

# Incident Response Report

---



---

Miles Workstation

November 2021

**Company Sensitive and Proprietary  
For Authorized Use Only**

**Authors:**



**Alexandre OHAYON**



**Erwan SINOU**



**Eric BELLOTTO**



**Theodore FARAUT**

**RESTRICTED INFORMATION  
DISTRIBUTION FOR OFFICIAL USE ONLY**

All copies must follow statewide protection and handling standards. Destroy by shredding.

## CERT TEAM MEMBERS

Name	Role	Contact
Alexandre OHAYON	Forensic Operator, Code analyzer, technical writer	alexandre.ohayon@epitech.eu
Erwan SINOU	Technical Writer, Forensic Operator	erwan.sinou@epitech.eu
Eric BELLOTTO	Script writer and code analyzer	eric.bellotto@epitech.eu
Theodore FARAUT	Script writer and code analyzer	theodore.faraut@epitech.eu

## DOCUMENT CHANGE LOG

Version	Date	Comment	Author(s)
0.1	28/10/2021	[ADD] structure of the report	alexandre.ohayon@epitech.eu, erwan.sinou@epitech.eu
0.2	29/10/2021	[ADD] explanation of the intrusion of the workstation, findings some malwares	eric.bellotto@epitech.eu, alexandre.ohayon@epitech.eu, theodore.faraut@epitech.eu
0.3	01/11/2021	[ADD] redaction of the cleaning scripts	alexandre.ohayon@epitech.eu eric.bellotto@epitech.eu

## EXECUTIVE SUMMARY

The purpose of this system security report is to provide an overview of the security failures of the system and describe the controls in place, or planned, for meeting those requirements.

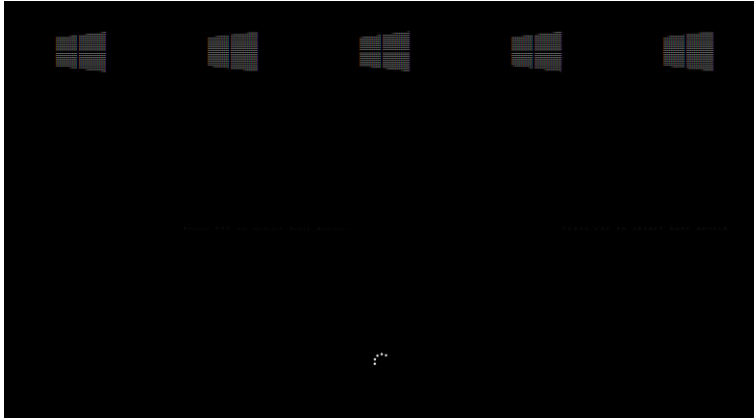
Miles Workstation is a Windows 10 operating system; we do not know so many things except there is a strange behavior from the system and we don't have the password.

## TABLE OF CONTENTS

CERT TEAM MEMBERS.....	2
DOCUMENT CHANGE LOG .....	2
EXECUTIVE SUMMARY .....	3
ASSESSMENT SPECIFICATIONS.....	4
KEY FINDINGS .....	15
MAIN REMEDIATION ADVICE .....	17
SETUP.....	17
METHODOLOGY .....	17

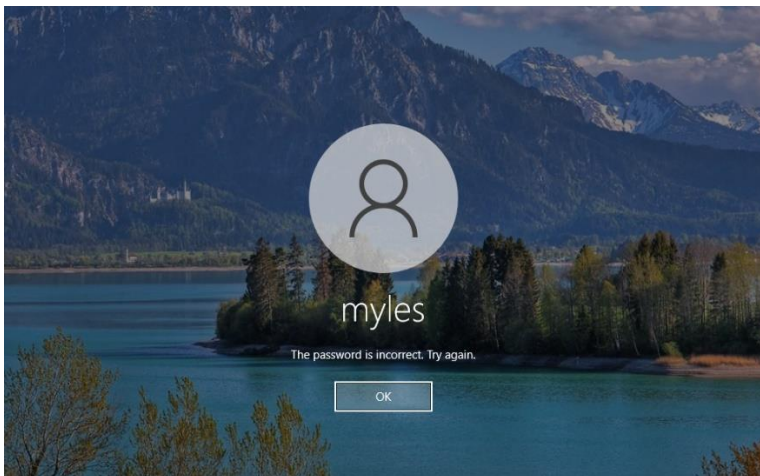
## ASSESSMENT SPECIFICATIONS

First, we launch the Miles Workstation in a virtual machine, using Virtual Box. We can see a strange behavior at the start of Windows.



