64)            : +PrSihNNM[illegible]Mpq4nK+spXzw==
AUTHKEY (Base64)        : Gr7icnpxp4j[illegible]hmLnbIfe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY            : SubDir
INSTALLNAME             : Client.exe
INSTALL                 : FALSE
STARTUP                 : FALSE
MUTEX                   : cOKPyGqEqejOVh0SaOwHlXa5
STARTUPKEY              : Quasar Client Startup
HIDEFILE                : FALSE
ENABLELOGGER            : FALSE
ENCRYPTIONKEY           : mAAxGFFqcOAMw8HMJ3Iq
TAG                     : Office04
LOGDIRECTORYNAME        : Logs
HIDEDIRECTORY           : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url            : none
Proxy                   :
```



Configuration Encryption

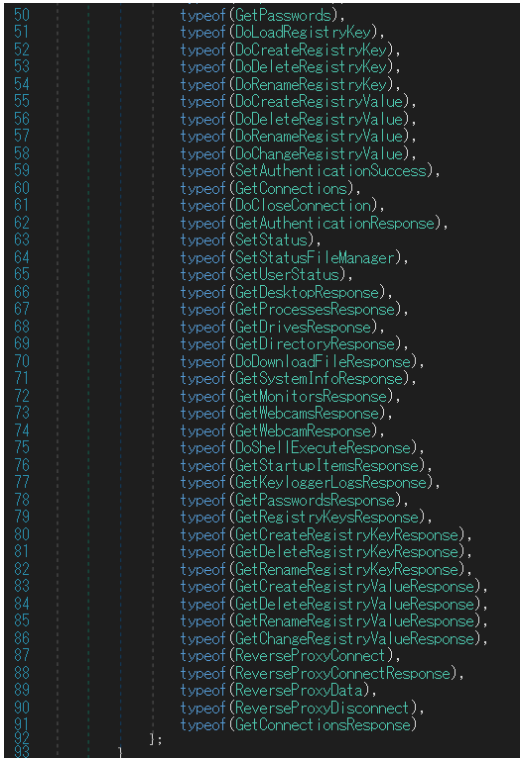
APT10 QuasarRAT

QuasarRAT

Commands

APT10 QuasarRAT

QuasarRAT



Commands

APT10 QuasarRAT

Added Functions

- **DoPlugin**
  - Add plugin module function
- **DoPluginResponse**
  - Delete plugin module function

Error Log File

APT10 QuasarRAT

```
62      // Token: 0x060000DA RID: 218
63      private static void create_error_log(Exception e)
64      {
65          if (e != null)
66          {
67              using (StreamWriter streamWriter = new StreamWriter(new FileStream("c:¥¥temp¥¥CEgibifi.tmp", FileMode.Append)))
68              {
69                  streamWriter.AutoFlush = true;
70                  streamWriter.WriteLine(string.Format("==================[0]==================", DateTime.Now.ToString()));
71                  streamWriter.WriteLine(e.Message);
72                  streamWriter.WriteLine(e.StackTrace);
73                  if (e.InnerException != null)
74                  {
75                      streamWriter.WriteLine(e.InnerException.Message);
76                      streamWriter.WriteLine(e.InnerException.StackTrace);
77                  }
78              }
79          }
80      }
81
```

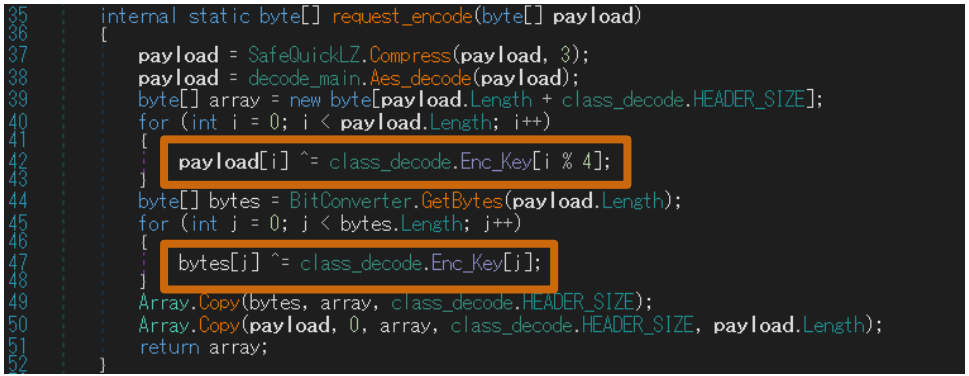
File Name

- C:¥temp¥CEgibifi.tmp

Communication

APT10 QuasarRAT **Added XOR using encryption key.**

Size(4byte) + XOR	HMAC(32byte) + XOR	QuickLZ + AES(mode CFB) + XOR



```
internal static byte[] request_encode(byte[] payload)
{
    payload = SafeQuickLZ.Compress(payload, 3);
    payload = decode_main.Aes_decode(payload);
    byte[] array = new byte[payload.Length + class_decode.HEADER_SIZE];
    for (int i = 0; i < payload.Length; i++)
    {
        payload[i] ^= class_decode.Enc_Key[i % 4];
    }
    byte[] bytes = BitConverter.GetBytes(payload.Length);
    for (int j = 0; j < bytes.Length; j++)
    {
        bytes[j] ^= class_decode.Enc_Key[j];
    }
    Array.Copy(bytes, array, class_decode.HEADER_SIZE);
    Array.Copy(payload, 0, array, class_decode.HEADER_SIZE, payload.Length);
    return array;
}
```

Command(1byte)	argv

* QuickLZ compress level3

1 QuasarRAT Internals

2 Quasar Family

3 Campaigns using QuasarRAT

4 Hunting Quasar Family C2

Hunting Quasar Family C2

VirusTotal

- Hunting Quasar family malware in VT.

