

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.

The screenshot shows the GitHub repository page for 'quasar / QuasarRAT'. The page title is 'Remote Administration Tool for Windows'. Key statistics displayed include 1,249 commits, 2 branches, 0 packages, 9 releases, 14 contributors, and an MIT license. The repository has 330 stars, 2.8k forks, and 1.2k issues. A pull request from 'MaxXor' was merged, simplifying networking code and updating dependencies. Other recent commits include updates to issue templates, licenses, and client/test components.

Commit	Description	Date
MaxXor	Simplify networking code and update dependencies	24 Apr
.github/ISSUE_TEMPLATE	Add issue templates	last year
Licenses	Add BouncyCastle license	last year
Quasar.Client.Tests	More refactoring	last year
Quasar.Client	Simplify networking code and update dependencies	7 months ago
Quasar.Common.Tests	Add basic TLS support	last year
Quasar.Common	Simplify networking code and update dependencies	7 months ago
Quasar.Server.Tests	Adjust namespaces	last year

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.

Threat Spotlight
MenuPass/Quasar RAT Backdoor

ThreatVector > Research & Intelligence

by The Cyance Threat Research Team | June 10, 2019

Introduction

During the latter half of 2018, BlackBerry Cyance researchers identified multiple spear phishing attacks targeting companies from several verticals across the globe. These attacks were related to the MenuPass (a.k.a. APT10/Stone) open-source backdoor named QuasarRAT to achieve persistence. Unit 42 researchers identified several distinct loader variants tailored to machine learning (ML) to analyse our malware corpus. We saw the wild since late 2018, roughly coinciding with the MenuPass group.

Home > FireEye Blogs > Threat Research > Spear Phishing Campaign Targeting Ukraine; Infrastructure Related to Luhansk People's Republic

Threat Research

Spear Phishing Campaign Targets Ukraine; Infrastructure Related to Luhansk People's Republic

April 16, 2019 | by John Hultquist, Ben Read, Oleg Bondarenko

GOVERNMENT, UKRAINE, SPEAR PHISHING

In early 2019, FireEye Threat Intelligence identified a spear phishing campaign targeting Ukraine. The spear phishing email included a malicious link to a landing page that stage payload from the command and control (C&C) server. The C&C server was located in Ukraine and included lure content related to the sale of diamonds.

This latest activity is a continuation of spear phishing that has been targeting Ukraine. The email is linked to activity that previously targeted the Luhansk People's Republic (LPR).

The spear phishing email, sent on Jan. 22, 2019, used the subject "STANDARD," and the sender was forged as Armtrac, a de-

SPEC-20T-MK2-000-ISS

Armtrac

SPEC-20T-MK2-000-ISS-4.10-09-2018-STANDARD

Komy:

Tools Playbooks Speaking Events About Us

The Gorgon Group: Slithering State and Cybercrime

27,759 people reacted 5 17 min. read

By Robert Falcone, David Fuertes, Josh Grunzweig and others on August 2, 2018 at 5:00 AM Category: Unit 42 Tags: CVE-2017-0199, Gorgon Group, Subaat

unit42 THREAT RESEARCH

paloalto

This post is also available in: 日本語 (Japanese)

Unit 42 researchers have been tracking Subaat, an actor that has been renewed targeted attack activity. Part of monitoring Subaat is identifying the larger crew of individuals responsible for carrying out these attacks against organizations. Technical analysis on some of the attack samples have been depicted by 360 and Tuisec, in which four Unit 42 researchers have been tracking, which we are doing so far. In addition to the numerous targeted attacks, Unit 42

PRODUCTS SERVICES

BLOG

Patchwork APT Group Targets US Think Tanks

JUNE 7, 2018

by Matthew Meltzer, Sean Koessel, Steven Adair

[Facebook](#) [Twitter](#) [Email](#)

PATCHWORK APT GROUP

US THINK TANK CAMPAIGN

- ① Strategic Spear Phishing
- ② Weaponized RTF Document
- ③ CVE-2017-8750 Exploitation
- ④ QuasarRAT Variant Payload

In March and April 2018, Volexity identified multiple spear phishing campaigns attributed to Patchwork, an Indian APT group also known as Dropping Elephant. This increase in threat activity was consistent with other observations documented over the last few months in blogs by 360 Threat Intelligence Center analyzing attacks on Chinese organizations and Trend Micro noting targets in South Asia. From the attacks observed by Volexity, what is most notable is that Patchwork has

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

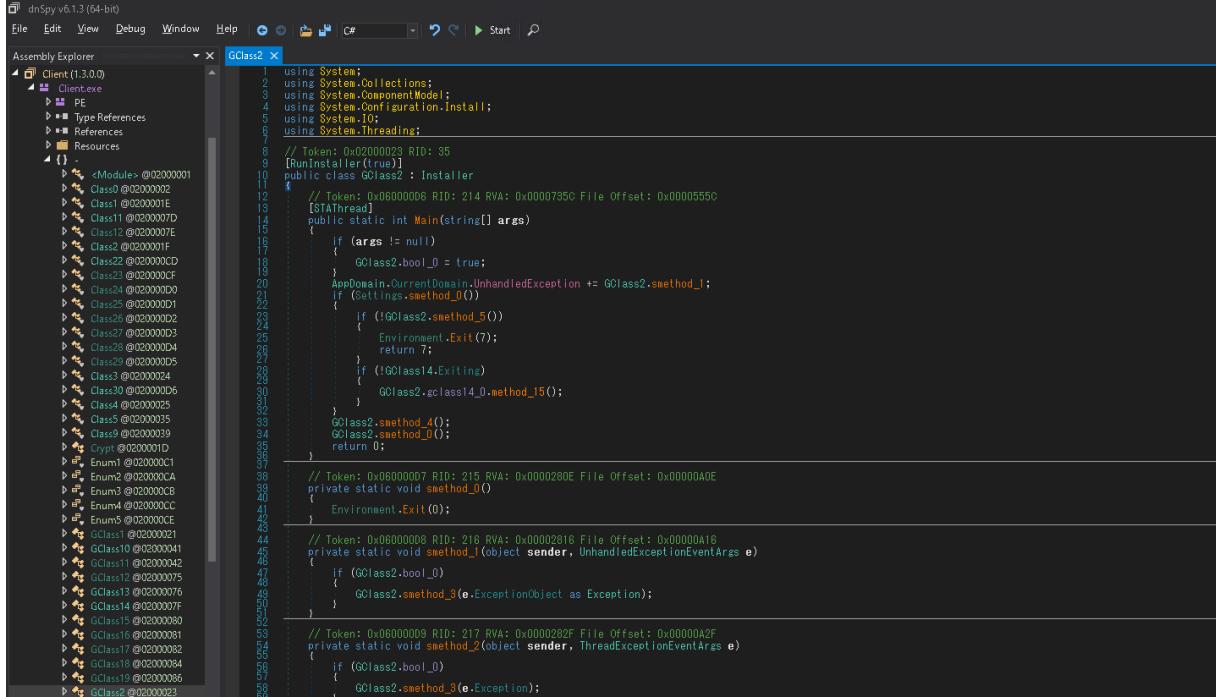
Campaigns using QuasarRAT

4

Hunting Quasar Family C2

QuasarRAT Internals

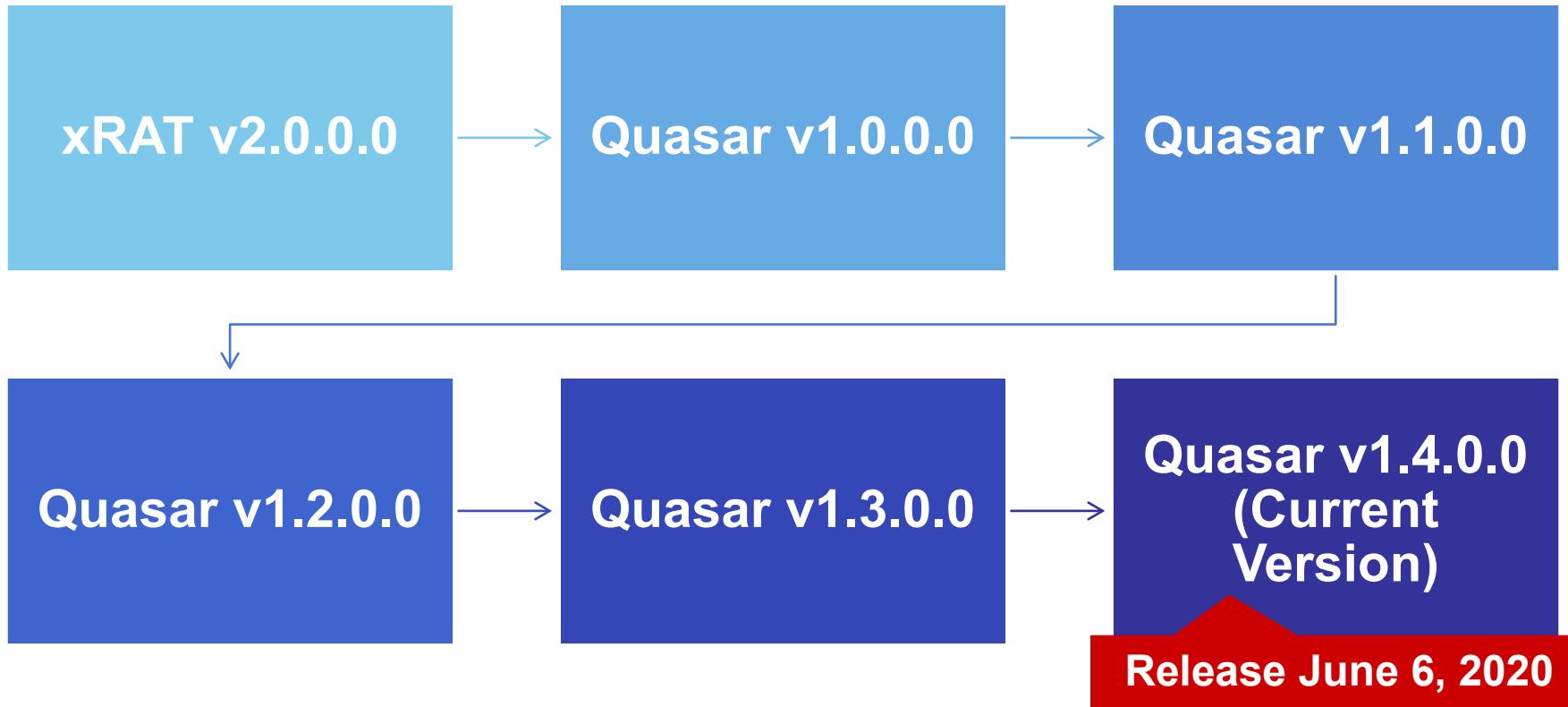
QuasarRAT is coded in C#.



The screenshot shows the dnSpy interface with the Assembly Explorer on the left and the code editor on the right. The code editor displays the following C# code for the GClass2 class:

```
1  using System;
2  using System.Collections;
3  using System.ComponentModel;
4  using System.Configuration.Install;
5  using System.IO;
6  using System.Threading;
7
8  // Token: 0x02000023 RID: 35
9  [RunInstaller(true)]
10 public class GClass2 : Installer
11 {
12     // Token: 0x02000006 RID: 214 RVA: 0x0000735C File Offset: 0x00005550
13     [STAThread]
14     public static int Main(string[] args)
15     {
16         if (args != null)
17         {
18             GClass2.bool_0 = true;
19         }
20         AppDomain.CurrentDomain.UnhandledException += GClass2.smethod_1;
21         if (Settings.smethod_0())
22         {
23             if (!GClass2.smethod_5())
24             {
25                 Environment.Exit(7);
26             }
27             if (!GClass14.Exiting)
28             {
29                 GClass2.smethod_15();
30             }
31         }
32         GClass2.smethod_4();
33         GClass2.smethod_0();
34         return 0;
35     }
36
37     // Token: 0x020000C1 RID: 193 RVA: 0x0000280E File Offset: 0x00000A0E
38     private static void smethod_0()
39     {
40         Environment.Exit(0);
41     }
42
43     // Token: 0x02000008 RID: 216 RVA: 0x00000A16
44     private static void smethod_1(object sender, UnhandledExceptionEventArgs e)
45     {
46         if (GClass2.bool_0)
47         {
48             GClass2.smethod_3(e.ExceptionObject as Exception);
49         }
50     }
51
52     // Token: 0x02000009 RID: 217 RVA: 0x00000A2F
53     private static void smethod_2(object sender, ThreadExceptionEventArgs e)
54     {
55         if (GClass2.bool_0)
56         {
57             GClass2.smethod_3(e.Exception);
58         }
59     }
60 }
```

QuasarRAT Version History



Communication

xRAT 2.0.0.0

Size(4byte)

IV(16byte)

AES(mode CBC)

```
99 // Token: 0x0600057E RID: 1406 RVA: 0x00015890 File Offset: 0x00013B90
100 public static string Decrypt(string input, string keyy)
101 {
102     string result;
103     try
104     {
105         byte[] key;
106         using (MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider())
107         {
108             key = md5CryptoServiceProvider.ComputeHash(Encoding.UTF8.GetBytes(keyy));
109         }
110         byte[] array2;
111         int count;
112         using (MemoryStream memoryStream = new MemoryStream(Convert.FromBase64String(input)))
113         {
114             using (RijndaelManaged rijndaelManaged = new RijndaelManaged())
115             {
116                 byte[] array = new byte[16];
117                 memoryStream.Read(array, 0, 16);
118                 rijndaelManaged.IV = array;
119                 rijndaelManaged.Key = key;
120                 using (CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateDecryptor(),
121                 {
122                     array2 = new byte[memoryStream.Length - 16L + 1L];
123                     count = cryptoStream.Read(array2, 0, array2.Length);
124                 }
125             )
126         }
127         result = Encoding.UTF8.GetString(array2, 0, count);
128     }
129     catch
130     {
131         result = string.Empty;
132     }
133     finally
134     {
135         byte[] array2 = null;
136         byte[] key = null;
137     }
138     return result;
139 }
```



Command(1byte)

argv

AES_KEY = md5(ENCRYPTIONKEY)