Screen capture of the start of the Miles Workstation

Once the operating system is launched, we observe that the myles user is protected by a password that we do not have.



Screen capture of the lock screen of the Miles Workstation

The workstation is not encrypted. We can get into the machine by setting the password blank.

We attach the Miles Workstation as a drive into our Linux instance from VirtualBox. We can see it from Kali with the tool gparted by the command line: parted -l

```
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 85.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
  1      1049kB  84.9GB  84.9GB  primary  ext4          boot
  2      84.9GB  85.9GB  1022MB  extended
  5      84.9GB  85.9GB  1022MB  logical  linux-swap(v1)

Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
  1      1049kB  53.5MB  52.4MB  primary  ntfs          boot
  2      53.5MB  42.4GB  42.4GB  primary  ntfs
  3      42.4GB  42.9GB  530MB   primary  ntfs          msftres
```

Screen capture of the Miles Workstation disk from Kali Linux

We can see that the disk from /dev/sdb is with the msdos partition table. The content of the files from the operating system are located on /dev/sdb2. We mount it with the command line: mount -t ntfs -o noexc, rw /dev/sdb2 /mnt. Then, we move were the list of the users of the Workstation is located and we display it with the command: chntpw -l SAM.

```
(kali㉿kali)-[~]
$ sudo mount -t ntfs -o noexc,rw /dev/sdb2 /mnt

(kali㉿kali)-[~]
$ cd /mnt/Windows/System32/config

(kali㉿kali)-[/mnt/Windows/System32/config]
$ sudo chntpw -l SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive <SAM> name (from header): <\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 686c <lh>
File size 65536 [10000] bytes, containing 7 pages (+ 1 headerpage)
Used for data: 318/31800 blocks/bytes, unused: 29/13032 blocks/bytes.

| RID | Username | Admin? | Lock? |
|-----|-----|-----|-----|
| 01f4 | Administrator | ADMIN | dis/lock |
| 01f7 | DefaultAccount |  | dis/lock |
| 01f5 | Guest |  | dis/lock |
| 03e9 | myles | ADMIN |  |
| 01f8 | WDAGUtilityAccount |  | dis/lock |
```

Screen capture of the users of the Miles Workstation

Next, we clear the password of Miles with the command: `chntpw -u myles SAM`.

```
(kali@kali)-[/mnt/Windows/System32/config]
└─$ sudo chntpw -u myles SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive <SAM> name (from header): <\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 686c <lh>
File size 65536 [10000] bytes, containing 7 pages (+ 1 headerpage)
Used for data: 318/31800 blocks/bytes, unused: 29/13032 blocks/bytes.

===== USER EDIT =====
RID      : 1001 [03e9]
Username: myles
fullname:
comment:
homedir:

00000220 = Administrators (which has 2 members)

Account bits: 0x0214 =
[ ] Disabled           [ ] Homedir req.      [X] Passwd not req.
[ ] Temp. duplicate    [X] Normal account  [ ] NMS account
[ ] Domain trust ac    [ ] Wks trust act.  [ ] Srv trust act
[X] Pwd don't expir    [ ] Auto lockout   [ ] (unknown 0x08)
[ ] (unknown 0x10)     [ ] (unknown 0x20)  [ ] (unknown 0x40)

Failed login count: 0, while max tries is: 0
Total login count: 20

- - - User Edit Menu:
1 - Clear (blank) user password
2 - Unlock and enable user account [seems unlocked already]
3 - Promote user (make user an administrator)
4 - Add user to a group
5 - Remove user from a group
q - Quit editing user, back to user select
Select: [q] > 1
Password cleared!

===== USER EDIT =====
RID      : 1001 [03e9]
Username: myles
fullname:
comment:
homedir:

00000220 = Administrators (which has 2 members)

Account bits: 0x0214 =
[ ] Disabled           [ ] Homedir req.      [X] Passwd not req.
[ ] Temp. duplicate    [X] Normal account  [ ] NMS account
[ ] Domain trust ac    [ ] Wks trust act.  [ ] Srv trust act
[X] Pwd don't expir    [ ] Auto lockout   [ ] (unknown 0x08)
[ ] (unknown 0x10)     [ ] (unknown 0x20)  [ ] (unknown 0x40)

Failed login count: 0, while max tries is: 0
Total login count: 20

- - - User Edit Menu:
1 - Clear (blank) user password
2 - Unlock and enable user account [seems unlocked already]
3 - Promote user (make user an administrator)
4 - Add user to a group
5 - Remove user from a group
q - Quit editing user, back to user select
Select: [q] > q

Hives that have changed:
# Name
0 <SAM>
Write hive files? (y/n) [n] : y
0 <SAM> - OK
```