IoT Search Engine

- Using Shodan, Censys, FOFA
- Search for features of Quasar family C2 from scan data.

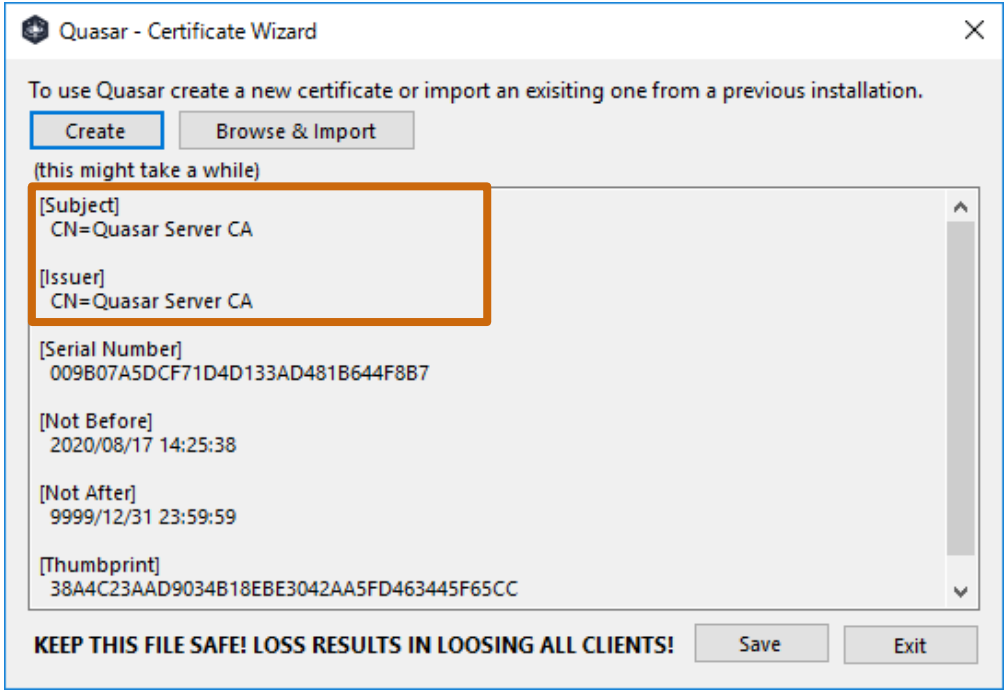
Whole Internet Scan with original scanner

- Find Quasar family C2 on internet.