Communication Format

QuasarRAT 1.3.0.0

```
143 public static byte[] decrypt_data(byte[] input)
144 {
145     if (Crypto.byte_0 == null || Crypto.byte_0.Length == 0)
146     {
147         throw new Exception("Key can not be empty.");
148     }
149     if (input == null && input.Length != 0)
150     {
151         byte[] array = new byte[0];
152         try
153         {
154             using (MemoryStream memoryStream = new MemoryStream(input))
155             {
156                 using (AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider())
157                 {
158                     aesCryptoServiceProvider.KeySize = 128;
159                     aesCryptoServiceProvider.BlockSize = 128;
160                     aesCryptoServiceProvider.Mode = CipherMode.CBC;
161                     aesCryptoServiceProvider.Padding = PaddingMode.PKCS7;
162                     aesCryptoServiceProvider.Key = Crypto.byte_0;
163                     using (HMACSHA256 hmacsha = new HMACSHA256(Crypto.byte_1))
164                     {
165                         byte[] a = hmacsha.ComputeHash(memoryStream.ToArray(), 32, memoryStream.ToArray().Length - 32);
166                         byte[] array2 = new byte[32];
167                         memoryStream.Read(array2, 0, array2.Length);
168                         if (!Golos7.smethod_0(a, array2))
169                         {
170                             return array;
171                         }
172                         byte[] array3 = new byte[16];
173                         memoryStream.Read(array3, 0, 16);
174                         aesCryptoServiceProvider.IV = array3;
175                         using (CryptoStream cryptoStream = new CryptoStream(memoryStream, aesCryptoServiceProvider.CreateDecryptor(), CryptoStreamMode.Read))
176                         {
177                             byte[] array4 = new byte[memoryStream.Length - 16L + 1L];
178                             array = new byte[cryptoStream.Read(array4, 0, array4.Length)];
179                             Buffer.BlockCopy(array4, 0, array, 0, array.Length);
180                         }
181                     }
182                 }
183             }
184         }
185         catch
186         {
187         }
188     }
189     return array;
190 }
191 throw new ArgumentException("Input can not be empty.");
192 }
```

* QuickLZ compress level3

Size(4byte)

HMAC(32byte)

IV(16byte)

QuickLZ + AES(mode CBC)

Command(1byte)

argv

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

Communication Format

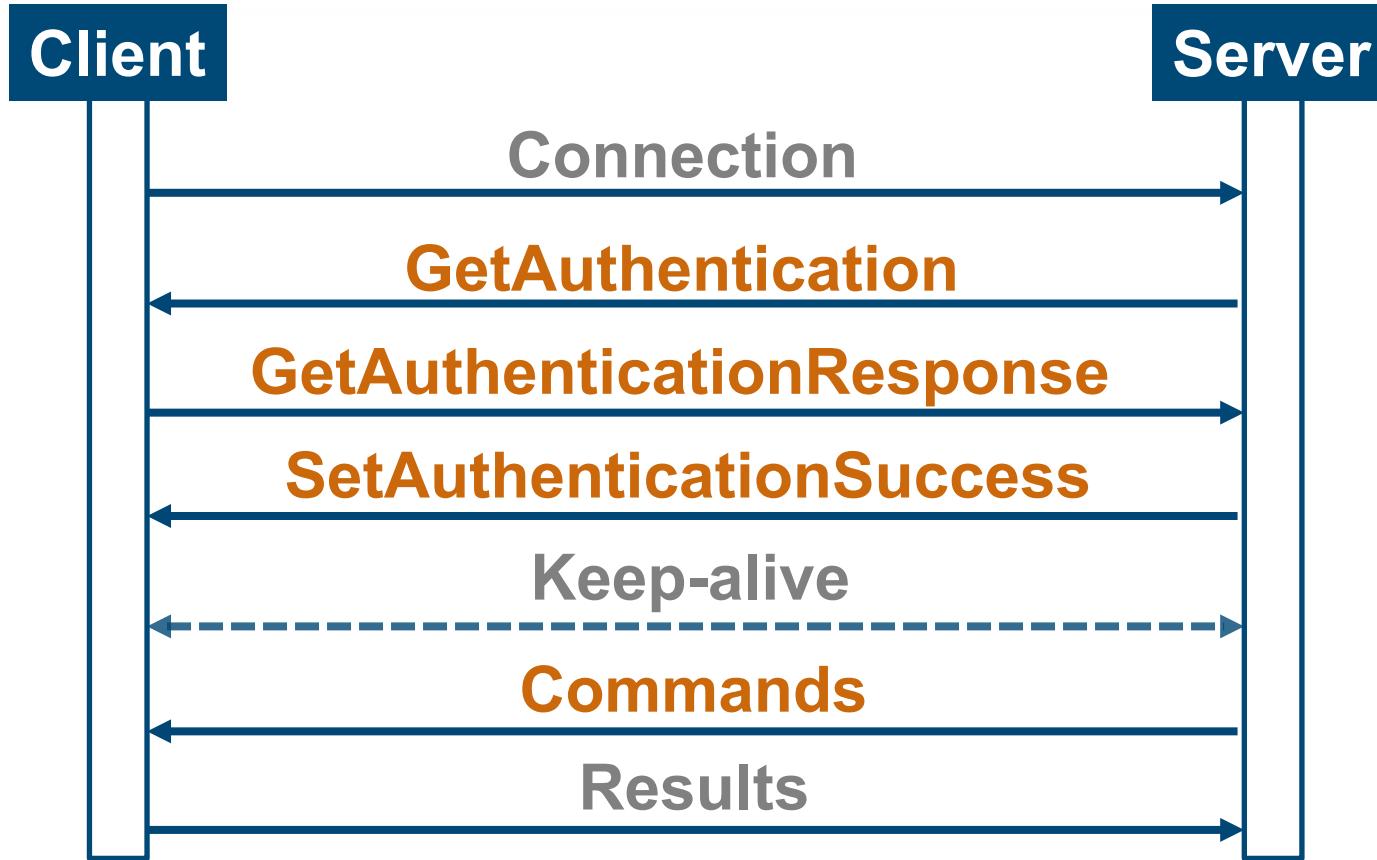
QuasarRAT 1.4.0.0

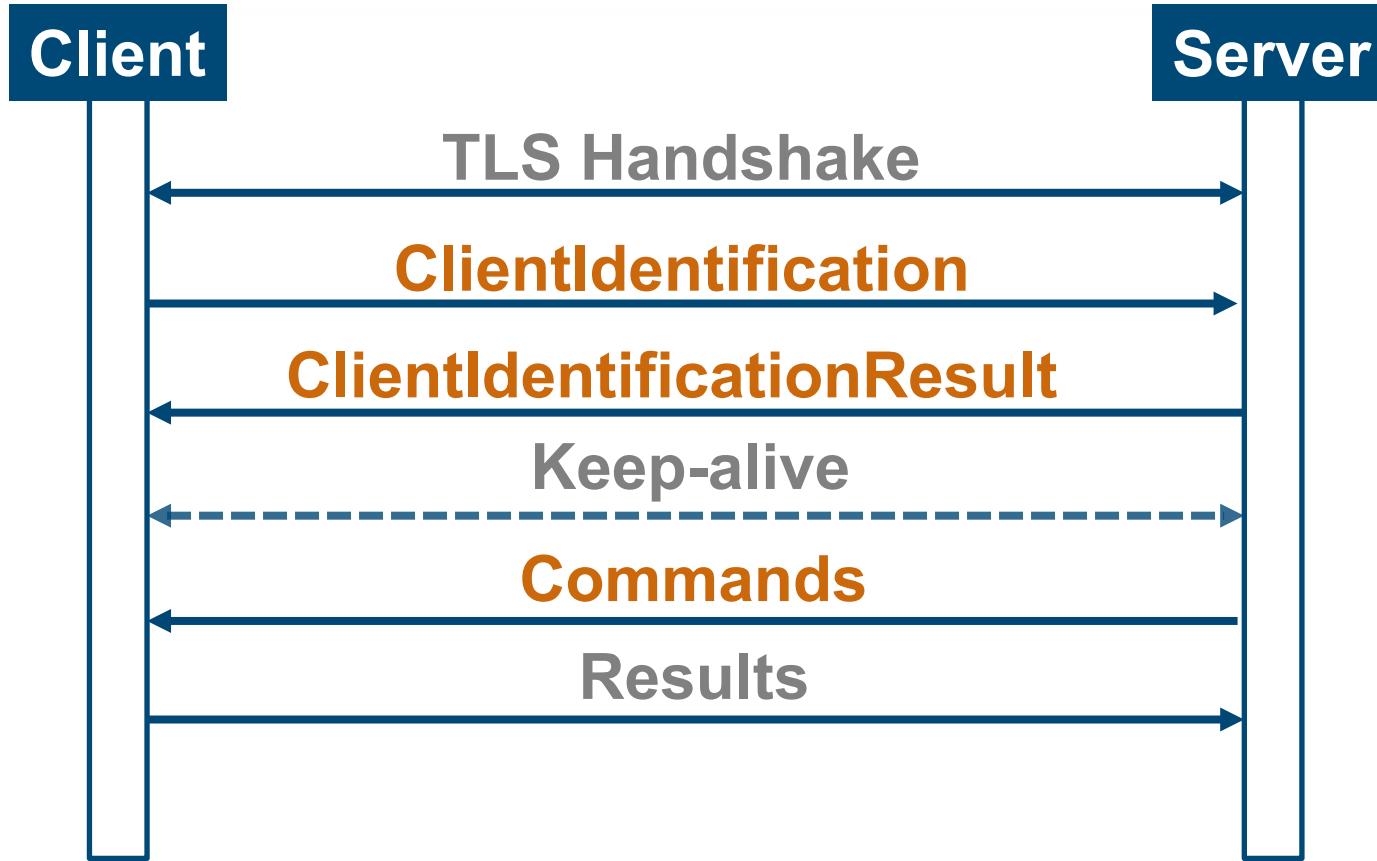
* Not use AES + QuickLZ

Command (ProtoBuf)



TLS





Configuration

xRAT

```
21 // Token: 0x0400019C RID: 412
22 public static string Version = "MuchfhtzNzb0II0yn8I4gEoQ1Q306NftZEnYCSfG0+jx1tBjs0i7fIC4u//Boun";
23
24 // Token: 0x0400019D RID: 413
25 public static string Host = "gtIwo0UG8NwIDphZ4Djf24Z11k8IMyFubqcbznA7Sw=";
26
27 // Token: 0x0400019E RID: 414
28 public static ushort Port = 1312;
29
30 // Token: 0x0400019F RID: 415
31 public static int Reconnectdelay = 5000;
32
33 // Token: 0x040001A0 RID: 416
34 public static string Password = "y+CxG6V8fhj3D43N5VJeFcAzu9LzsERJCPxyn+tpogNkF8I1hWSX1peY6ve0wSR215QB5U9nvM9UySDsN
35
36 // Token: 0x040001A1 RID: 417
37 public static string Dir = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);
38
39 // Token: 0x040001A2 RID: 418
40 public static string Subfolder = "dsGnkIxnh9R5I8B2Ag+6gtvNidaCoZ2YDAKZ+QGdw=";
41
42 // Token: 0x040001A3 RID: 419
43 public static string Installname = "lvzieirDIDtVp+X8KDQJvwmGI6bE2GQzv//7FBCu920c=";
44
45 // Token: 0x040001A4 RID: 420
46 public static bool Install = false;
47
48 // Token: 0x040001A5 RID: 421
49 public static bool Startup = false;
50
51 // Token: 0x040001A6 RID: 422
```

Base64



AES_KEY = md5(ENCRYPTIONKEY)

IV(16byte)

AES(mode CBC)

Configuration

QuasarRAT

```
45 // Token: 0x04000008 RID: 8
46 public static string VERSION = "UvADP/X6YS4Pi/TtR6RCQ49f9HntDsrnWcHNg4Sp7M9cq0JryNboozr0AdD2e39pa5nVY Ae1k9E5+dv0GMw=";
47 // Token: 0x04000009 RID: 9
48 public static string HOSTS = "Tqu7dmK/Zlsz66TL510uE8/L0hIri5uxAat iTHCs4hvRJd5xp4c035SrBtDLtrndk+sDe2ewmNbSJWjue8HEoms8v0d8aFt I28Dlw4QsEkk5X";
49
50 // Token: 0x0400000A RID: 10
51 public static int RECONNECTDELAY = 3000;
52
53 // Token: 0x0400000B RID: 11
54 public static string KEY = "ebd0sS6Q89sUEWPiItdABQ=";
55
56 // Token: 0x0400000C RID: 12
57 public static string AUTHKEY = "SqC2h9nanECTE5yMFKy1PRF90HS1bZLcdE0ccui1HMsbDR80X07Fv9X0KcDJAU/mouBB4/KAEpaIUM9KJEJZBQ=";
58
59 // Token: 0x0400000D RID: 13
60 public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;
61
62 // Token: 0x0400000E RID: 14
63 public static string DIRECTORY = Environment.GetFolderPath(Init_config.specialFolder_0);
64
65 // Token: 0x0400000F RID: 15
66 public static string SUBDIRECTORY = "di7ALK0A1S1Pt5o4bbnsEuESGLSgwUEYBMBMHkzTrR9Suuw#6dmfsS9Z1Xlyrt318xpkwScvpuQA9aFCGCVa=";
67
68 // Token: 0x04000010 RID: 16
69 public static string INSTALLNAME = "/np6aAFTWlsNmJIk0C428HV7MCpkYc+gwJURNOYibATkpC00pq19ablwFMLXptV/cDEbdJaYbN/Zfs2B3/JSBJQ=";
70
71 // Token: 0x04000011 RID: 17
72 public static bool INSTALL = false;
73
74 // Token: 0x04000012 RID: 18
75 public static bool STARTUP = false;
76
77 // Token: 0x04000013 RID: 19
78 public static string MUXFEX = "X1T4erV+ttsrcFJxWIPtQllveVzcL157988Aa0T08g1Rt00eFDW1v095F1237.IFFURuJNTBLcxHkMayYPERoqLQMVITWE/f25p0tXESORxwA=";
```

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
RECONNECTDELAY	TAG 1.3
KEY	LOGDIRECTORY 1.3
AUTHKEY 1.3	SERVERSIGNATURE 1.4
DIRECTORY	SERVERCERTIFICATESTR 1.4
SUBDIRECTORY	SERVERTCERTIFICATE 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

```
7  public class Commands
8  {
9      // Token: 0x06000377 RID: 887
10     public static Type[] command()
11     {
12         return new Type[]
13         {
14             typeof(GetAuthentication),
15             typeof(DoClientDisconnect),
16             typeof(DoClientReconnect),
17             typeof(DoClientUninstall),
18             typeof(DoWebcamStop),
19             typeof(DoAskElevate),
20             typeof(DoDownloadAndExecute),
21             typeof(DoUploadAndExecute),
22             typeof(GetDesktop),
23             typeof(GetProcesses),
24             typeof(DoProcessKill),
25             typeof(DoProcessStart),
26             typeof(DoDrives),
27             typeof(GetDirectory),
28             typeof(DoDownloadFile),
29             typeof(DoMouseEvent),
30             typeof(DoKeyboardEvent),
31             typeof(GetSystemInfo),
32             typeof(DoVisitWebsite),
33             typeof(DoShowMessageBox),
34             typeof(DoClientUpdate),
35             typeof(GetMonitors),
36             typeof(GetWebcams),
37             typeof(GetWebcam),
38             typeof(DoShellExecute),
39             typeof(DoPathRename),
40             typeof(DoPathDelete),
41             typeof(DoShutdownAction),
42             typeof(GetStartupItems),
43             typeof(DoStartupItemAdd),
44             typeof(DoStartupItemRemove),
45             typeof(DoDownloadFileCancel),
46             typeof(GetLoggerkeyLogs),
47             typeof(DoUploadFile),
48             typeof(GetPasswords),
49             typeof(DoLoadRegistryKey),
50             typeof(DoCreateRegistryKey)
51         };
52     }
53 }
```

Define the
commands with
typeof.