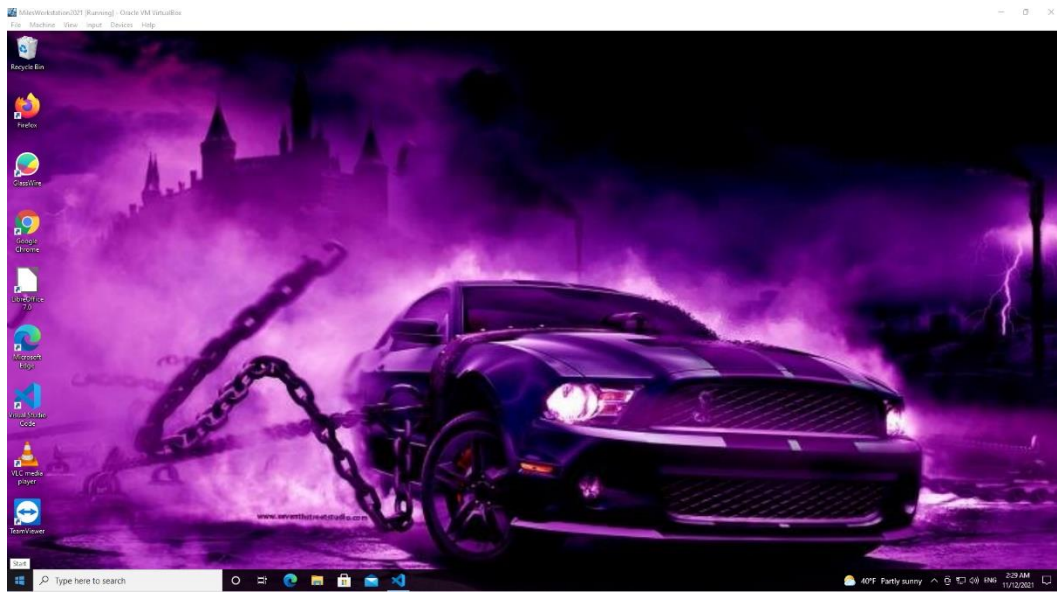
Screen captures of the clearing of the password of the user myles

We reboot on the Miles Workstation, and we can now enter in the session myles.

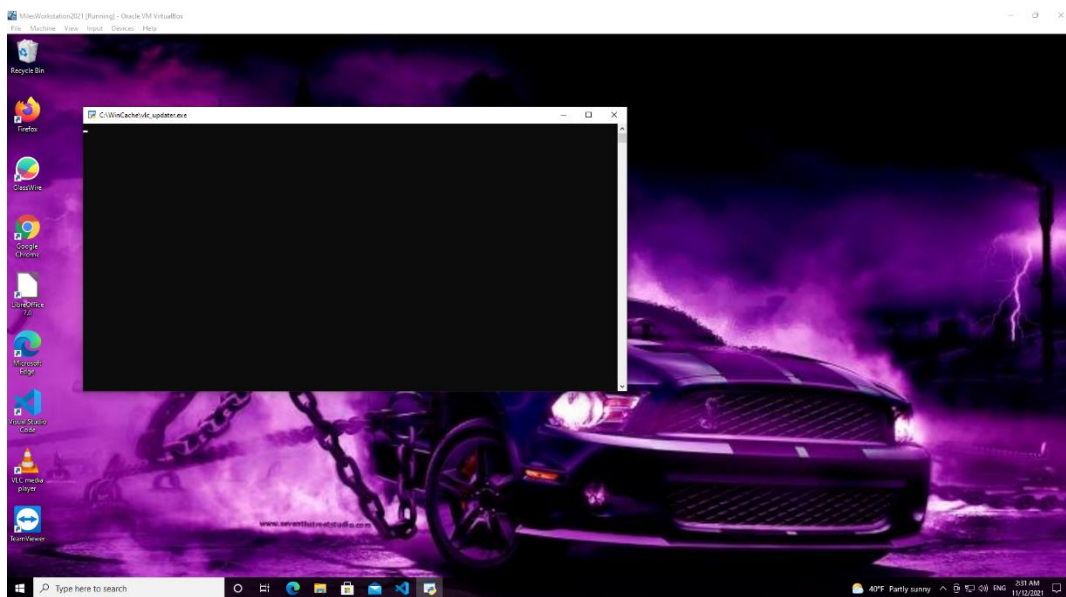


Screen captures showing the password of myles is now blank

Once in the workstation, we see a weird wallpaper and terminals that open on startup.