Certificate created by the QuasarRAT builder

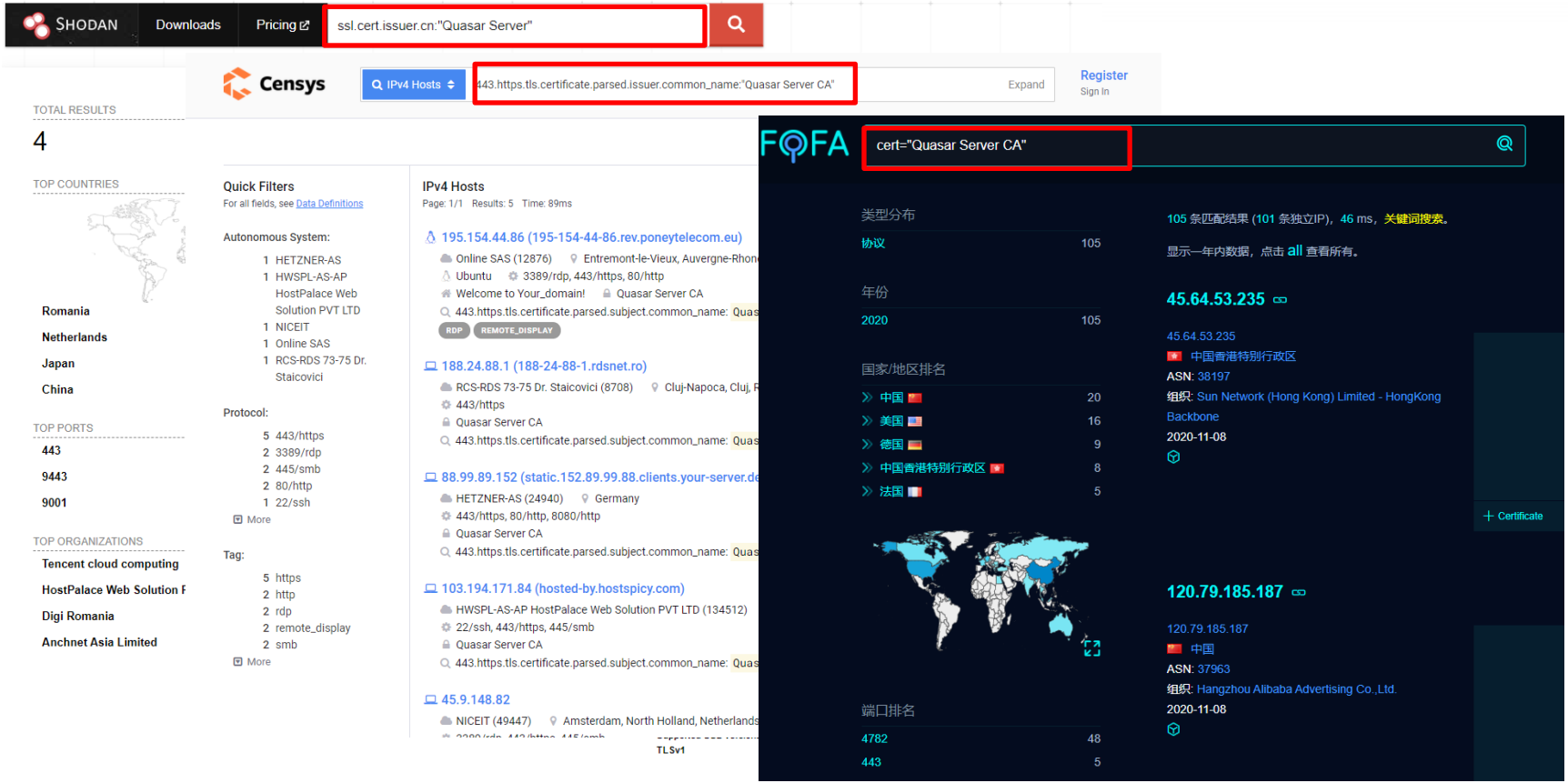
QuasarRAT 1.4

The certificate issuer contains the "Quasar Server CA“.



Search IoT Engine

QuasarRAT 1.4



Key Facts to Finding the Quasar Family C2

QuasarRAT 1.3

Malware name	Source code	Settings	Communication	In the wild
Quasar	https://github.com/quasar/QuasarRAT	-	-	Yes
Golden Edition	not published	Original	Original	Yes
XPCTRA	not published	Custom	Original	Yes
CinaRAT	https://github.com/wearelegal/CinaRAT	Original	Original	Yes
Xtremis 2.0	https://github.com/pa		Original	No
QuasarStrike	https://github.com/Q-		Original	No
VenomRAT	not published		Original	No
RSMaster	https://github.com/Ne		Original	No
Void-RAT	https://github.com/KadeDev/Void-RAT	Custom	Original	Yes
AsyncRAT	https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp	Custom	Original	Yes

Use the same communication method.

You can find all Quasar family C2 in the same method.

Key Facts to Finding the Quasar Family C2

QuasarRAT 1.3

Quasar Family Packet Format

Size(4byte)	HMAC(32byte)	IV(16byte)	QuickLZ + AES(mode CBC)

C2 First Response

```
@vmhost:~/svn/data/QuasarRAT$ xxd server_send_data_1st
00000000: 4000 0000 cc19 36aa db4e 7665 69c8 1095  @.....6..Nvei...
00000010: 5b72 38a1 34f4 7698 64a0 29fb c054 9df3  [r8.4.v.d.)..T..