A structure is
defined in the
executable file.

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

Decoded QuasarRAT traffic

```
keybaordinjector.py
test@debian:~$ python3 quasarrat_client.py -s
coMC0y4koy/SRZGydhNmno81Z0W0vdfg7sv0Cj5ad2ROU
[+] 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

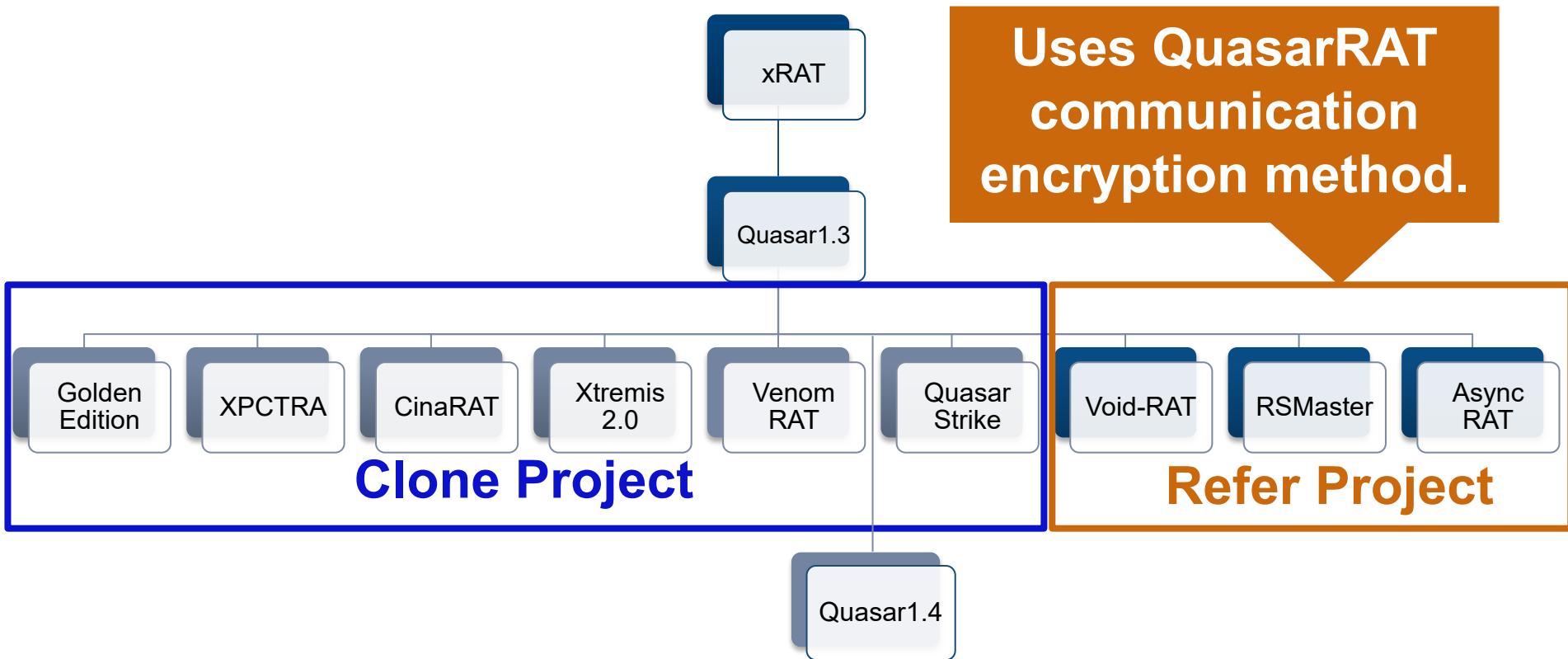
3

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 is the banking malware in Brazil.

XPCTRA Config

```
11  public static class Settings
12  {
13      public static string VERSION = "5.0";
14      public static string HOSTS = "coca.cheddarmcmelt.top:8799";
15      public static int RECONNECTDELAY = 500;
16      public static object[] HOSTS2 = new object[2]
17      {
18          (object) "coca.cheddarmcmelt.top",
19          (object) 222
20      };
21      public static string PhpHttps = "http://fritas.cheddarmcmelt.top/master/PhpTrafico.php";
22      public static string PhpSeParador = "http://fritas.cheddarmcmelt.top/master/Controle.php";
23      public static bool ConectandoMercadobitcoin = true;
24      public static bool FormPin = false;
25      public static bool ConectandoBlockChain = true;
26      public static bool FormBlockchain = false;
27      public static bool ConectandoNeteller = true;
28      public static bool FormNetller = false;
29      public static bool Perfecmoney = true;
30      public static string LinkHtmlSpam = "http://fritas.cheddarmcmelt.top/master/conf/Html.txt";
31      public static string KEY = "1WvgEMPjdwfqIMeM9MclyQ==";
32      public static string AUTHKEY = "NcFtjbD0csw7Ev3coMC0y4koy/SRZGydhNmno81Z0W0vdfg7sv0Cj";
33      public static Environment.SpecialFolder SPECIALFOLDER = Environment.SpecialFolder.ApplicationData;
34      public static string DIRECTORY = Environment.GetFolderPath(Settings.SPECIALFOLDER);
35      public static string MUTEX = "123Ak82kA,yAo2kAlUS2kYkala!";
36      public static string TAG = "Infectado";
37
38      public static bool Initialize()
39      {
40          return true;
41      }
42  }
```

QuasarRAT Config

```
10  public static class Settings
11  {
12      #if DEBUG
13          public static string VERSION = System.Windows.Forms.Application.ProductVersion;
14          public static string HOSTS = "localhost:4782";
15          public static int RECONNECTDELAY = 500;
16          public static string KEY = "1WvgEMPjdwfqIMeM9MclyQ==";
17          public static string AUTHKEY = "NcFtjbD0csw7Ev3coMC0y4koy/SRZGydhNmno81Z0W0vdfg7sv0Cj";
18          public static Environment.SpecialFolder SPECIALFOLDER = Environment.SpecialFolder.ApplicationData;
19          public static string DIRECTORY = Environment.GetFolderPath(Settings.SPECIALFOLDER);
20          public static string SUBDIRECTORY = "Test";
21          public static string INSTALLNAME = "test.exe";
22          public static bool INSTALL = false;
23          public static bool STARTUP = false;
24          public static string MUXTEX = "123Ak82kA,yAo2kAlUS2kYkala!";
25          public static string STARTUPKEY = "Test key";
26          public static bool HIDEFILE = false;
27          public static bool ENABLELOGGER = false;
28          public static string TAG = "DEBUG";
29          public static string LOGDIRECTORYNAME = "Logs";
30          public static bool HIDELOGDIRECTORY = false;
31          public static bool HIDEINSTALLSUBDIRECTORY = false;
32
33          public static bool Initialize()
34          {
35              FixDirectory();
36              return true;
37          }
38  }
```

XPCTRA is the banking malware in Brazil.

XPCTRA Commands

```
12 namespace VERMELHO265MONARCA.Core.Packets
13 {
14     public class PacketRegistry
15     {
16         public static Type[] GetPacketTypes()
17         {
18             return new Type[76]
19             {
20                 typeof (GetAuthentication),
21                 typeof (DoClientDisconnect),
22                 typeof (DoClientReconnect),
23                 typeof (DoClientUninstall),
24                 typeof (DoWebcamStop),
25                 typeof (DoAskElevate),
26                 typeof (DoDownloadAndExecute),
27                 typeof (DoUploadAndExecute),
28                 typeof (GetDesktop),
29                 typeof (GetProcesses),
30                 typeof (DoProcessKill),
31                 typeof (DoProcessStart),
32                 typeof (GetDrives),
33                 typeof (GetDirectory),
34                 typeof (DoDownloadFile),
35                 typeof (DoMouseEvent),
36                 typeof (DoKeyboardEvent),
37                 typeof (GetSystemInfo),
38                 typeof (DoVisitWebsite),
39                 typeof (DoShowMessageBox),
40                 typeof (DoClientUpdate),
41                 typeof (GetMonitors),
42                 typeof (GetWebcams),
43                 typeof (GetWebcam),
44                 typeof (DoShellExecute),
45                 typeof (DoPathRename),
```

QuasarRAT Commands

```
6  namespace xClient.Core.Packets
7  {
8      public class PacketRegistry
9      {
10         public static Type[] GetPacketTypes()
11         {
12             return new Type[]
13             {
```

Added Commands

```
90             typeof (GetChangeRegistryValueResponse),
91             typeof (FAROFA785MANOBRA),
92             typeof (FAROFA785MANOBRAResponse),
93             typeof (ANGUSTIADO349MONITORIA),
94             typeof (ReverseProxyDisconnect),
95             typeof (GetConnectionsResponse)
```

```
96             typeof (Packets.ServerPackets.DoShellExecute),
97             typeof (Packets.ServerPackets.DoPathRename),
98             typeof (Packets.ServerPackets.DoPathDelete),
99             typeof (Packets.ServerPackets.DoShutdownAction),
```

Golden Edition

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

Leak QuasarRAT Golden Edition - Best free RAT - DOWNLOAD

Author: CrackStar · Message · 01-10-2018, 01:44 AM #1

QuasarRAT Golden Edition

Hi guys, I'll post the 1.4.1.0 version of QuasarRAT.

The file is completely clean, if it is reported as a virus it is normal because it is a hacking program.
Have fun and do not make it a bad use.
The download link can be found after reply this topic.

DOWNLOAD LINK:

Code:
<https://goo.gl/gIMoyh>

(This post was last modified: 01-10-2018, 12:52 PM by CrackStar.)

Golden Edition

2362d50341e0d12f7f24b824af3824005ec2057010f33fadabaef87106c4d84

46 / 72

① 46 engines detected this file

2362d50341e0d12f7f24b824af3824005ec2057010f33fadabaef87106c4d84
Windows Security Health Microsoft
4.61 MB | 2020-02-17 12:44:51 UTC | 2 months ago | EXE

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Ad-Aware	① Trojan.GenericKD.33197166	AegisLab	① Trojan MSIL_Quasar IIc
AhnLab-V3	① Malware/Win32.Generic.C3353243	Alibaba	[+] Searching memory by Yara rules. [+] Detect malware by Yara rules.
ALYac	① Trojan.GenericKD.33197166	Arcabit	[+] Process Name : rsios.exe [+] Process ID : 4056 [+] Malware name : Quasar [+] Base Address(VAD) : 0x400000 [+] Size : 0x486000
Avast	① Win32:Malware-gen	AVG	
Avira (no cloud)	① TR/Spy.Quasar.vqxpmp	BitDefender	
BitDefenderTheta	① Gen.NN.ZexxF.34090 @B3@aCqIYQg	Bkav	Process: rsios.exe (4056)
CAT-QuickHeal	① TrojanSpy.MSIL	CrowdStrike	File Info VERSION : 1.4.0.0
Cybereason	① Malicious.3dc1c9	Cylance	HOSTS : 10.200.240.187:4673; KEY (Base64) : Lx0qk/UfcERYeWu10J1Gw== AUTHKEY (Base64) : St:jSo+IFlynnrMpzSu2+kukcyPfdmDWtVFUaGdpKAhPftczHmIT3d7MXGwQwVeNhZbr50U6n49yewJUmLQyw==
Cyren	① W32/Trojan.TAOA-2201	DrWeb	SUBDIRECTORY : jre1.8.0_221 INSTALLNAME : jawn.exe
eGambit	① Unsafe_AI_Score_93%	Emsisoft	MUTEX : B0S0SOHzGn9AAECIV STARTUPKEY : Java(TM) Platform SE binary platform
Endgame	① Malicious (high Confidence)	eScan	ENCRYPTIONKEY : oh5jR3oUk78wpWhz8TdU TAG : NewAve
ESET-NOD32	① A Variant Of Win32/Packed EnigmaProt...	F-Secure	_LOGDIRECTORYNAME : TeamViewer

The screenshot shows a blog post on the G DATA website. The header includes the G DATA logo and navigation links for 'G DATA Campus', 'SMB Security', 'Tips and tricks', and 'Techblog'. Below the header is a menu bar with categories: 'Ransomware', 'Warning', 'Malware', 'CyberCrime', 'Exploits', 'Phishing', and 'Bots & Botnets'. The main title of the post is 'Strange Bits: Sodinokibi Spam, CinaRAT, and Fake G DATA'. A timestamp indicates it was published on '06/04/2019'. The background of the post features a stylized, abstract graphic with the word 'BITS' prominently displayed in red.

In the second part of our Strange Bits series we are taking a closer look at Sodinokibi Spam E-Mails, CinaRAT and a Malware that tries to imitate G DATA.

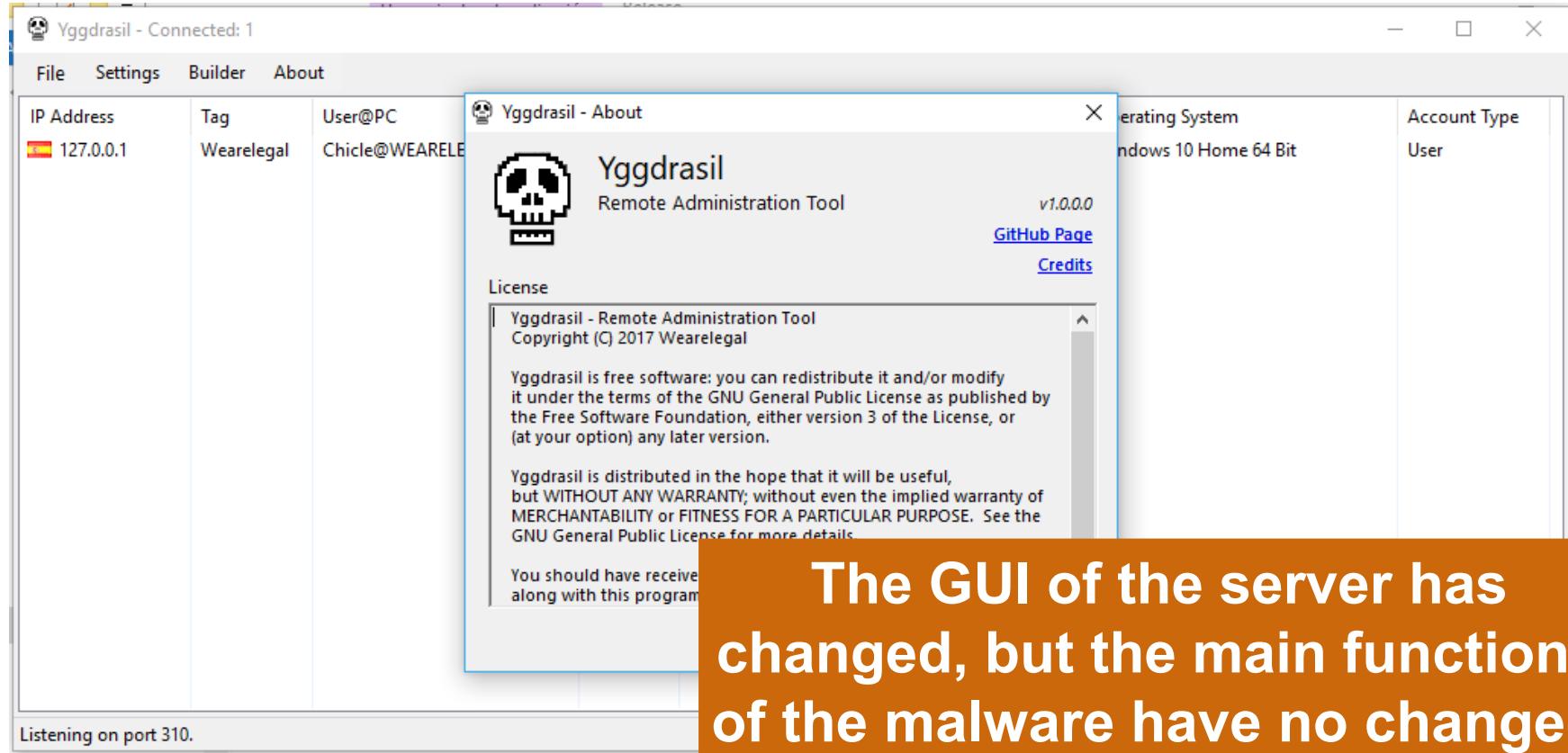
"That's strange..."

Many important discoveries do not start with a shouting of „Eureka“ anymore, as they did in the days of old. Instead, the most intriguing bits of modern research will at some point contain the phrase "That's strange..." followed by more prodding and poking and – hopefully – a



Karsten Hahn
Malware Analyst

CinaRAT



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

VenomRAT is a RAT sold for \$150 per month.

The screenshot shows the official website for Venom Software. At the top is a blue header bar with the logo 'V' on the left and navigation links: Home, About, Contact Us, and Join On Discord. Below the header is a large image of a dark blue software box labeled 'VENOM' with the tagline 'FAST AND LIGHT WEIGHT'. To the right of the box is a large 'V' logo and the text 'VENOM SOFTWARE'. A paragraph explains that the tool was developed for macro exploit users and highlights remote file management and registry/command access. A prominent blue button below the text says 'BUY 1 MONTH - 150\$'. At the bottom of the page is a dark footer section containing a grid of 24 checkmark icons, each representing a feature: IPV4 & IPV6 SUPPORT, PROTOCOL BUFFERS, ENCRYPTED COMMUNICATION, ROOTKIT, HIDDEN PROCESS, HIDDEN FILE, HIDDEN STARTUP, VELOS STEALER, MULTI-THREADED, UPNP SUPPORT, NO-IP.COM SUPPORT, EXPLOIT WORD MACRO, ANONFILE UPLOAD, PASTEBIN UPLOADER, OFFLINE & ONLINE KEYLOGGER, REMOTE WEBCAM, TASK MANAGER, REMOTE SHELL, REGISTRY EDITOR, UAC BYPASS, ELEVATE CLIENT PERMISSIONS, HIDDEN - HRDP, HIDDEN - VNC, BINDER, UNICODE KEYLOGGER, REVERSE PROXY, PASSWORD RECOVERY, SHOW MESSAGEBOX, VISIT WEBSITE (HIDDEN), COMPUTER COMMANDS, UPLOAD & EXECUTE, and DOWNLOAD & EXECUTE.

VenomRAT has some additional unique features

VenomRAT Commands