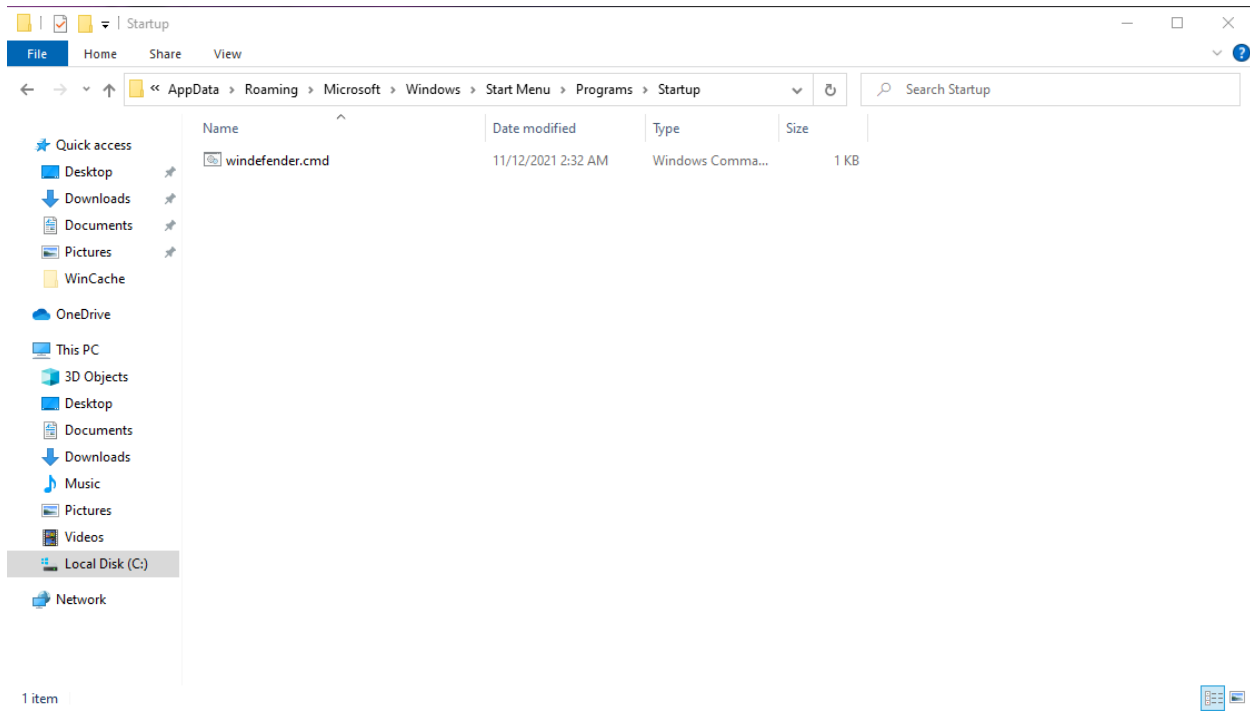


*Screen captures showing the wallpaper*



*Screen captures showing the terminal at startup*

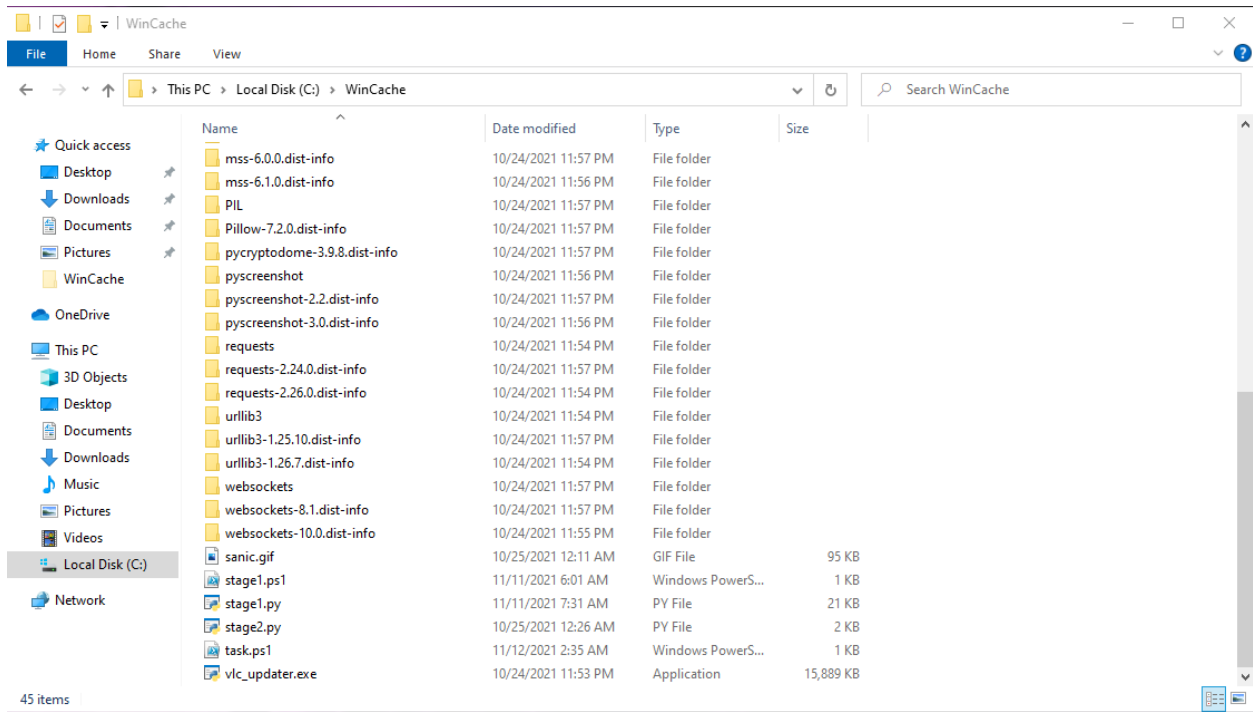
We are thinking of looking directly in the folder where the windows startup executables are located in C: \ Users \ myles \ AppData \ Roaming \ Microsoft \ Windows \ Start Menu \ Programs \ Startup



*Screen captures showing windefender.cmd*