00000020: c022 9f97 59e0 e268 40fb 3d84 37e0 f85c  ."..Y..h@.=.7..\
00000030: 5a83 5c63 9775 c651 0d32 1477 98d9 7239  Z.\c.u.Q.2.w..r9
00000040: c3d2 433a                                ..C:
```

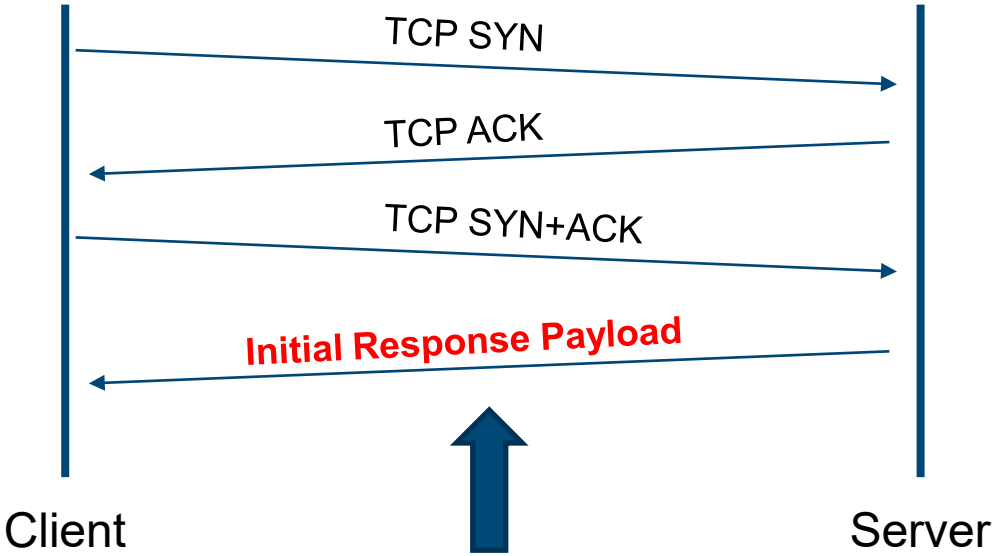
First response size = 0x44

First response header = 0x40 0x00 0x00 0x00

Key Facts to Finding the Quasar Family C2

QuasarRAT 1.3

Quasar Family Initial TCP Negotiation



Size: `0x44`

TCP payload header: `0x40 0x00 0x00 0x00`

Search Shodan

QuasarRAT 1.3

ts	ip	cc2	proto	port	hostnames	org	data
2020-04-04 03:19:41.435 56+00	108.160.130.94	JP	TCP	80	{108.160.130.94 .vultr.com}	Reliable Servers LLC	@¥x00¥x00¥x00¥x08¥xfc¥xaeQHX¥xdf¥xe9¥x82¥xab¥x8d¥x15&j¥xbc¥x0b=B¥xea~¥x1e¥x9c9¥x1cC¥x1dF¥xb2¥xf6¥x15¥x8c¥xc9¥x9f¥x94¥xf8¥x97¥xee¥xcc¥x1c¥xdb¥xe6C2!Q¥xc6¥xca¥xde¥xf5-Ah¥xec¥xd3¥x9f¥x8f¥xb77B/¥x0b¥x01¥x198
2020-03-16 02:30:56.020 369+00	198.13.54.11	JP	TCP	443	{www.google.com}	Vultr Holdings, LLC	@¥x00¥x00¥x00¥x8cT¥xc2@¥xca(0c¥x1fO¥x91¥xc2¥x00Ns¥xf0¥xd2z~Z¥x99¥x16¥xb5¥xcf¥xd0¥xc8¥x0be¥xed¥x18¥xfd¥xd4¥x03¥x03¥xfe¥xfc¥x11¥x03¥xa0:¥xfeM¥x8a¥x0b¥xf9¥xc9¥xe9¥xc6¥xa9&¥xc0n¥x82
2019-12-22 10:58:09.512 504+00	45.32.51.172	JP	TCP	8080	{45.32.51.172.vultr.com}	Choopa, LLC	@¥x00¥x00¥x00+¥xad+$¥x90¥'¥xaf¥xa5¥xa3¥x89¥x1b¥x7f¥xf4]G¥x08¥xdb/¥xab¥xe6¥x8dd¥x81¥xb6¥xfb¥xae¥xb8¥x1f¥xc5¥x8a¥x01;~¥xa9¥x91¥x1b¥xfb¥x17¥x9dn¥x0eYC¥xa9¥x8dy¥xe7&¥xb0¥xd1¥x9eQ¥xce>n¥xc5¥xf2¥x16¥x89¥xbbt|i¥xde
2019-12-29 15:12:49.552 671+00	167.179.78.239	JP	TCP	4782	{167.179.78.239 .vultr.com}	Choopa, LLC	@¥x00¥x00¥x00c}~¥x03¥xf6`¥t¥x06¥xd9h¥'¥¥¥xeb9¥xc0¥t¥xa4q¥xe9¥xc1¥x14¥xb3¥xc5$¥x19¥x0f¥x96G.¥x8a¥xad¥x84¥xcd¥x0e¥¥l¥x85h¥xf4&¥xe5_¥xf0T¥x0e¥xea¥x1b¥xe7Y¥x90S¥x90!¥xd4P¥x0b¥x0f+¥xb4¥x91,M¥xa8¥x1f

Quasar Servers on the Internet

QuasarRAT 1.3
QuasarRAT 1.4