```
3  public static Type[] GetPacketTypes()
4  {
5      return new Type[]
6      {
7          typeof(GetAuthentication),
8          typeof(DoClientDisconnect),
9          typeof(DoClientReconnect),
10         typeof(DoClientUninstall),
11         typeof(DoWebcamStop),
12         typeof(DoAskElevate),
13         typeof(DoRemoveRdp),
14         typeof(DoRemoveVnc),
15         typeof(DoDownloadAndE),
16         typeof(DoInstallVNC),
17         typeof(DoInstallRDP),
18         typeof(DoStealer),
19         typeof(DoUploadAndExe),
20         typeof(DoGetDesktop),
21         typeof(DoGetProcesses),
22         typeof(DoProcessKill),
23         typeof(DoProcessStart),
24         typeof(DoGetDrives),
25         typeof(DoGetDirectory),
26         typeof(DoDownloadFile),
27         typeof(DoMouseMoveEvent),
28         typeof(DoKeyboardEvent),
29         typeof(DoGetSystemInfo),
30         typeof(DoGetVncInfo),
31         typeof(DoGetRdpInfo),
32         typeof(DoVisitWebsite),
33         typeof(DoShowMessageBox),
34         typeof(DoClientUpdate),
35         typeof(DoGetMonitors),
36         typeof(DoGetWebcams),
37         typeof(DoGetWebcam),
38         typeof(DoShellExecute),
39     }
40 }
```

Added Commands

```
typeof(DoRemoveRdp),
typeof(DoRemoveVnc),
typeof(DoInstallVNC),
typeof(DoInstallRDP),
typeof(DoStealer),
```

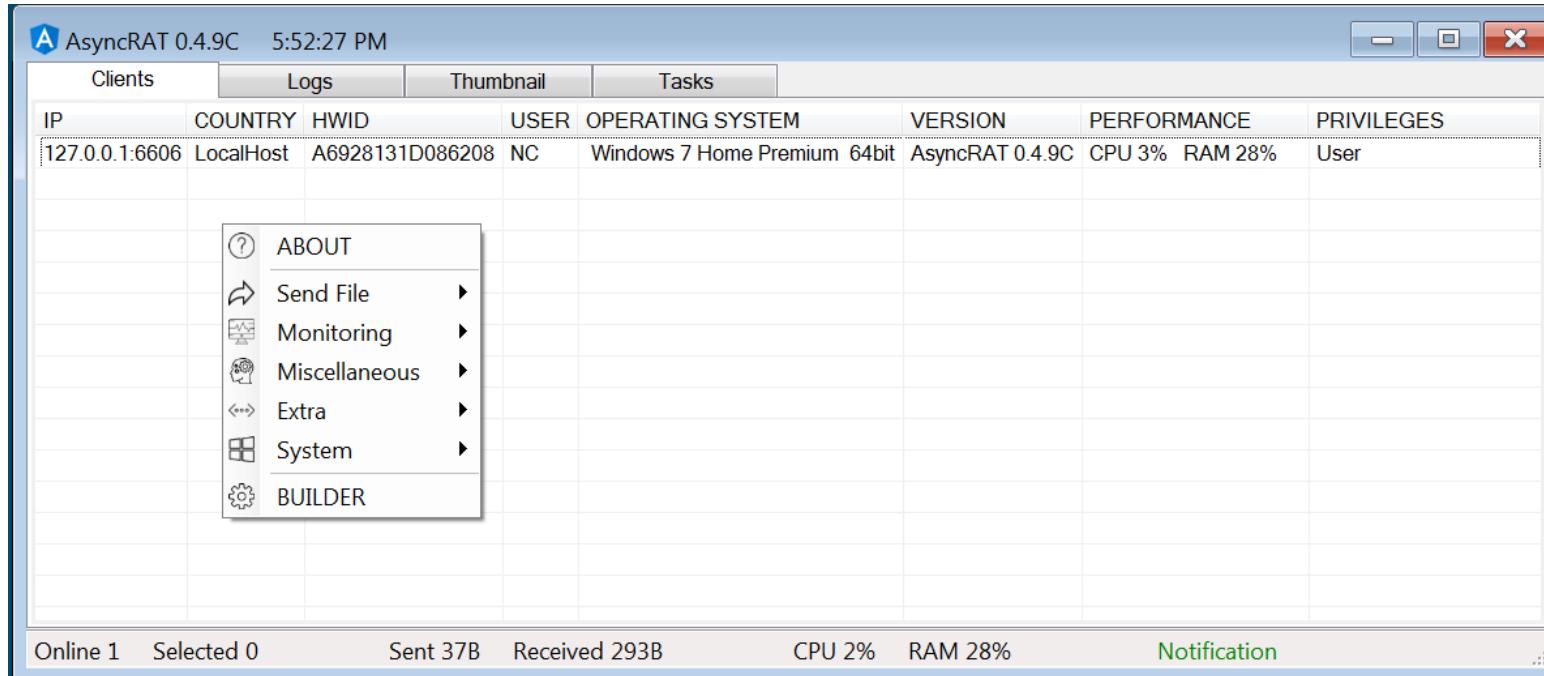
```
typeof(GetVncInfo),
typeof(GetRdpInfo),
typeof(GetAllPasswords),
```

QuasarRAT Commands

```
6  namespace xClient.Core.Packets
7  {
8      public class PacketRegistry
9      {
10         public static Type[] GetPacketTypes()
11         {
12             return new Type[]
13             {
14                 typeof(Packets.ServerPackets.GetAuthentication),
15                 typeof(Packets.ServerPackets.DoClientDisconnect),
16                 typeof(Packets.ClientPackets.IentReconnect),
17                 ientUninstall),
18                 bcamStop),
19                 kElevate),
20                 nloadAndExecute),
21                 loadAndExecute),
22                 esktop),
23                 rocesses),
24                 ocessKill),
25                 ocessStart),
26                 drives),
27                 irectory),
28                 nloadFile),
29                 useEvent),
30                 yboardEvent),
31                 systemInfo),
32                 sitWebsite),
33                 owMessageBox),
34                 ientUpdate),
35                 onitors),
36             }
37         }
38     }
39 }
40 }
41 }
```

AsyncRAT

AsyncRAT is a plugin-based RAT.



AsyncRAT

AsyncRAT is a plugin-based RAT.

ツイートする

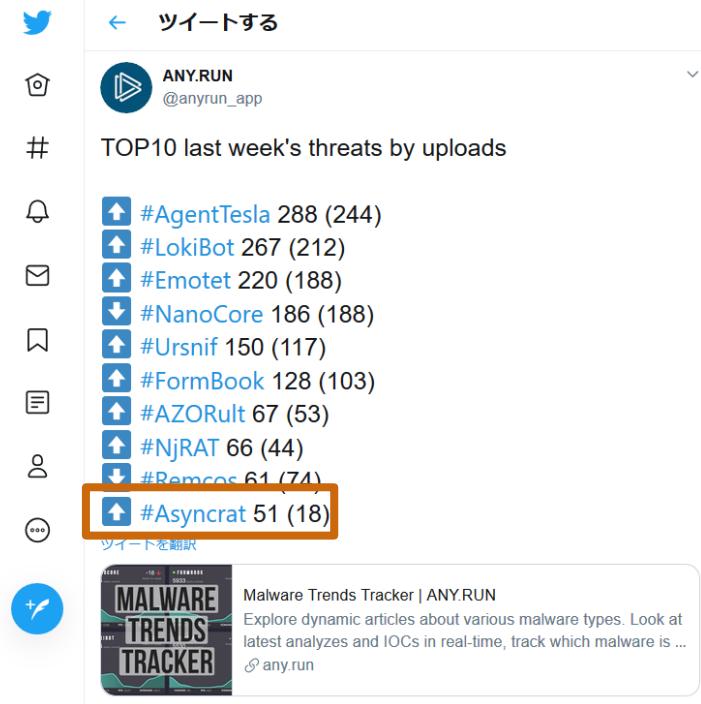
ANY.RUN
@anyrun_app

TOP10 last week's threats by uploads

#AgentTesla 288 (244)
#LokiBot 267 (212)
#Emotet 220 (188)
#NanoCore 186 (188)
#Ursnif 150 (117)
#FormBook 128 (103)
#AZORult 67 (53)
#NjRAT 66 (44)
#Remcos 61 (71)
#Asyncrat 51 (18)

ツイートを翻訳

MALWARE TRENDS TRACKER | ANY.RUN
Explore dynamic articles about various malware types. Look at latest analyzes and IOCs in real-time, track which malware is ...
any.run



AsyncRAT

AsyncRAT uses the same communication method as QuasarRAT.

AsyncRAT

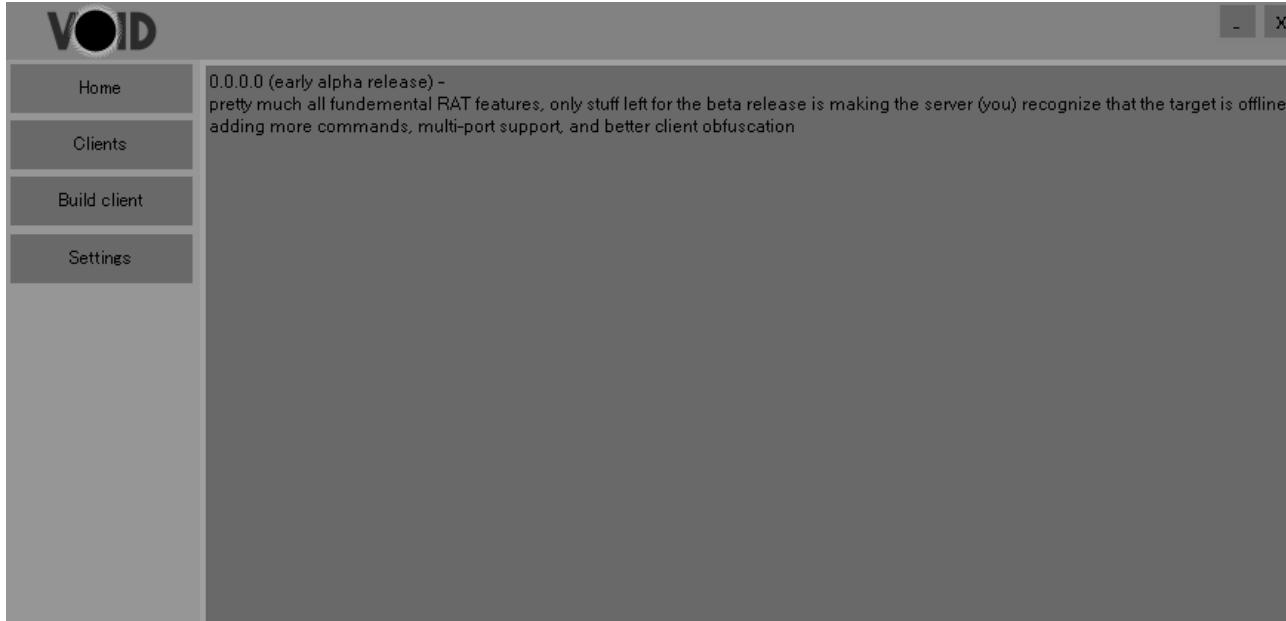
```
9  namespace Client.Algorithm
10 {
11     public class Aes256
12     {
13         private const int KeyLength = 32;
14         private const int AuthKeyLength = 64;
15         private const int IvLength = 16;
16         private const int HmacSha256Length = 32;
17         private readonly byte[] _key;
18         private readonly byte[] _authKey;
19
20         private static readonly byte[] Salt =
21         {
22             0xBF, 0xEB, 0x1E, 0x56, 0xFB, 0xCD, 0x97, 0x3B, 0xB2, 0x19, 0x2, 0x24, 0x30, 0xA5, 0x78, 0x43, 0x0, 0x3D, 0x56,
23             0x44, 0xD2, 0x1E, 0x62, 0xB9, 0xD4, 0xF1, 0x80, 0xE7, 0xE6, 0xC3, 0x39, 0x41
24         };
25     }
26 }
```

QuasarRAT

