



# Seguridad en Open Source



# Introducción

- Seguridad basada en un sentimiento.
- Seguridad Total: El mayor mito
- Certificaciones vs Experiencia
- y al final ... ¿Qué?



# Seguridad

- Tener seguridad es estar seguro.
- ¿Estar seguro . . . de que?
- Conocer los riesgos
- Factibilidad de riesgos



# Riesgos Informáticos

- Discutidos hasta la saciedad
- Nunca están previstos en BCP /  
DRP y planes de seguridad
- No todos son hackers y no  
todos los hackers son malos
- No debe molestar el ruidoso,  
debe molestar el callado.



# ¿Qué tan seguro es el Open Source?

- Tan seguro como el que lo programa.
- Un millón de gente que no sabe que busca no lo hace mas seguro.
- Ejemplos: twitter, 80% de las aplicaciones de PHP, C, perl, etc.



# ¿Qué lenguaje es seguro?

- Tan fácil como decir:
  - Ninguno
- No es el lenguaje es el que programa en el lenguaje
- La mentalidad mata
- La ignorancia se paga



# Ejemplos

- PHP: `include($base_file);`
- C: `sprintf(foo, "%s", bar);`
- perl: `open(FILE, $user_file);`
- Errores lógicos y de diseño
  - Los mas costosos
  - A veces imposibles de arreglar





## ¿Vida Real?

```
File Edit View Terminal Tabs Help
#include <stdlib.h>
#include <string.h>
nahual@neuromancer:~$ gcc -o kernel_exploit -static -Wno-format kernel_exploit.c
nahual@neuromancer:~$ uname -a; id
Linux neuromancer 2.6.22-14-generic #1 SMP Fri Feb 1 04:59:50 UTC 2008 i686 GNU/
Linux
uid=1000(nahual) gid=1000(nahual) groups=4(adm),20(dialout),24(cdrom),25(floppy)
,29(audio),30(dip),44(video),46(plugdev),104(scanner),112(netdev),113(lpadmin),1
15(powerdev),117(admin),1000(nahual)
nahual@neuromancer:~$ ./kernel_exploit
-----
Linux vmsplince Local Root Exploit
By qaaz
-----
[+] mmap: 0x0 .. 0x1000
[+] page: 0x0
[+] page: 0x20
[+] mmap: 0x4000 .. 0x5000
[+] page: 0x4000
[+] page: 0x4020
[+] mmap: 0x1000 .. 0x2000
[+] page: 0x1000
[+] mmap: 0xb7fb8000 .. 0xb7fea000
[+] root
root@neuromancer:~# uname -a; id
Linux neuromancer 2.6.22-14-generic #1 SMP Fri Feb 1 04:59:50 UTC 2008 i686 GNU/
Linux
uid=0(root) gid=0(root) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio)
,30(dip),44(video),46(plugdev),104(scanner),112(netdev),113(lpadmin),115(powerde
v),117(admin),1000(nahual)
root@neuromancer:~# exit
exit
nahual@neuromancer:~$
```





# Errores de Diseño

- Son la pesadilla de cualquier programador.
- Si tu software NECESITA root para correr:
  - ¿Porque requiere root?
  - ¿Necesita hacer TODO como root?
  - Si no piensas como te van a atacar ... 8(



# Errores de Diseño

- Errores encontrados durante mis consultorias:
  - Apache DEBE correr como root
  - La base de datos es “distribuida” (unas tablas aqui y otras alla)
  - Cliente -> PHP -> C (root) -> mysql



# Errores de Diseño

- Socket de DB sin password
- WWW en DMZ, DB en Zona Corporativa



# La Ignorancia mata

- No es que el programador sea malicioso, de verdad no conoce los riesgos.
- Actitud de saber todo
- Certificaciones no ayudan en el punto anterior.



# La Ignorancia mata

- Frases que te hacen llorar:
  - “Java no es software libre” (+)
  - El OpenSource es mas seguro que el Closed Source porque es abierto. (?)
  - “Funciona, no le muevas”
  - “La información quiere ser libre”



# Programación Segura

- La programación segura es como un deporte extremo.
- Un programador seguro conoce los riesgos.
- Al mejor cazador .. Se le va un fallo.



# Programación Segura

- ¿Código Seguro?

```
function upload_file() {  
    $file = _GET['file'];  
    @ext = splice('.', $file);  
    if( $ext[1] != "html") {  
        not_permited();  
    }  
    else {  
        set_file_public_html();  
    }  
}
```



# Programación Segura

- ¿Código Seguro?

```
#!/usr/bin/perl
include CGI;
$file = CGI::parameter['file'];
open(FILE, $file) or die "Cannot open file $file:$_!\n";
while(<FILE>) {
    print;
}
close(FILE);
```



# Programación Segura

- ¿Código Seguro?

```
#include <stdio.h>
int main(int argc, char **argv) {
    char b[512];
    sprintf(b, "/usr/sbin/myexec %s", argv[1]);
    system(b);
    return 0;
}
```



# Programación Segura

- ¿Todo esta perdido?
- Soluciones de ayuda
  - PaX
  - Parches Glibc
  - Virtualización
  - Compartimentalización
  - Rezar ayuda (el milagrito)



# Programación Segura

- Capacitación
- Romper Esquemas
- “Terapia de choque”
- Cajas Mágicas
- Soluciones Mágicas



# Capacitación

- En reino de ciegos, el tuerto es rey (?)
- ¿Quién hace las mejores alarmas de carros?
- Los atacantes no son administrativos generalmente.