76 Active Quasar Servers have Discovered on Nov 2020.

Count by types

Quasar Family	44
Quasar 1.4	32

TOP 5 Country

United Stats	15
China	12
Hong Kong	10
Russia	6
Netherlands	5

Bonus
- Analysis Tools -

Analysis Tools

MalConfScan

- Volatility plugin for extracts configuration data of known malware

quasarrat_decode.py

- Communication data decoding/encoding tool.

quasarrat_panel.py

- Dummy C2 that communicates with QuasarRAT.

quasarrat_client.py

- Client that communicates with QuasarRAT C2.

What is MalConfScan?

- **MalConfScan** is a Volatility plugin that extracts configuration data of known malware.
- Volatility is an open-source memory forensics framework for incident response and malware analysis.
- MalConfScan searches for malware in memory images and dumps configuration data.

MalConfScan with QuasarRAT

```
(malconfscan) [vmware@       malconfscan]$  sudo vol.py -f /mnt/        /Win7_64JP/Win7_64bit-JP-a66ad5ed.vmem --profile=Win7SP1x64
malconfscan -p 6660
Volatility Foundation Volatility Framework 2.6.1
[+] Searching memory by Yara rules.
[+] Detect malware by Yara rules.
[+]   Process Name      : a23c18.exe
[+]   Process ID        : 6660
[+]   Malware name      : Quasar
[+]   Base Address(VAD) : 0x1F80000
[+]   Size              : 0x57000
------------------------------------------------------------------------
Process: a23c18.exe (6660)

[Config Info]
VERSION              : 1.3.0.0
HOSTS                : mywinnetwork.ddns.net:80;
KEY (Base64)         : k4Ea0cde7hd7+Ytx2xhLHw==
AUTHKEY (Base64)     : R4WSD1GysgjiC0jFGVpFz3Y3gZ6n9yS5m5HqjtkK8xb7PW5GAGUkktId4YSptS9ffyKNQxWXQOrH6RIwL6Um7A==
SUBDIRECTORY         : SubDir
INSTALLNAME          : Client.exe
MUTEX                : QSR_MUTEX_RjrwbI1Icgf7POFqQF
STARTUPKEY           : Windows Session Manager
ENCRYPTIONKEY        : StbdXcw885BNQigDTAQD
TAG                  : Office04
LOGDIRECTORYNAME     : Logs
```

quasarrat_decode.py

Communication data decoding/encoding tool.

```
test@debian:~$ python3 quasarrat_decode.py -d client_send_data -k 1WvgEMPjdwf
qIMeM9MclyQ== -a NcFtjbDOcsw7Evd3coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj5ad2RO
UfX4QMscAIjYJdjrrs41+qcQwg==
[+] Decrypt mode.
[+] Hash check OK.
[+] Created client_send_data.decode
test@debian:~$ xxd client_send_data.decode
00000000: 3805 0455 7365 7208 0755 6e6b 6e6f 776e  8..User..Unknown
00000010: 0807 556e 6b6e 6f77 6e02 012d 4140 3336  ..Unknown..-A@36
00000020: 3646 3434 3833 4435 4431 3730 3438 4441  6F4483D5D17048DA
00000030: 4642 4534 4542 3535 4343 4338 3242 3938  FBE4EB55CCC82B98
00000040: 3139 4534 4542 3334 4443 4641 3632 3433  19E4EB34DCFA6243
00000050: 3041 4431 3041 3237 3746 4338 3343 001d  0AD10A277FC83C..
00000060: 1c57 696e 646f 7773 2031 3020 456e 7465  .Windows 10 Ente
00000070: 7270 7269 7365 2036 3420 4269 7410 0f57  rprise 64 Bit..W
00000080: 494e 3130 5f36 344a 502d 4f32 3031 0807  IN10_64JP-O201..