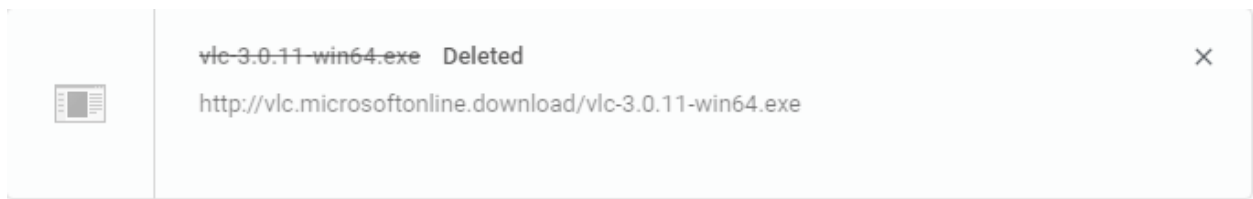
We come across a malicious program hiding under the name of a Windows security program, which executes another python program with a ntfs method (wallpaper.jpeg::stage2.py). This program runs a program located in a cache folder at the root of the disk.





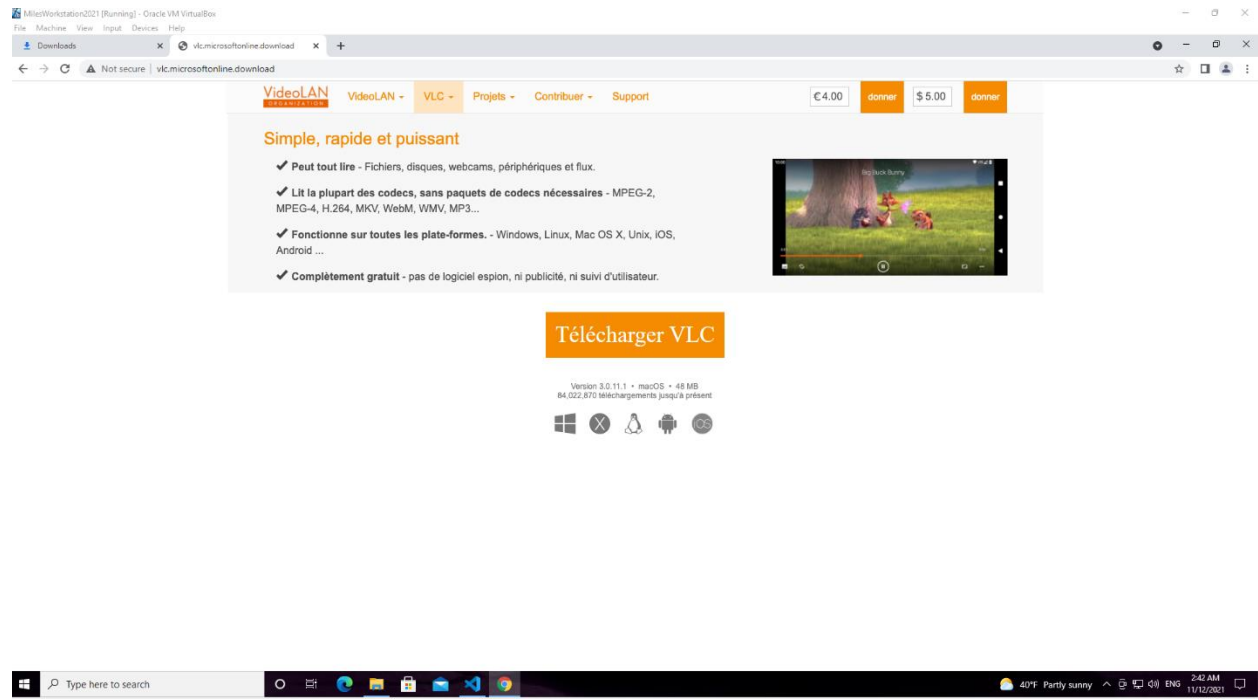
Screen captures showing the WinCache folder