```
7  namespace xClient.Core.Cryptography
8  {
9      public static class AES
10     {
11         private const int IvLength = 16;
12         private const int HmacSha256Length = 32;
13         private static byte[] _defaultKey;
14         private static byte[] _defaultAuthKey;
15
16         public static readonly byte[] Salt =
17         {
18             0xBF, 0xEB, 0x1E, 0x56, 0xFB, 0xCD, 0x97, 0x3B, 0xB2, 0x19, 0x2, 0x24, 0x30, 0xA5, 0x78, 0x43, 0x0, 0x3D, 0x56,
19             0x44, 0xD2, 0x1E, 0x62, 0xB9, 0xD4, 0xF1, 0x80, 0xE7, 0xE6, 0xC3, 0x39, 0x41
20         };
21     }
22 }
```

Same
AES256 salt
value and
communication format.

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



Void-RAT

The image shows a screenshot of a Twitter search results page for the hashtag #VoidRAT. The search bar at the top contains the text "#VoidRAT". Below the search bar, there are five navigation links: 話題のツイート (Latest), 最新 (Latest), アカウント (Accounts), 画像 (Images), and 動画 (Videos). The "最新" link is currently selected. A button labeled "このスレッドを表示" (Show thread) is visible. The results list four tweets from the account ScumBots, all posted on April 13th. Each tweet provides a pastebin link and SHA256 hash for a VoidRAT sample, along with its C2 information.

ScumBots (@ScumBots · 4月13日)	#VoidRAT found at pastebin.com/raw/WwA67bVf SHA256: 70801b9f05d61d2c585711f9ea3c59a980a8d9590f84fe7aa76d4bc84046d4c4 C2: 127[.]0[.]0[.]1:4728
ScumBots (@ScumBots · 4月13日)	#VoidRAT found at pastebin.com/raw/bVBQwPYb SHA256: 19e5e863763a7d8a8ff4875b9c433c86acc70bbdff16c09ea36d136938b4c2cd C2: impawn[.]ddns[.]net:80
ScumBots (@ScumBots · 4月13日)	#VoidRAT found at pastebin.com/raw/83sJcPHk SHA256: 01bcbba7c8d15fc97a8c3923dc430f822bb3955dd17611b65008a081679f6910 C2: micalter-62870[.]portmap[.]host:62870
ScumBots (@ScumBots · 4月13日)	#VoidRAT found at pastebin.com/raw/hX8g2kmX SHA256: e9c623f9abfb6529763899c99d7a93911c645d803e9756a01295a4a6577c27df C2: holydns[.]warzoneds[.]com:7974

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

Void-RAT

```
int HmacSha256Length = 32;  
private static 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,  
    0};
```

QuasarRAT decompile

```
// Token: 0x040001BD RID: 445  
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,  
    0};  
221  
222  
223
```

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

```
3 RID: 2025 RVA: 0x000198A8 File Offset: 0x00017B48
d smethod_1()
{
    [64BIT]
    {
        Environment.SpecialFolder.SystemX86;
        Settings.specialFolder_0 = Environment.SpecialFolder.System;
        break;
        break;
        case Environment.SpecialFolder.ProgramFilesX86;
        Settings.specialFolder_0 = Environment.SpecialFolder.ProgramFiles;
        break;
        break;
        Settings.string_4 = Environment.GetFolderPath(Settings.specialFolder_0);
    }
}

// Token: 0xd4000416 RID: 1048
public static string string_0 = "f1\WnUsiuny4Erd4vzQ2Cuc3rRWhZbB0lsOvc0kx2j0kH8uk9RE40fVeta1lC97e0JmSPeYn1qVt5avg==";
// Token: 0xd4000417 RID: 1047
public static string string_1 = "Pac+ZDeph0CRE9TRgbTAK+TRw0lEys4HfxaxhJsczrUvyxgrayR3s4Pzv/NkE3YJVsxnX4cITxq1KeaJd0RN5t/Nk/sxOMcb4=";
// Token: 0xd4000418 RID: 1048
public static int int_0 = 3000;
public static string string_2 = "%42a0cd87hd7Y1x2vhLHw==";
// Token: 0xd400041A RID: 1050
public static string string_3 = "R4SD01Grsqj{0}F0lpFzY9z2Bn8y85e5#ajtkk8xb7P#5GA6ukktld4Sp1SBffyKNQzVX90rH6R1vLBu7a7==";
// Token: 0xd400041B RID: 1051
public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;
// Token: 0xd400041C RID: 1052
public static string string_4 = Environment.GetFolderPath(Settings.specialFolder_0);
// Token: 0xd400041D RID: 1053
public static string string_5 = "2#P7Pv82jVhsIvv+kxD1FeV28nRen/fkAb44/2TB4rc7Ducc8uJH/2Sv0p30c11tsSaak3dnIf/ud5hQ==";
// Token: 0xd400041E RID: 1054
public static string string_6 = "#02+jRUg2y8bETelLn6xbshGtaZAH0P4ZT71lw0l0xoks/aCh1en0075sDjaMox0Co75ByExIRawxtJntg==";
// Token: 0xd400041F RID: 1055
public static bool bool_0 = false;
// Token: 0xd4000420 RID: 1056
public static bool bool_1 = true;
// Token: 0xd4000421 RID: 1057
public static string string_7 = "/NGZNWa0RGxvUwv3sZkLMF53CJ1/F02xcibekIByRnf06100ID580iurvf#04LaNvW+f1J48nVRkzJlywz/ec3JvvKb8nfzGw==";
// Token: 0xd4000422 RID: 1058
public static string string_8 = "Nuuhz958R9Ky0xI4hvxtDFdvVNx1GbTy52oAyrrFZlVzAMJM3AgzBe0tZWzeouucht2m5S01yksN6jove9thq0M8dfl81V48PuZag==";
// Token: 0xd4000423 RID: 1059
public static bool bool_2 = false;
// Token: 0xd4000424 RID: 1060
public static bool bool_3 = true;
// Token: 0xd4000425 RID: 1061
public static string string_9 = "Stdtxce855BN0j2D1AQ";
// Token: 0xd4000426 RID: 1062
public static string string_10 = "E250fWj1go006iTfa0kaujezlz8E88E1rXkJzagH5ibgPmNoEEFelogGMfrswuNsRJDVSHNq120n/vs==";
// Token: 0xd4000427 RID: 1063
public static string string_11 = "0xzFwuhg178mW0EY4QhskWpzq2YDopg80/aJ7oRlb1881rGjdHDKvsnUr40gT3T2hX0fMFrrLc5tP2gz==";
// Token: 0xd4000428 RID: 1064
public static bool bool_4 = false;
// Token: 0xd4000429 RID: 1065
public static bool bool_5 = false;
```

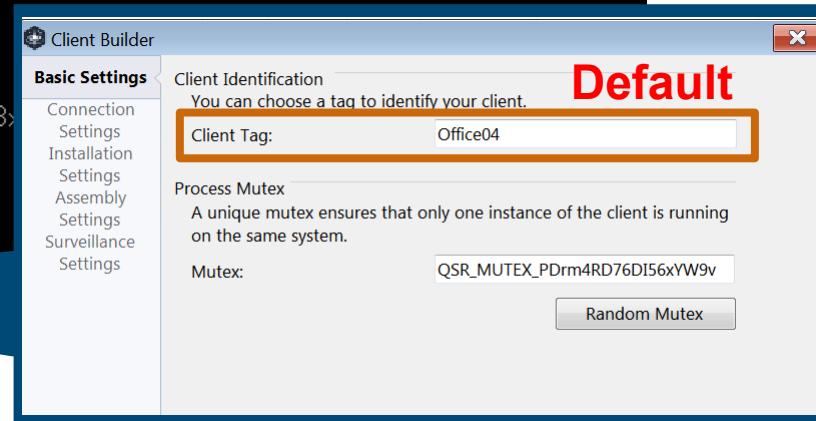
APT33 QuasarRAT Configuration

```
[+] 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)    : k4Ea0cde7hd7+Ytx2xhLHw==  
AUTHKEY (Base64) : R4WSD1Gysgj i0cjFGVpFz3Y3gZ6n9yS5m5HqjtkK8;  
SUBDIRECTORY    : SubDir  
INSTALLNAME     : Client.exe  
MUTEX            : QSR_MUTEX_RjrwbI1Icfgf7P0FqQF  
STARTUPKEY      : Windows Session Manager  
ENCRYPTIONKEY   : SthdV~w99EBNQigDTAQD  
TAG              : Office04  
LOGDIRECTORYNAME: Logs
```



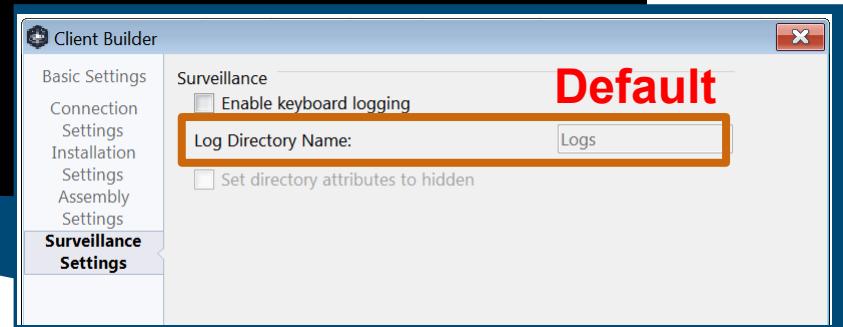
APT33 QuasarRAT Configuration

```
[+] 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) : R4WSD1Gysgj i0jFGVpFz3Y3gZ6n9yS5m5HqjtkK8xb7PW5GAGUkkt l d4YSptS9ffyKNQxWXQOrH6RIwL6Um7A==  
SUBDIRECTORY     : SubDir  
INSTALLNAME     : Client.exe  
MUTEX            : QSR_MUTEX_RjrwbI1Icfgf7P0FqQF  
STARTUPKEY       : Windows Session Manager  
ENCRYPTIONKEY   : StbdXcw885BNQigDTAQD  
TAG               : Office04  
LOGDIRECTORYNAME : Logs
```



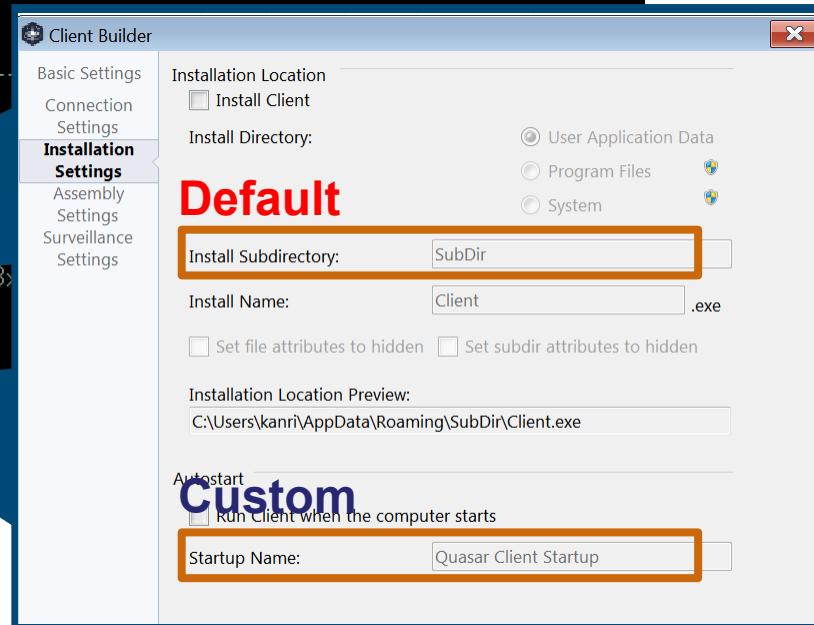
APT33 QuasarRAT Configuration

```
[+] 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+Ytx2xbU  
MUTHEK (Base64) : RMMSB1GvzJ1Q01F777L2v7g2onaysoompHqjtkK8  
SUBDIRECTORY : SubDir  
INSTALLNAME : Client.exe  
MUTEX : OSRMUTEX_P1mbhT1IogfZP0FqQF  
STARTUPKEY : Windows Session Manager  
ENCRYPTIONKEY : stbaACW88jDBNQ1gDUtAQD  
TAG : Office04  
LOGDIRECTORYNAME : Logs
```

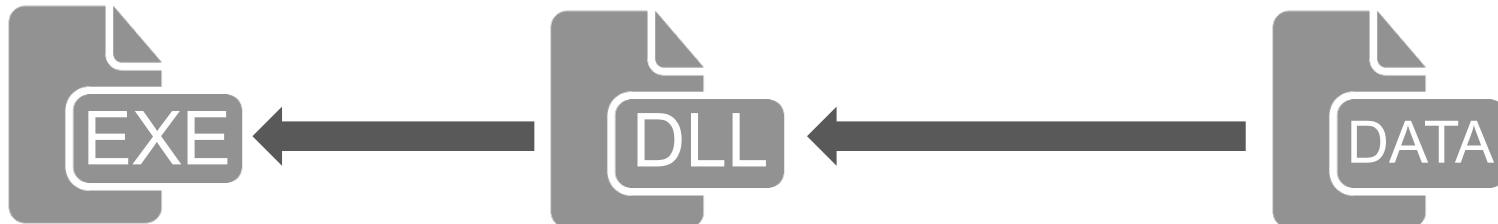


QuasarRAT Configuration Pattern

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

"Custom" Case of APT10

Service



svchost.exe

UsdwuerDataAccessRes.dll

sbscmp20_mscorwks.dll.rsp

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
; .data:of=0000000000000000
public FuckYouAnti
proc near
; DATA XREF
; .rdata:of=0000000000000000
; .text:000000018000FFA0
; .text:000000018000FB0
; .text:000000018000FFC0
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:000000018000FB0 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

Config

Encryption

Commands

Error Log

Communication

APT10 QuasarRAT Configuration

APT10 QuasarRAT