00000090: 556e 6b6e 6f77 6e09 084f 6666 6963 6530  Unknown..Office0
000000a0: 3406 056b 616e 7269 0807 312e 332e 302e  4..kanri..1.3.0.
000000b0: 30                                       0
test@debian:~$
test@debian:~$
```

quasarrat_panel.py

Dummy C2 that communicates with QuasarRAT.

```
test@debian:~$ sudo python3 quasarrat_panel.py -l -a Gr7icnpxp4j1oWFhmLnbIfe2f
nmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g== -k +PrSihNNK8M
pq4nK+spXzw== --apt10
[+] APT10 mode.
[+] Listen port 0.0.0.0:443.
[+] Get packet from ('210.144.230.100', 53
[+] Get data size: 260
[+] Hash check OK.
[+] Decoded data: b',\x05\x04UserA@277A0C1465A881FB4EA1447C92BF253AF63E337E522
C149A244C693A9FA4DD58\x1f\x1e210.144.230.100,169.254.30.206\x06\x05ja-JP\t\x08
16383 MB\x1d\x1cWindows 10 Enterprise 64 Bit\x10\x0fDESKTOP-51VDEJN\x01\x06\x0
5kanri\x12\x08\x072.0.0.0'
```

Receive data from client.

quasarrat_client.py

Client that communicates with QuasarRAT C2.

```
test@debian:~$ python3 quasarrat_client.py -s 210.144.121.100 -p 4782 -k
coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj5ad2ROUfX4QMscAIjYJdjrrs41+qcQwg==
[+] Thread count 1.
[+] Connect port 210.144.121.100:4782.
[+] Get data size: 68
[+] Hash check OK.
[+] Command: GetAuthentication
[+] Decoded data: b'j'