We can see that the executable vlc\_updater.exe is actually a python interpreter that allows malicious code to be executed. We are thinking of looking in the downloaded files to understand how a fake vlc could have been installed.



Screen captures showing the password of myles is now blank

We go to the address indicated and we come across a fake scam site, which is the main entry point where the dropper has been installed on the machine



*Screen captures showing the fake vlc website*

Before we start to analyze the code, we want to find out more about the author of the virus.

```
aohayon@asus:/mnt/c/Users/work$ dig NS 104.238.188.130
; <<>> DiG 9.16.1-Ubuntu <<>> NS 104.238.188.130
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NXDOMAIN, id: 5480
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;104.238.188.130.          IN      NS
;; AUTHORITY SECTION:
.                314     IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2021111100 1800 900 604800 86400
;; Query time: 20 msec
;; SERVER: 172.28.96.1#53(172.28.96.1)
;; WHEN: Thu Nov 11 13:16:14 CET 2021
;; MSG SIZE rcvd: 119
```

*Screen captures showing informations about the fake website*

We find several information:

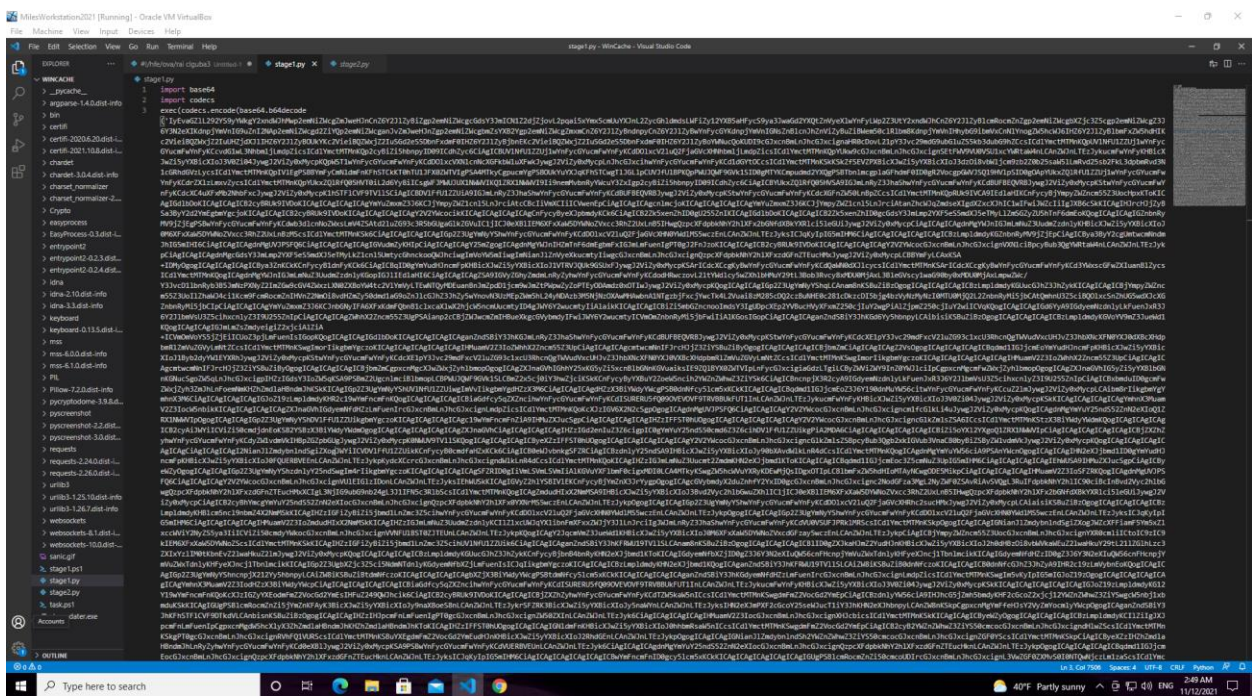
Registrar URL: [www.ovh.com](http://www.ovh.com)