```
49 // Token: 0x04000056 RID: 86
50 public static string VERSION = "0x6cN0+IHvn9Im0xEc40eyrUS0RpH10J3P0/r<SE0WjyInSrDvB1db91vSz7JaMfKXu1cfQlx08YTp/Zfsg==";
51 // Token: 0x04000057 RID: 87
52 public static string HOSTS = "TmP21/zkScJvK3aijv/9JB70fJbt5ZsY91Devkzs2bnrJX3XvNTDj/sej1YJFgXTo1h80T1mcEnvQI0jNwNRExqX0E4k1";
53 // Token: 0x04000058 RID: 88
54 public static int RECONNECTDELAY = 34109;
55 // Token: 0x04000059 RID: 89
56 public static string KEY = "+PrSiHNK8Mq4nK+spXzw==";
57 // Token: 0x0400005A RID: 90
58 public static string AUTHKEY = "Gr7jcnpx4j1oMfrLnbfFe2tmzjMEiu0u0201SsJ24cE3b0d7/zV0sPV1eNsELT00Mh50+00ckZ2cn0g==";
59 // Token: 0x0400005B RID: 91
60 public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;
61 // Token: 0x0400005C RID: 92
62 public static string DIRECTORY = Environment.GetFolderPath(init_config.specialFolder_0);
63 // Token: 0x0400005D RID: 93
64 public static string SUBDIRECTORY = "(00+Xc7TV1b01c)B0+75wn0Gax!NokNmRLAfLs1!0RyKoXUMs1Ar0tlZvWkvJ5pUdpCjaeo0sDn1pmw==";
65 // Token: 0x0400005E RID: 94
66 public static string INSTALLNAME = "/tpW0tt1TFTAlm/v1naefP0s23Lcv3hnr0VrxxjYctKsmgTdHJNTsSmwvylK9h5943fEz2Auzk4rBLP7m4hw==";
67 // Token: 0x0400005F RID: 95
68 public static bool INSTALL = false;
69 // Token: 0x04000060 RID: 96
70 public static bool STARTUP = false;
71 // Token: 0x04000061 RID: 97
72 public static string MUXEX = "yaXh4f4uVnHvZ0M1s5A2J9JpTkda4ufU01S017swc0Vyyi0J4egU0LTsJ1k3B0X1Hpk81xJ9sfIZF0X0Lmf0r0Bz0F2";
73 // Token: 0x04000062 RID: 98
74 public static string STARTUPKEY = "17CimfdhnuGWhzkT5Dyzl3qMdhjNS0+x534Ef37R86xxZ00KaHvMb3Yh5c02rte2Bu40UJ3QR1WCrJvEzZIMfJ8r";
75 // Token: 0x04000063 RID: 99
76 public static bool HIDEFILE = false;
77 // Token: 0x04000064 RID: 100
78 public static bool ENABLELOGGER = false;
79 // Token: 0x04000065 RID: 101
80 public static string ENCRYPTIONKEY = "mAxAFFoc04Mw0HJM3ia";
81 // Token: 0x04000066 RID: 102
82 public static string TAG = "MOP7874N14bnTawBeUxHvn2ar915mdly[0zcv4G0Hfc9L4rxYduEhmEc820rjsTc0/Sn6nn12MSdH8Soa==";
83 // Token: 0x04000067 RID: 103
84 public static string LOGIDIRECTORYNAME = "Easuram0btYMHMh3cPHip/s871B7ge25x28J+20cbxb2kzAE89aGBSYINraFuv1cpu8t0wwfY52uyg==";
85 // Token: 0x04000068 RID: 104
86 public static bool HIDELOGIDIRECTORY = false;
87 // Token: 0x04000069 RID: 105
88 public static bool HIDEINSTALLSUBDIRECTORY = false;
89 // Token: 0x0400006A RID: 106
90 public static string Download_url = "080f98deG4E/07n/DYsKkzxrxG/2hbs8GUjp63oe07ZUNnFjPe7u2/bdreqdR0w+jnwA2s/lmw60zuMs==";
91 // Token: 0x0400006B RID: 107
92 public static string Proxy = "tIN-fstXhrpBr298hH8t10xrCSB1LAYapEysCPr/HnA2-Kd0/2z2000.lnfcoYmew7UksPv02L7Kva==";
```

QuasarRAT

```
49 // Token: 0x04000008 RID: 8
50 public static string VERSION = "7UvADP//8YS4Pi/T1R6R049f9HntDsm#FchNq8ca0vJrsNb0ozr0kd2e39e5nVYa1k9B5+dYQGMw==";
51 // Token: 0x04000009 RID: 9
52 public static string HOSTS = "Tau77dmUjk/Zlsz66TL5TUxEx/L0h1r5uxat1THc4hvRJd5xp4:035SrB1DLtdkd+tsDe2ewmNbSJW1us8HEoms8v9dbAf128Dlw40sBk5X";
53 // Token: 0x0400000A RID: 10
54 public static int RECONNECTDELAY = 3000;
55 // Token: 0x0400000B RID: 11
56 public static string KEY = "ebd0s0089sJEWp1ItdABQ==";
57 // Token: 0x0400000C RID: 12
58 public static string AUTHKEY = "30c2h3nanEdTE5yMFYK1PR90H51bZLcqE0ccu1HMgtDR80xD7Fv9X0KcDjAU/muBB4/KAExalUM8KJEJZB0==";
59 // Token: 0x0400000D RID: 13
60 public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;
61 // Token: 0x0400000E RID: 14
62 public static string DIRECTORY = Environment.GetFolderPath(init_config.specialFolder_0);
63 // Token: 0x0400000F RID: 15
64 public static string SUBDIRECTORY = "d7ALk0A1S1Pt5e4bbnsEuESGLsawIE9BMHkzTeR93Jnru#6drfsS9Z1Xlyrt318xpkw8cvpu0A9aFOGCVa==";
65 // Token: 0x04000010 RID: 16
66 public static string INSTALLNAME = "/npeA2FTtNn1tk0428hV7McokYcrwJUNR0y1AtKc00eq9ib1vFMLetVcDE0)albNv/Zs2B8/JSBQj==";
67 // Token: 0x04000011 RID: 17
68 public static bool INSTALL = false;
69 // Token: 0x04000012 RID: 18
70 public static bool STARTUP = false;
71 // Token: 0x04000013 RID: 19
72 public static string MUXEX = "X1T4grV+qtsrifJxUJpt0MeVzclj57988a0t08g1Rt0gfDl/r1y05F1237.JEEUbJNTB1CxtkMavYFp-s0LOMvUTWE/fabpxtXFSQrwA==";
73 // Token: 0x04000014 RID: 20
74 public static string STARTUPKEY = "1p3jtNTns9mo1MzfUvddz0Y0Mv0izpAV8RFtK93RtWGNPAT0ChK0JxJ+0Tr7GEJhNuK3zhW0V/ZkNa0sLL7er90H+Ht0ybu0-";
75 // Token: 0x04000015 RID: 21
76 public static bool HIDEFILE = false;
77 // Token: 0x04000016 RID: 22
78 public static bool ENABLELOGGER = true;
79 // Token: 0x04000017 RID: 23
80 public static string ENCRYPTIONKEY = "2XmEP5ycdalwbl0z781";
81 // Token: 0x04000018 RID: 24
82 public static string TAG = "K1A1rxysot1MbhNyNj1H+pry2N4dMnt1M11C08PBFbTPSM0az18ysA16V088eivz1ZtMfnD1NM4ZQgus==";
83 // Token: 0x04000019 RID: 25
84 public static string LOGIDIRECTORYNAME = "aX4+0Tw0952T790C1FmX/vFGG5SSu4YjIRzF60Jdt3LJ2US0X6J3p+fsq9EeN8YnhhhoY84DKNFRgw==";
85 // Token: 0x0400001A RID: 26
86 public static bool HIDELOGIDIRECTORY = true;
87 // Token: 0x0400001B RID: 27
88 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 
SETUP	PROXY 
MUTEX	

APT10 QuasarRAT Configuration

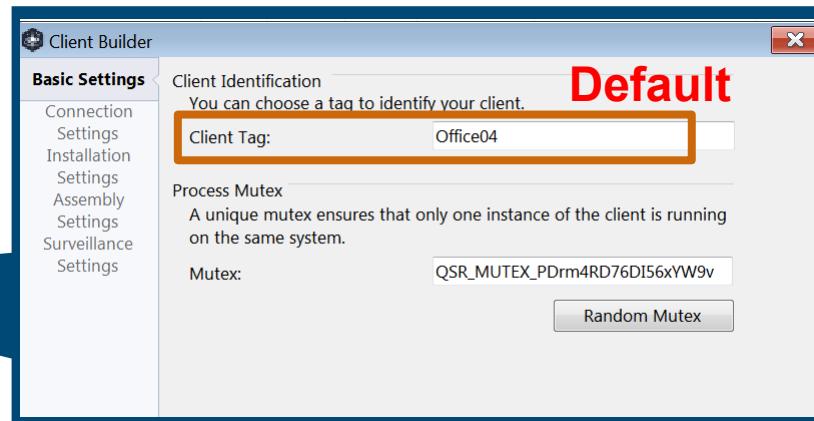
Config

```
VERSION          : 2.0.0.0
HOSTS            : 178.32.24.180:443;
RECONNECTDELAY   : 34109
KEY (Base64)     : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64) : Gr7icnpxp4j1oWFhmLnbfFe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY     : SubDir
INSTALLNAME      : Client.exe
INSTALL          : FALSE
STARTUP          : FALSE
MUTEX             : cOKPyGqEqejOVh0SaOwH1xa5
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

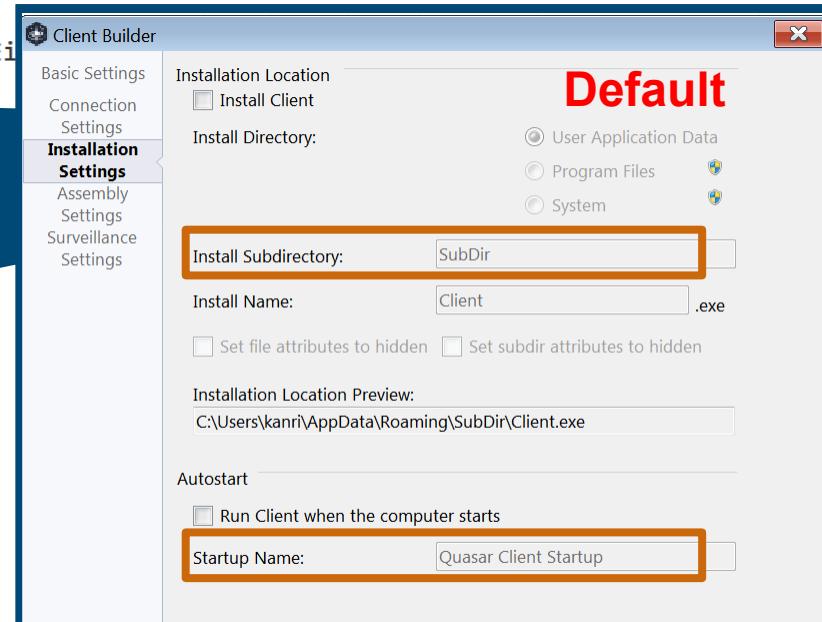
```
VERSION          : 2.0.0.0
HOSTS            : 178.32.24.180:443;
RECONNECTDELAY   : 34109
KEY (Base64)     : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64) : Gr7icnpxp4j1oWFhmLnbfFe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY     : SubDir
INSTALLNAME      : Client.exe
INSTALL          : FALSE
STARTUP          : FALSE
MUTEX             : cOKPyGqeojVh0Sa0wH1xa5
STARTUPKEY       : Quasar Client Startup
HIDEFILE         : FALSE
ENABLELOGGER     : FALSE
ENCRYPTIONKEY    : mAAXGFFacOAMw8HMJ3Iq
TAG               : Office04
LOGDIRECTORYNAME : Logs
HIDEDIRECTORY    : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url     : none
Proxy             :
```



APT10 QuasarRAT Configuration

Config

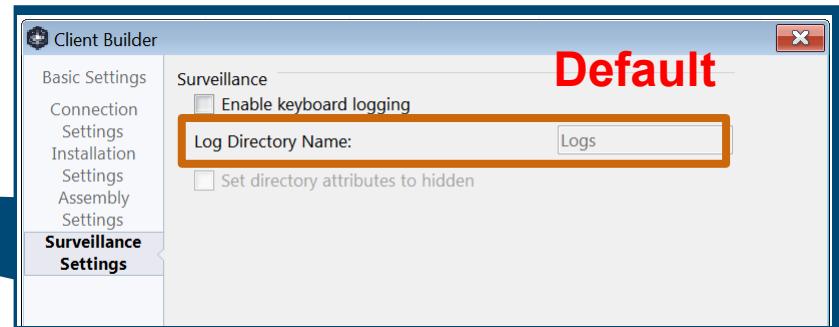
```
VERSION          : 2.0.0.0
HOSTS            : 178.32.24.180:443;
RECONNECTDELAY   : 34109
KEY (Base64)     : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64) : Gr7icnpxp4j1oWFhmLnbfFe2fnmzIMEi
SUBDIRECTORY     : SubDir
INSTALLNAME      : Client.exe
INSTALL          : FALSE
STARTUP          : FALSE
Mutex             : cOKPyGqejejOVh0SaOwH1xa5
STARTUPKEY       : Quasar Client Startup
HIDEFILE         : FALSE
ENABLELOGGER     : FALSE
ENCRYPTIONKEY    : mAxAxGFFqcOAMw8HMJ3Iq
TAG               : Office04
LOGDIRECTORYNAME : Logs
HIDEDIRECTORY    : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url     : none
Proxy              :
```



APT10 QuasarRAT Configuration

Config

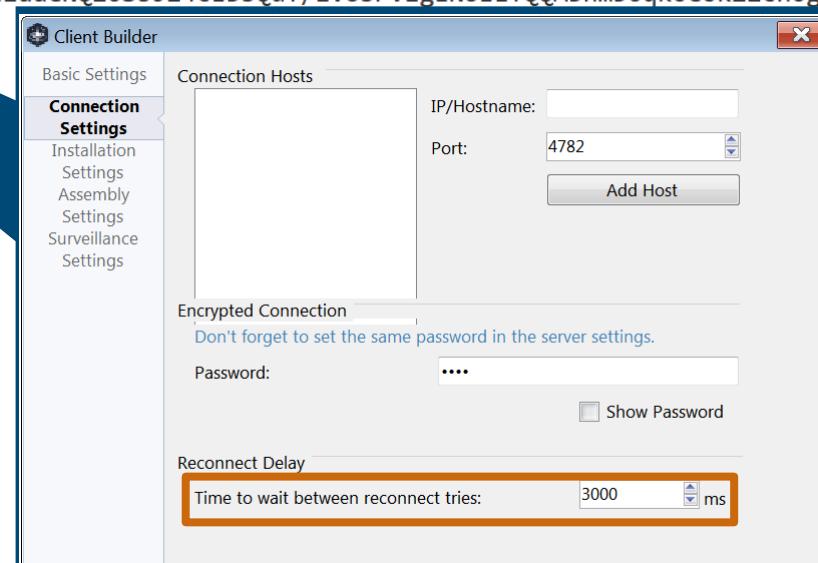
```
VERSION          : 2.0.0.0
HOSTS            : 178.32.24.180:443;
RECONNECTDELAY   : 34109
KEY (Base64)     : +PrSihNNK8Mpq4nK+spXzw==
AUTHKEY (Base64) : Gr7icnpxp4j1oWFhmLnbfFe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY     : SubDir
INSTALLNAME      : Client.exe
INSTALL          : FALSE
STARTUP          : FALSE
MUTEX             : cOKPyGqeojVh0SaOwH1xa5
STARTUPKEY       : Quasar Client Startup
HIDEFILE         : FALSE
ENABLELOGGER     : FALSE
ENCRYPTIONKEY    : mAxAxGFFqcOAMw8HMJ3Iq
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)     : +PrSihNNPBMpq4nK+spXzw==
AUTHKEY (Base64)  : Gr7icnpxp4J1...hmLnbfIfe2fnmzIMEiuuCNQ20SsJ24cE3bQdT/zVObPVlgENoELTQQM3hm50qk0CokZ2cn0g==
SUBDIRECTORY     : SubDir
INSTALLNAME      : Client.exe
INSTALL          : FALSE
STARTUP          : FALSE
MUTEX             : cOKPyGqeojOVh0Sa0wH1xa5
STARTUPKEY        : Quasar Client Startup
HIDEFILE          : FALSE
ENABLELOGGER      : FALSE
ENCRYPTIONKEY    : mAxAxGFFqcOAMw8HMJ3Iq
TAG               : Office04
LOGDIRECTORYNAME : Logs
HIDEDIRECTORY    : FALSE
HIDEINSTALLSUBDIRECTORY : FALSE
Download_url      : none
Proxy              :
```



Configuration Encryption

APT10 QuasarRAT