[+] Send data.
[+] Send data size: 212
[+] Get data size: 68
[+] Hash check OK.
[+] Command: SetAuthenticationSuccess
[+] Decoded data: b';'
[+] Get data size: 68
[+] Hash check OK
```

quasarrat_client.py

```
test@debian:~$ python3 quasarrat_client.py -s 210.144.121.100 -p 4782 -k 1WvgEMPj
coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj5ad2ROUfX4QMscAIjYJdjrrs41+qcQwg== -t Offic
[+] Thread count 1.
[+] Connect port 210.144.121.100:4782.
[+] Get data size: 68
[+] Hash check OK.
[+] Command: GetAuthentication
[+] Decoded data: b'j'
[+] Send data.
[+] Send data size: 228
[+] Get data size: 68
[+] Hash check OK.
[+] Command: SetAuthenticationSuccess
[+] Decoded data: b';'
[+] Get data size: 84
[+] Hash check OK.
[+] Command: DoShellExecute
[+] Decoded data: b'P\x07\x06whoami'
```

Receive command to execute shell command from C2.

Bonus
- Counter Attack -

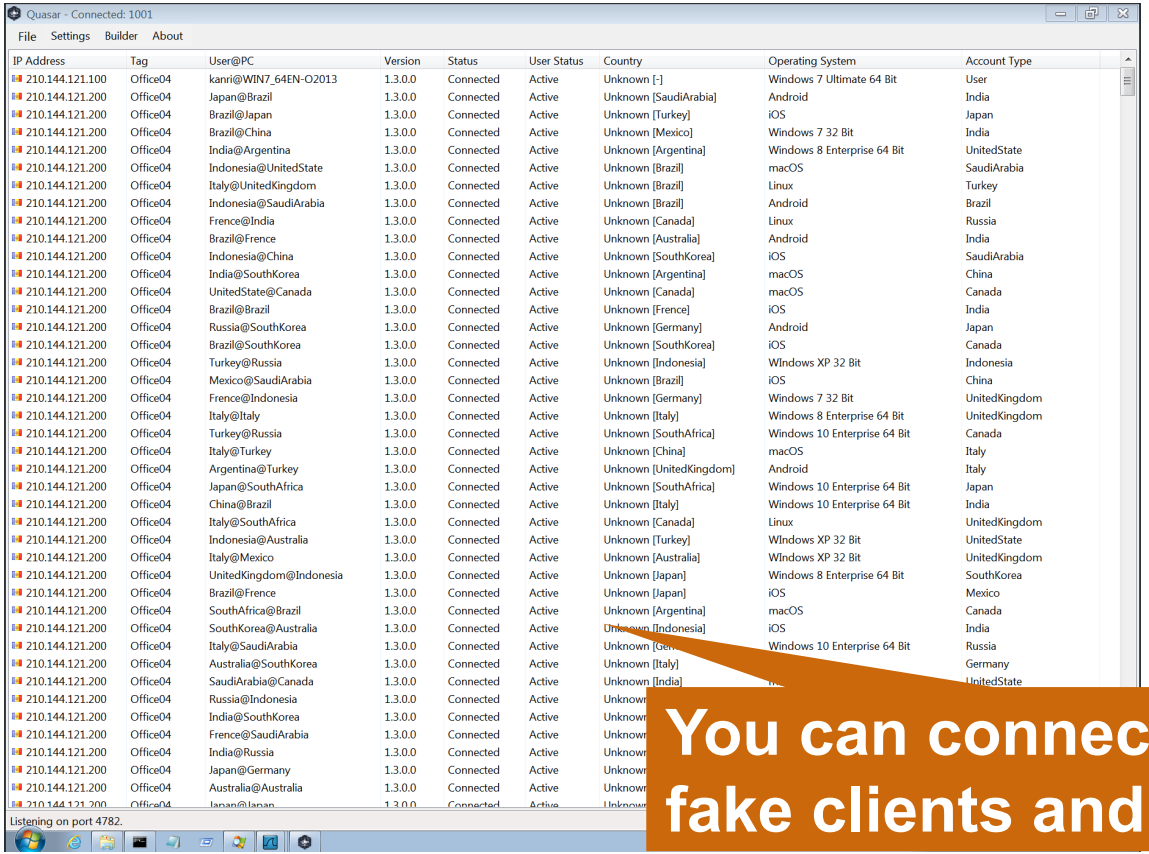
Fake Malware DoS Attack for C2

By cloning the communication of QuasarRAT, you can make a DoS attack on the server.

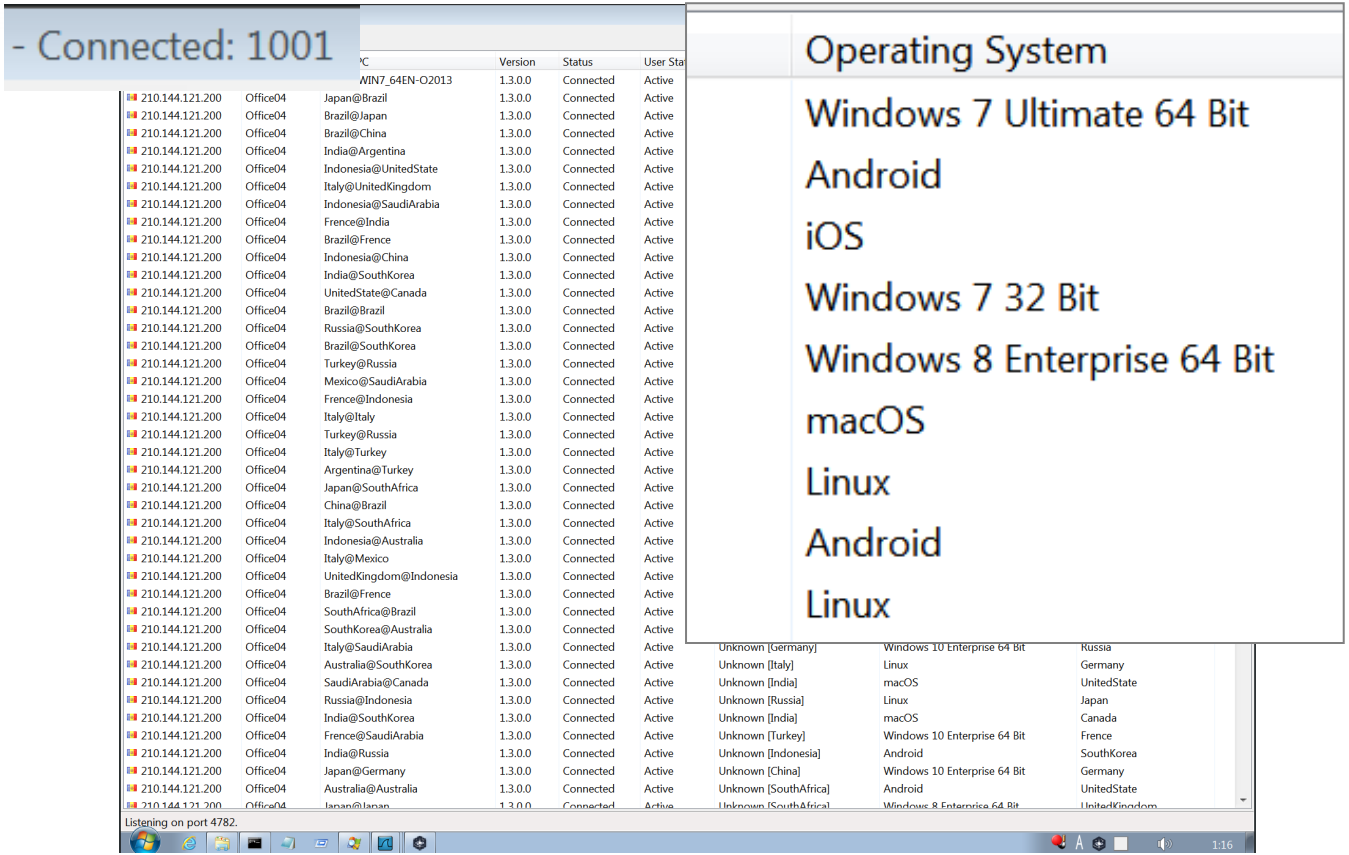
```
$ python quasarrat_client.py  -h
usage: quasarrat_client.py [-h] [-k KEY] [-a AUTHKEY] [--apt10] [-s SERVER]
                           [-p PORT] [-t TAG] [-c COUNT]

QuasarRAT panel scanner.