Updated Date: 2021-07-06T16:47:11Z

Creation Date: 2019-07-11T13:47:10Z

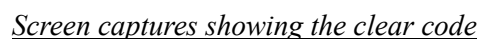
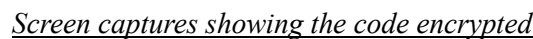
Registry Expiry Date: 2022-07-11T13:47:10Z

The dropper seems to be hosted at OVH, on Ubuntu. We now turn to the code. The code is encoded.

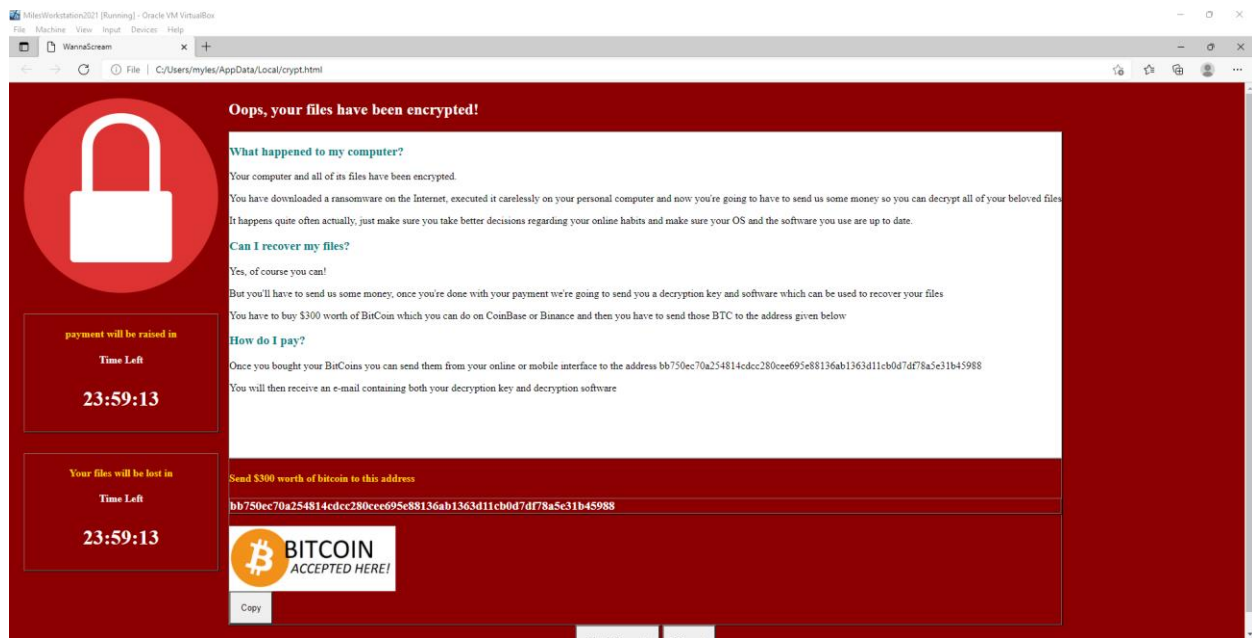


Screen captures showing the code encoded in base 64

We decode it. But the code is encrypted. We need to find its encryption.



We can see that the virus is a trojan. This one has installed a keylogger and takes screenshots regularly. It gives an identification number to the machine uid. It also hides data in a gif. It uses this hidden data in the GIF to download data, send it, or run programs on the computer. The program can therefore launch ransomware.