# Capacitación

- Si en lugar de defender atacas
- Si por un día tu trabajo es romper



# Terapia de Choque

- Si te muestran un ejemplo vs tu código
- Experto vs Compañero
- Romper vs Explotar



# Cajas Mágicas

- Son solo eso . . . cajas
- Defienden de lo que saben defender
- Al final necesitas un humano
- Y lo demás que?



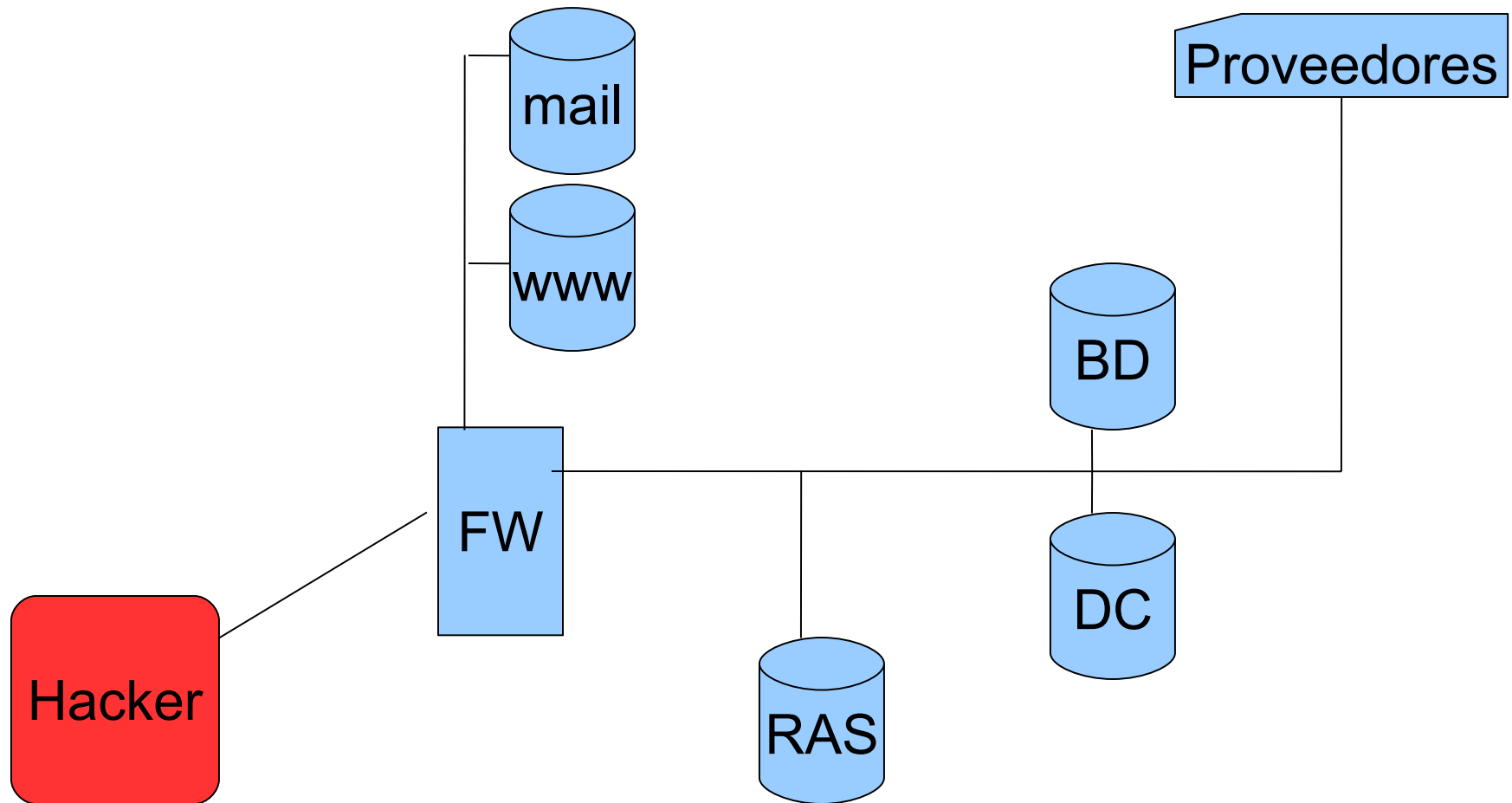
# Soluciones Mágicas

- Dependen de las cajas
- Soluciones tan inteligentes que son . . . . no inteligentes
- La interface psiquica TODAVIA no esta tan desarrollada como deberia



# Soluciones Mágicas

- Dependenden como las instalaes
- Si no sabes donde poner ..  
mucho no pueden ayudar





# La mejor defensa

- La mejor defensa es el ataque
- Seguridad relativa a tus vecinos
- I+D
- I+D
- I+D



Centro  
**Banamex**  
CIUDAD DE MÉXICO

27-29, Febrero 2008

Open source is everywhere



Preguntas?

[enrique.sanchez@yaguarete.com.mx](mailto:enrique.sanchez@yaguarete.com.mx)

[www.0hday.org](http://www.0hday.org)