```
// Token: 0x000000BE RID: 190
public static byte[] decode_data(byte[] input)
{
    if (decode_main.byte_0 == null || decode_main.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())
            {
                using (AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider())
                {
                    aesCryptoServiceProvider.KeySize = 128;
                    aesCryptoServiceProvider.BlockSize = 128;
                    aesCryptoServiceProvider.Mode = CipherMode.CFB;
                    aesCryptoServiceProvider.Padding = PaddingMode.PKCS7;
                    aesCryptoServiceProvider.Key = decode_main.byte_0;
                    using (HMACSHA256 hmacsha = new HMACSHA256(decode_main.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 (!key_create.method_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(),
                        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.");
}
```

CFB Mode

QuasarRAT

```
public static byte[] decode_data(byte[] input)
{
    if (decode_main.byte_0 == null || decode_main.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())
            {
                using (AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider())
                {
                    aesCryptoServiceProvider.KeySize = 128;
                    aesCryptoServiceProvider.BlockSize = 128;
                    aesCryptoServiceProvider.Mode = CipherMode.CBC;
                    aesCryptoServiceProvider.Padding = PaddingMode.PKCS7;
                    aesCryptoServiceProvider.Key = decode_main.byte_0;
                    using (HMACSHA256 hmacsha = new HMACSHA256(decode_main.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 (!key_create.method_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(),
                        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.");
}
```

CBC Mode

Commands

APT10 QuasarRAT

```
48      typeof(object),
49      typeof(DoLoadRegistryKey),
50      typeof(DoCreateRegistryKey),
51      typeof(DoDeleteRegistryKey),
52      typeof(DoRenameRegistryKey),
53      typeof(DoCreateRegistryValue),
54      typeof(DoDeleteRegistryValue),
55      typeof(DoRenameRegistryValue),
56      typeof(DoChangeRegistryValue),
57      typeof(SetAuthenticationSuccess),
58      typeof(GetConnections),
59      typeof(DoCloseConnection),
60      typeof(GetAuthenticationResponse),
61      typeof(SetStatus),
62      typeof(SetStatusFileManager),
63      typeof(object),
64      typeof(DoUserStatus),
65      typeof(GetProcessesResponse),
66      typeof(GetDrivesResponse),
67      typeof(GetDirectoryResponse),
68      typeof(DoDownloadFile),
69      typeof(GetSystemInfoResponse),
70      typeof(GetMonitorsResponse),
71      typeof(object),
72      typeof(object),
73      typeof(object),
74      typeof(object),
75      typeof(object),
76      typeof(object),
77      typeof(GetRegistryKeysResponse),
78      typeof(GetCreateRegistryKeyResponse),
79      typeof(GetDeleteRegistryKeyResponse),
80      typeof(GetRenameRegistryKeyResponse),
81      typeof(GetCreateRegistryValueResponse),
82      typeof(GetDeleteRegistryValueResponse),
83      typeof(GetRenameRegistryValueResponse),
84      typeof(GetChangeRegistryValueResponse),
85      typeof(ReverseProxyConnect),
86      typeof(ReverseProxyConnectResponse),
87      typeof(ReverseProxyData),
88      typeof(ReverseProxyDisconnect),
89      typeof(GetConnectionsResponse),
90      typeof(DoPluginResponse),
91      typeof(DoPlugin)
92      ];
93  ];
```

Disable

QuasarRAT

```
50      typeof(GetPasswords),
51      typeof(DoLoadRegistryKey),
52      typeof(DoCreateRegistryKey),
53      typeof(DoDeleteRegistryKey),
54      typeof(DoRenameRegistryKey),
55      typeof(DoCreateRegistryValue),
56      typeof(DoDeleteRegistryValue),
57      typeof(DoRenameRegistryValue),
58      typeof(DoChangeRegistryValue),
59      typeof(SetAuthenticationSuccess),
60      typeof(GetConnections),
61      typeof(DoCloseConnection),
62      typeof(GetAuthenticationResponse),
63      typeof(SetStatus),
64      typeof(SetStatusFileManager),
65      typeof(SetUserStatus),
66      typeof(GetDesktopResponse),
67      typeof(GetProcessesResponse),
68      typeof(GetDrivesResponse),
69      typeof(GetDirectoryResponse),
70      typeof(DoDownloadFileResponse),
71      typeof(GetSystemInfoResponse),
72      typeof(GetMonitorsResponse),
73      typeof(GetWebcamsResponse),
74      typeof(DoShellExecuteResponse),
75      typeof(GetStartupItemsResponse),
76      typeof(GetKeyloggerLogResponse),
77      typeof(GetPasswordsResponse),
78      typeof(GetRegistryKeysResponse),
79      typeof(DoCreateRegistryKeyResponse),
80      typeof(DoDeleteRegistryKeyResponse),
81      typeof(DoRenameRegistryKeyResponse),
82      typeof(DoCreateRegistryValueResponse),
83      typeof(DoDeleteRegistryValueResponse),
84      typeof(DoRenameRegistryValueResponse),
85      typeof(DoChangeRegistryValueResponse),
86      typeof(ReverseProxyConnect),
87      typeof(ReverseProxyConnectResponse),
88      typeof(ReverseProxyData),
89      typeof(ReverseProxyDisconnect),
90      typeof(GetConnectionsResponse),
91      typeof(DoPluginResponse),
92      typeof(DoPlugin)
93  ];
```

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 }
```

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



```
35 internal static byte[] request_encode(byte[] payload)
36 {
37     payload = SafeQuickLZ.Compress(payload, 3);
38     payload = decode_main.Aes_decode(payload);
39     byte[] array = new byte[payload.Length + class_decode.HEADER_SIZE];
40     for (int i = 0; i < payload.Length; i++)
41     {
42         payload[i] ^= class_decode.Enc_Key[i % 4];
43     }
44     byte[] bytes = BitConverter.GetBytes(payload.Length);
45     for (int j = 0; j < bytes.Length; j++)
46     {
47         bytes[j] ^= class_decode.Enc_Key[j];
48     }
49     Array.Copy(bytes, array, class_decode.HEADER_SIZE);
50     Array.Copy(payload, 0, array, class_decode.HEADER_SIZE, payload.Length);
51     return array;
52 }
```

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.

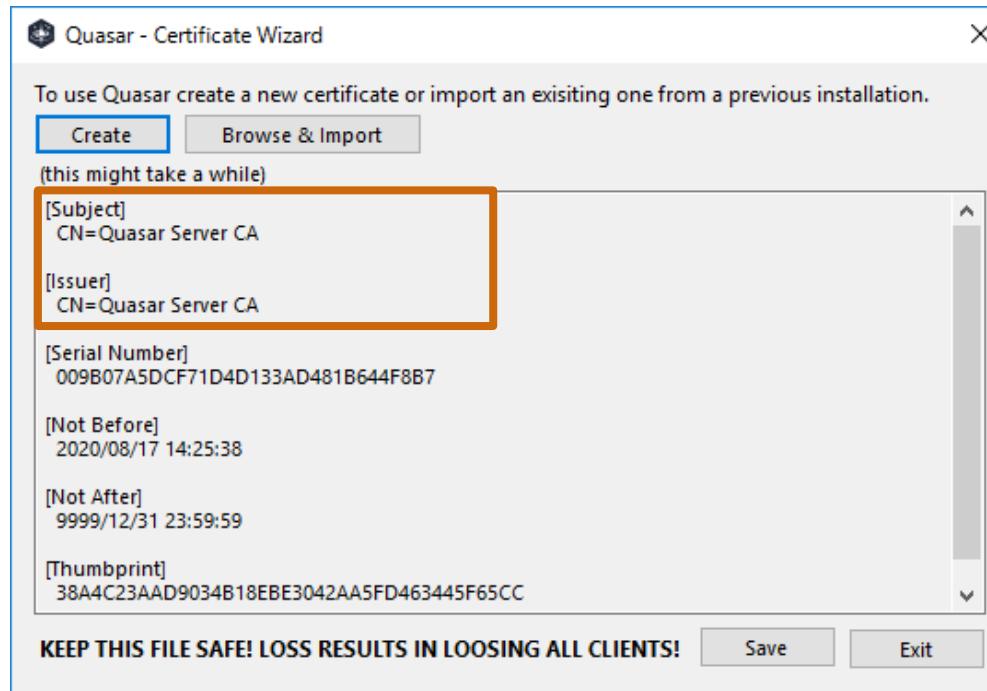
IoT Search Engine

- 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.

The certificate issuer contains the "Quasar Server CA".



Search IoT Engine

QuasarRAT 1.4

SHODAN Downloads Pricing

Censys

TOTAL RESULTS

4

TOP COUNTRIES



Romania

Netherlands

Japan

China

TOP PORTS

443

9443

9001

Quick Filters

For all fields, see [Data Definitions](#)

Autonomous System:

1 HETZNER-AS

1 HWSP-AS-AP

HostPalace Web
Solution PVT LTD

1 NICEIT

1 Online SAS

1 RCS-RDS 73-75 Dr.
Staicovici

Protocol:

5 443/https

2 3389/rdp

2 445/smb

2 80/http

1 22/ssh

More

TOP ORGANIZATIONS

Tencent cloud computing

HostPalace Web Solution F

Digi Romania

Anchnet Asia Limited

Tag:

5 https

2 http

2 rdp

2 remote_display

2 smb

More

IPv4 Hosts

Page: 1/1 Results: 5 Time: 89ms

195.154.44.86 (195-154-44-86.rev.poneytelecom.eu)

Online SAS (12876) Entremont-le-Vieux, Auvergne-Rhone-Alpes, France
Ubuntu 3389/rdp, 443/https, 80/http
Welcome to Your_domain! Quasar Server CA
443.https.tls.certificate.parsed.subject.common_name: Quasar Server CA

188.24.88.1 (188-24-88-1.rdsnet.ro)

RCS-RDS 73-75 Dr. Staicovici (8708) Cluj-Napoca, Cluj, Romania
443/https
Quasar Server CA
443.https.tls.certificate.parsed.subject.common_name: Quasar Server CA

88.99.89.152 (static.152.89.99.88.clients.your-server.de)

HETZNER-AS (24940) Germany
443/https, 80/http, 8080/http
Quasar Server CA
443.https.tls.certificate.parsed.subject.common_name: Quasar Server CA

103.194.171.84 (hosted-by.hostspicy.com)

HWSP-AS-AP HostPalace Web Solution PVT LTD (134512)
22/ssh, 443/https, 445/smb
Quasar Server CA
443.https.tls.certificate.parsed.subject.common_name: Quasar Server CA

45.9.148.82

NICEIT (49447) Amsterdam, North Holland, Netherlands
443/https, 80/http, 445/ssh
TLSv1



cert="Quasar Server CA"

类型分布

协议 105

年份

2020 105

国家/地区排名

中国 20

美国 16

德国 9

中国香港特别行政区 8

法国 5

105 条匹配结果 (101 条独立IP), 46 ms, 关键词搜索。

显示一年内数据, 点击 all 查看所有。

45.64.53.235

45.64.53.235

中国香港特别行政区

ASN: 38197

组织: Sun Network (Hong Kong) Limited - HongKong Backbone

2020-11-08



+ Certificate



120.79.185.187

120.79.185.187

中国

ASN: 37963

组织: Hangzhou Alibaba Advertising Co.,Ltd.

2020-11-08



端口排名

4782 48

443 5

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/pavelskay/Xtremis	-	Original	No
QuasarStrike	https://github.com/QuasarStrike/QuasarStrike	-	Original	No
VenomRAT	not published	-	Original	No
RSMaster	https://github.com/Netsk yes/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

Use the same
communication
method.