Screen captures showing the ransomware

We tried to connect to the virus server using our script, but we did not succeed. The script is available in the folder project.

```
C:\Users\work> Downloads > JS scripts > client.on('connect') callback > connection.on('message') callback
4  var CryptoJS = require("crypto-js");
5  var fs = require('fs');
6  var client = new WebSocketClient();
7
8  client.on('connectFailed', function (error) {
9    console.log('Connect Error: ' + error.toString());
10 });
11
12 client.on('connect', function (connection) {
13   console.log('WebSocket Client Connected');
14   connection.on('error', function (error) {
15     console.log("Connection Error: " + error.toString());
16   });
17   connection.on('close', function () {
18     console.log('echo-protocol Connection Closed');
19   });
20   connection.on('message', function (message) {
21     console.log("-", message);
22     fs.writeFileSync("./t", message.binaryData);
23     var binary = fs.readFileSync("./t");
24     var adecrypt = message.binaryData.slice(0, 48).toString('utf-8');
25     console.log(message.binaryData, adecrypt);
26     var bytes = AES.decrypt(adecrypt, "UXRL_HREF\\nqzva");
27
28     var originalText = bytes.toString(CryptoJS.enc.Utf8);
29
30
31     console.log("--->", originalText);
32     //console.log("HERE", JSON.parse(JSON.stringify(message.binaryData)).data)
33     /*var ciphertext = AES.encrypt('gif', 'ebg-13').toString();
34     var number = Math.round(Math.random() * 0xFFFFFFFF);
35     //connection.sendUTF(number.toString());
36     console.log("--->", ciphertext)
37     //console.log("\n\n\n", message.binaryData)
38     //console.log(message.binaryData.toString('utf8'))
39     var bytes = AES.decrypt(message.binaryData.toString('utf8'), 'ebg-13');
40     //console.log(JSON.parse(JSON.stringify(message.binaryData).toString('utf8')).data)
41     var fileBuffer = new Buffer(message.binaryData, 'binary');
42     //console.log(bytes)
43     var file = message.binaryData.toString('utf-8');
44     console.log(file)
45     if (message.type === 'utf8') {
46       //console.log("Received: " + message.utf8Data + "");
47     }*/
48   });
49 }
50
51 function sendNumber() {
52   if (connection.connected) {
```

Screen captures showing our script

## KEY FINDINGS

<u>Name</u>	<u>Description</u>	<u>Location</u>	<u>Content</u>
<u>Windefender.cmd</u>		<u>C:\Users\myles\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup</u>	<u>wmic process call create "C:\Users\myles\Pictures\wallpaper.jpeg:py.exe C:\Users\myles\Pictures\wallpaper.jpeg:stage2.py</u>
<u>vlc_updater.exe</u>	<u>Python interpretor</u>	<u>C:\WinCache</u>	<u>Python executable</u>
<u>task.ps1</u>	<u>Powershell program that launch stage1.ps1 at the start up</u>	<u>C:\WinCache</u>	<u>schtasks /create /F /IT /tn WinCache /tr "powershell C:\WinCache\stage1.ps1" /sc onlogon /ru System</u>
<u>stage1.ps1</u>	<u>Powershell program that launch stage1.py and vlc_updater.exe in hidden mode</u>	<u>C:\WinCache</u>	<u>powershell.exe -windowstyle hidden -c "type C:\WinCache\stage1.py   C:\WinCache\vlc_updater.exe</u>
<u>stage1.py</u>	<u>Obfuscated code with aes 13, 2 times.</u> <u>Keylogger, screen catcher, all the virus</u>	<u>C:\WinCache</u>	
<u>stage2.py</u>	<u>Obfuscated code with aes 13, 2 times.</u> <u>Keylogger, screen catcher, all the virus</u>	<u>C:\WinCache</u>	



<u>sanic.gif</u>	<u>Gif containing a code :</u>	<u>C:\WinCache</u>	<u>HIDDEN_CONTENT SEPARATOR=</u> <u>“instructions”</u>
<u>wallpaper.jpeg</u>	<u>File used to use an ntfs method to launch stage2.py</u>	<u>C:\Users\myle</u> <u>s\Pictures</u>	
<u>Keys.log</u>	<u>keylogger</u>	<u>C:\Users\myle</u> <u>s\AppData\Local\keys.log</u>	
<u>Crypt.html</u>	<u>ransomware</u>	<u>C:\Users\myle</u> <u>s\AppData\Local\crypt.html</u>	
<u>Lastscreen.png</u>	<u>Screenshot taker</u>	<u>C:\Users\myle</u> <u>s\AppData\Local</u>	



## MAIN REMEDIATION ADVICE

Our advice is to pay attention to the sites and always check if the site where you download an application is the official site.

We have developed a script to remove the virus. He is in the GitHub folder where this document is also located.

- **Step 1** – Open the command prompt (cmd.exe).
- **Step 2** – Go to the location where the .bat or .cmd file is stored.
- **Step 3** – Write the name of the file as shown in the following image and press the Enter button to execute the batch file.

## SETUP

We used VirtualBox, Kali Linux. We also used glasswire, burp suite, Wireshark as tools.

## METHODOLOGY

We met every day, five days in a row.

One has the project fully set and share his screen.

One is helping the screen sharer.

One is writing everything found.

One is analyzing the code or developing scripts.