optional arguments:
  -h, --help            show this help message and exit
  -k KEY, --key KEY     Encryption or decryption key
  -a AUTHKEY, --authkey AUTHKEY
                        Authkey. (Base64 data)
  --apt10               Customized APT10 mode.
  -s SERVER, --server SERVER
                        Server IP address. (default: 127
  -p PORT, --port PORT  Server port. (default: 443)
  -t TAG, --TAG TAG     Tag value.
  -c COUNT, --count COUNT
                        Scan count. (Default: 1)
```

The COUNT option is the number of fake clients.

Fake Malware DoS Attack for C2

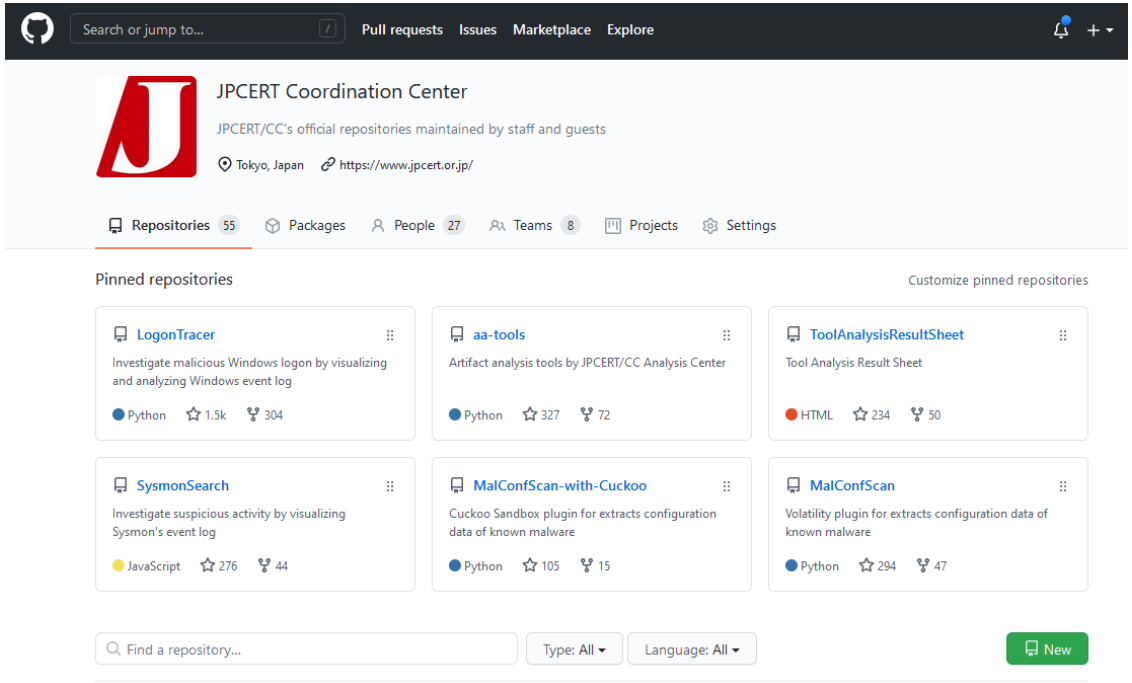


Fake Malware DoS Attack for C2



How to Download

JPCERT/CC Github



https://github.com/JPCERTCC/QuasarRAT-Analysis

Takeaways

Understand how variants are created from one open source RAT

Detailed analysis results of QuasarRAT and family

Case study of how to hunt malware family


JPCERT CC®
Thank you!
@jpcert_en
ir-info@jpcert.or.jp
PGP https://www.jpcert.or.jp/english/pgp/