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

Quasar Family Packet Format

Size(4byte)

HMAC(32byte)

IV(16byte)

QuickLZ + AES(mode CBC)

C2 First Response

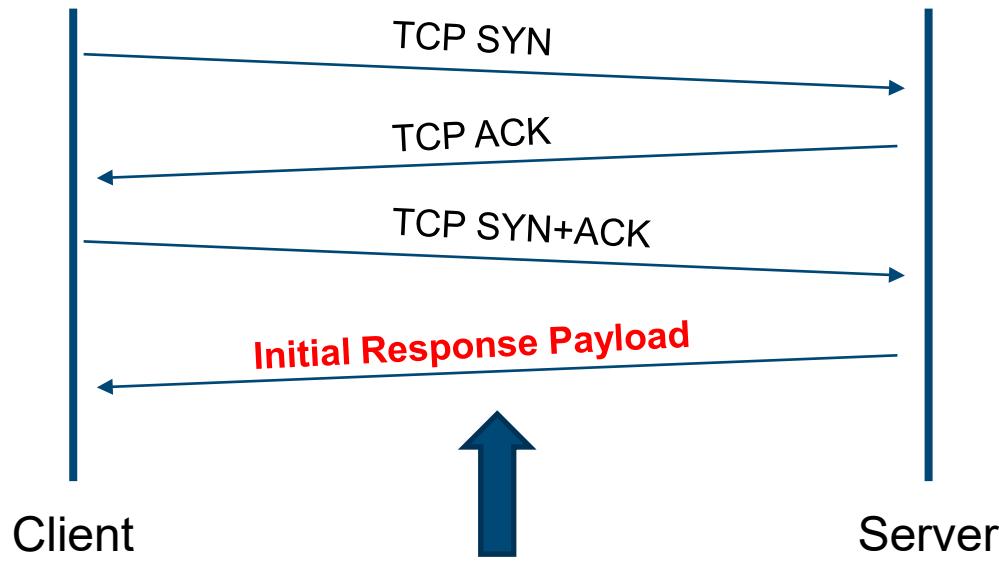
```
@vmhost:~/svn/data/QuasarRAT$ xxd server_send_data_1st
00000000: 4000 0000 cc19 36aa db4e 7665 69c8 1095 @.....6..Nvei...
00000010: 5b72 38a1 34f4 7698 64a0 29fb c054 9df3 [r8.4.v.d.)...T..
00000020: c022 9f97 59e0 e268 40fb 3d84 37e0 f85c ."..Y..h@.=.7..\.
00000030: 5a83 5c63 9775 c651 0d32 1477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



First response size = 0x44

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

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	108.160.130.94	JP	TCP	80	{108.160.130.94 Reliable .vultr.com}	Servers LLC	@¥x00¥x00¥x00¥x00¥x08¥xfc¥xaeQH¥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¥xcaxde¥xf5-Ah¥xec¥xd3¥x9f¥x8f¥xb77B¥x0b¥x01¥x198
2020-03-16 02:30:56.020	198.13.54.11	JP	TCP	443	{www.google.co m}	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¥xc1¥x11¥x03¥xa0:¥xfeM¥x8a¥x0b¥xf9¥xc9¥xe9¥xc6¥xa9&¥xc0n¥x82
2019-12-22 10:58:09.512	45.32.51.172	JP	TCP	8080	{45.32.51.172.v ultr.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 j¥xde
2019-12-29 15:12:49.552	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¥¥!¥x85h¥xf4&¥xe5_¥xf0T¥x0e¥xea¥x1b¥xe7Y¥x90S¥x90!¥xd4P¥x0b¥x0f+¥xb4¥x91,M¥xa8¥x1f

Quasar Servers on the Internet

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) : R4WS01Gysgjic0jFGVpFz3Y3gZ6n9yS5m5HqjtkK8xb7PW5GAGUkktId4YSptS9ffyKNQxWQ0rH6RIwL6Um7A==  
SUBDIRECTORY    : SubDir  
INSTALLNAME     : Client.exe  
MUTEX            : QSR_MUTEX_RjrwbI1Icgf7POFqOF  
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 NcFtjbD0csw7Evd3coMC0y4koy/SRZGydhNmno81Z0W0vdfg7sv0Cj5ad2R0
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-0201..
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 Gr7icnpxp4j1oWFhmLnbfFe2f  
nmzIMEiuuCNQ20SsJ24cE3bQdT/zV0bPVlgENoELTQQM3hm50qk0CokZ2cn0g== -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/SRZGydhNmno81Z0W0vdfg7sv0Cj5ad2R0UfX4QMscAIjYJdjrrs41+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

```
keybaseinfo.py
test@debian:~$ python3 quasarrat_client.py -s 210.144.121.100 -p 4782 -k 1WvgEMPj
coMC0y4koy/SRZGydhNmno81Z0W0vdfg7sv0Cj5ad2R0UfX4QMscAIjYJdjrrs41+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: 127.0.0.1)
  -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

IP Address	Tag	User@PC	Version	Status	User Status	Country	Operating System	Account Type
# 210.144.121.100	Office04	kann@WIN7_64EN-O2013	1.3.0.0	Connected	Active	Unknown [-]	Windows 7 Ultimate 64 Bit	User
# 210.144.121.200	Office04	Japan@Brazil	1.3.0.0	Connected	Active	Unknown [SaudiArabia]	Android	India
# 210.144.121.200	Office04	Brazil@Japan	1.3.0.0	Connected	Active	Unknown [Turkey]	iOS	Japan
# 210.144.121.200	Office04	Brazil@China	1.3.0.0	Connected	Active	Unknown [Mexico]	Windows 7 32 Bit	India
# 210.144.121.200	Office04	India@Argentina	1.3.0.0	Connected	Active	Unknown [Argentina]	Windows 8 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	Indonesia@UnitedState	1.3.0.0	Connected	Active	Unknown [Brazil]	macOS	SaudiArabia
# 210.144.121.200	Office04	Italy@UnitedKingdom	1.3.0.0	Connected	Active	Unknown [Brazil]	Linux	Turkey
# 210.144.121.200	Office04	Indonesia@SaudiArabia	1.3.0.0	Connected	Active	Unknown [Brazil]	Android	Brazil
# 210.144.121.200	Office04	Frence@India	1.3.0.0	Connected	Active	Unknown [Canada]	Linux	Russia
# 210.144.121.200	Office04	Brazil@France	1.3.0.0	Connected	Active	Unknown [Australia]	Android	India
# 210.144.121.200	Office04	Indonesia@China	1.3.0.0	Connected	Active	Unknown [SouthKorea]	iOS	SaudiArabia
# 210.144.121.200	Office04	India@SouthKorea	1.3.0.0	Connected	Active	Unknown [Argentina]	macOS	China
# 210.144.121.200	Office04	UnitedState@Canada	1.3.0.0	Connected	Active	Unknown [Canada]	macOS	Canada
# 210.144.121.200	Office04	Brazil@Brazil	1.3.0.0	Connected	Active	Unknown [France]	iOS	India
# 210.144.121.200	Office04	Russia@SouthKorea	1.3.0.0	Connected	Active	Unknown [Germany]	Android	Japan
# 210.144.121.200	Office04	Brazil@SouthKorea	1.3.0.0	Connected	Active	Unknown [SouthKorea]	iOS	Canada
# 210.144.121.200	Office04	Turkey@Russia	1.3.0.0	Connected	Active	Unknown [Indonesia]	Windows XP 32 Bit	Indonesia
# 210.144.121.200	Office04	Mexico@SaudiArabia	1.3.0.0	Connected	Active	Unknown [Brazil]	iOS	China
# 210.144.121.200	Office04	Frence@Indonesia	1.3.0.0	Connected	Active	Unknown [Germany]	Windows 7 32 Bit	UnitedKingdom
# 210.144.121.200	Office04	Italy@Italy	1.3.0.0	Connected	Active	Unknown [Italy]	Windows 8 Enterprise 64 Bit	UnitedKingdom
# 210.144.121.200	Office04	Turkey@Russia	1.3.0.0	Connected	Active	Unknown [SouthAfrica]	Windows 10 Enterprise 64 Bit	Canada
# 210.144.121.200	Office04	Italy@Turkey	1.3.0.0	Connected	Active	Unknown [China]	macOS	Italy
# 210.144.121.200	Office04	Argentina@Turkey	1.3.0.0	Connected	Active	Unknown [UnitedKingdom]	Android	Italy
# 210.144.121.200	Office04	Japan@SouthAfrica	1.3.0.0	Connected	Active	Unknown [SouthAfrica]	Windows 10 Enterprise 64 Bit	Japan
# 210.144.121.200	Office04	China@Brazil	1.3.0.0	Connected	Active	Unknown [Italy]	Windows 10 Enterprise 64 Bit	India
# 210.144.121.200	Office04	Italy@SouthAfrica	1.3.0.0	Connected	Active	Unknown [Canada]	Linux	UnitedKingdom
# 210.144.121.200	Office04	Indonesia@Australia	1.3.0.0	Connected	Active	Unknown [Turkey]	Windows XP 32 Bit	UnitedState
# 210.144.121.200	Office04	Italy@Mexico	1.3.0.0	Connected	Active	Unknown [Australia]	Windows XP 32 Bit	UnitedKingdom
# 210.144.121.200	Office04	UnitedKingdom@Indonesia	1.3.0.0	Connected	Active	Unknown [Japan]	Windows 8 Enterprise 64 Bit	SouthKorea
# 210.144.121.200	Office04	Brazil@France	1.3.0.0	Connected	Active	Unknown [Japan]	iOS	Mexico
# 210.144.121.200	Office04	SouthAfrica@Brazil	1.3.0.0	Connected	Active	Unknown [Argentina]	macOS	Canada
# 210.144.121.200	Office04	SouthKorea@Australia	1.3.0.0	Connected	Active	Unknown [Indonesia]	iOS	India
# 210.144.121.200	Office04	Italy@SaudiArabia	1.3.0.0	Connected	Active	Unknown [Germany]	Windows 10 Enterprise 64 Bit	Russia
# 210.144.121.200	Office04	Australia@SouthKorea	1.3.0.0	Connected	Active	Unknown [Italy]	Windows 10 Enterprise 64 Bit	Germany
# 210.144.121.200	Office04	SaudiArabia@Canada	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	Russia@Indonesia	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	India@SouthKorea	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	France@SaudiArabia	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	India@Russia	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	Japan@Germany	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	Australia@Australia	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState
# 210.144.121.200	Office04	Iranan@Iranan	1.3.0.0	Connected	Active	Unknown [India]	Windows 10 Enterprise 64 Bit	UnitedState

You can connect many fake clients and stack C2!

Fake Malware DoS Attack for C2

- Connected: 1001					Operating System			
IP	Version	Status	User Sta					
210.144.121.200	Office04	Japan@Brazil	1.3.0.0	Connected	Active	Windows 7 Ultimate 64 Bit		
210.144.121.200	Office04	Brazil@Japan	1.3.0.0	Connected	Active	Android		
210.144.121.200	Office04	Brazil@China	1.3.0.0	Connected	Active	iOS		
210.144.121.200	Office04	India@Argentina	1.3.0.0	Connected	Active	Windows 7 32 Bit		
210.144.121.200	Office04	Indonesia@UnitedState	1.3.0.0	Connected	Active	Windows 8 Enterprise 64 Bit		
210.144.121.200	Office04	Italy@UnitedKingdom	1.3.0.0	Connected	Active	macOS		
210.144.121.200	Office04	Indonesia@SaudiArabia	1.3.0.0	Connected	Active	Linux		
210.144.121.200	Office04	France@India	1.3.0.0	Connected	Active	Android		
210.144.121.200	Office04	Brazil@France	1.3.0.0	Connected	Active	Linux		
210.144.121.200	Office04	Indonesia@China	1.3.0.0	Connected	Active			
210.144.121.200	Office04	India@SouthKorea	1.3.0.0	Connected	Active			
210.144.121.200	Office04	UnitedState@Canada	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Brazil@Brazil	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Russia@SouthKorea	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Brazil@SouthKorea	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Turkey@Russia	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Mexico@SaudiArabia	1.3.0.0	Connected	Active			
210.144.121.200	Office04	France@Indonesia	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Italy@Italy	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Turkey@Russia	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Italy@Turkey	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Argentina@Turkey	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Japan@SouthAfrica	1.3.0.0	Connected	Active			
210.144.121.200	Office04	China@Brazil	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Italy@SouthAfrica	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Indonesia@Australia	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Italy@Mexico	1.3.0.0	Connected	Active			
210.144.121.200	Office04	UnitedKingdom@Indonesia	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Brazil@France	1.3.0.0	Connected	Active			
210.144.121.200	Office04	SouthAfrica@Brazil	1.3.0.0	Connected	Active			
210.144.121.200	Office04	SouthKorea@Australia	1.3.0.0	Connected	Active			
210.144.121.200	Office04	Italy@SaudiArabia	1.3.0.0	Connected	Active	Unknown [Germany]	Windows 10 Enterprise 64 Bit	Russia
210.144.121.200	Office04	Australia@SouthKorea	1.3.0.0	Connected	Active	Unknown [Italy]	Linux	Germany
210.144.121.200	Office04	SaudiArabia@Canada	1.3.0.0	Connected	Active	Unknown [India]	macOS	UnitedState
210.144.121.200	Office04	Russia@Indonesia	1.3.0.0	Connected	Active	Unknown [Russia]	Linux	Japan
210.144.121.200	Office04	India@SouthKorea	1.3.0.0	Connected	Active	Unknown [India]	macOS	Canada
210.144.121.200	Office04	France@SaudiArabia	1.3.0.0	Connected	Active	Unknown [Turkey]	Windows 10 Enterprise 64 Bit	France
210.144.121.200	Office04	India@Russia	1.3.0.0	Connected	Active	Unknown [Indonesia]	Android	SouthKorea
210.144.121.200	Office04	Japan@Germany	1.3.0.0	Connected	Active	Unknown [China]	Windows 10 Enterprise 64 Bit	Germany
210.144.121.200	Office04	Australia@Australia	1.3.0.0	Connected	Active	Unknown [SouthAfrica]	Android	UnitedState
210.144.121.200	Office04	Iranan@Iranan	1.3.0.0	Connected	Active	Unknown [SouthAfrica]	Windows 8 Enterprise 64 Bit	UnitedKingdom

Listening on port 4782.



How to Download

JPCERT/CC Github



The screenshot shows the GitHub interface for the JPCERT Coordination Center's repository. At the top, there is a search bar, a pull requests button, an issues button, a marketplace button, and an explore button. A notification bell icon and a plus sign are also present. The main header reads "JPCERT Coordination Center" with a red stylized "J" logo. Below the header, it says "JPCERT/CC's official repositories maintained by staff and guests" and "Tokyo, Japan". A link to the website "https://www.jpcert.or.jp/" is provided. The navigation bar includes "Repositories 55", "Packages", "People 27", "Teams 8", "Projects", and "Settings". The "Repositories" tab is currently selected. Below the navigation bar, the section "Pinned repositories" lists several projects:

- LogonTracer**: Investigate malicious Windows logon by visualizing and analyzing Windows event log. Python, 1.5k stars, 304 forks.
- aa-tools**: Artifact analysis tools by JPCERT/CC Analysis Center. Python, 327 stars, 72 forks.
- ToolAnalysisResultSheet**: Tool Analysis Result Sheet. HTML, 234 stars, 50 forks.
- SysmonSearch**: Investigate suspicious activity by visualizing Sysmon's event log. JavaScript, 276 stars, 44 forks.
- MalConfScan-with-Cuckoo**: Cuckoo Sandbox plugin for extracts configuration data of known malware. Python, 105 stars, 15 forks.
- MalConfScan**: Volatility plugin for extracts configuration data of known malware. Python, 294 stars, 47 forks.

At the bottom of the pinned repositories section, there is a "Customize pinned repositories" link. Below the pinned repositories, there is a search bar with the placeholder "Find a repository...", a "Type: All" dropdown, a "Language: All" dropdown, and a green "New" button.

<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

Thank you!



@jpcert_en



ir-info@jpcert.or.jp

PGP <https://www.jpcert.or.jp/english/pgp/>