

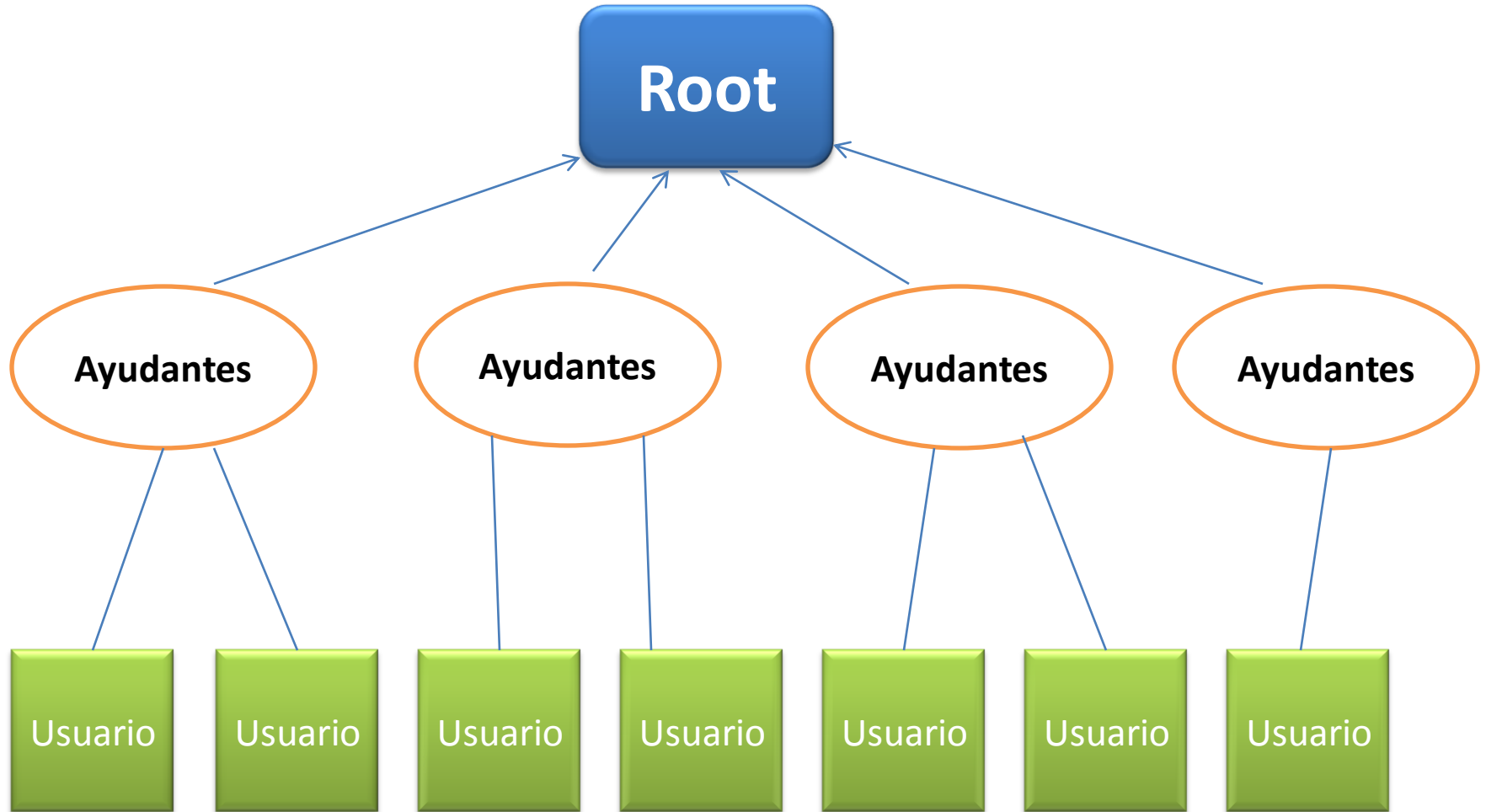


Un laboratorio bajo lupa

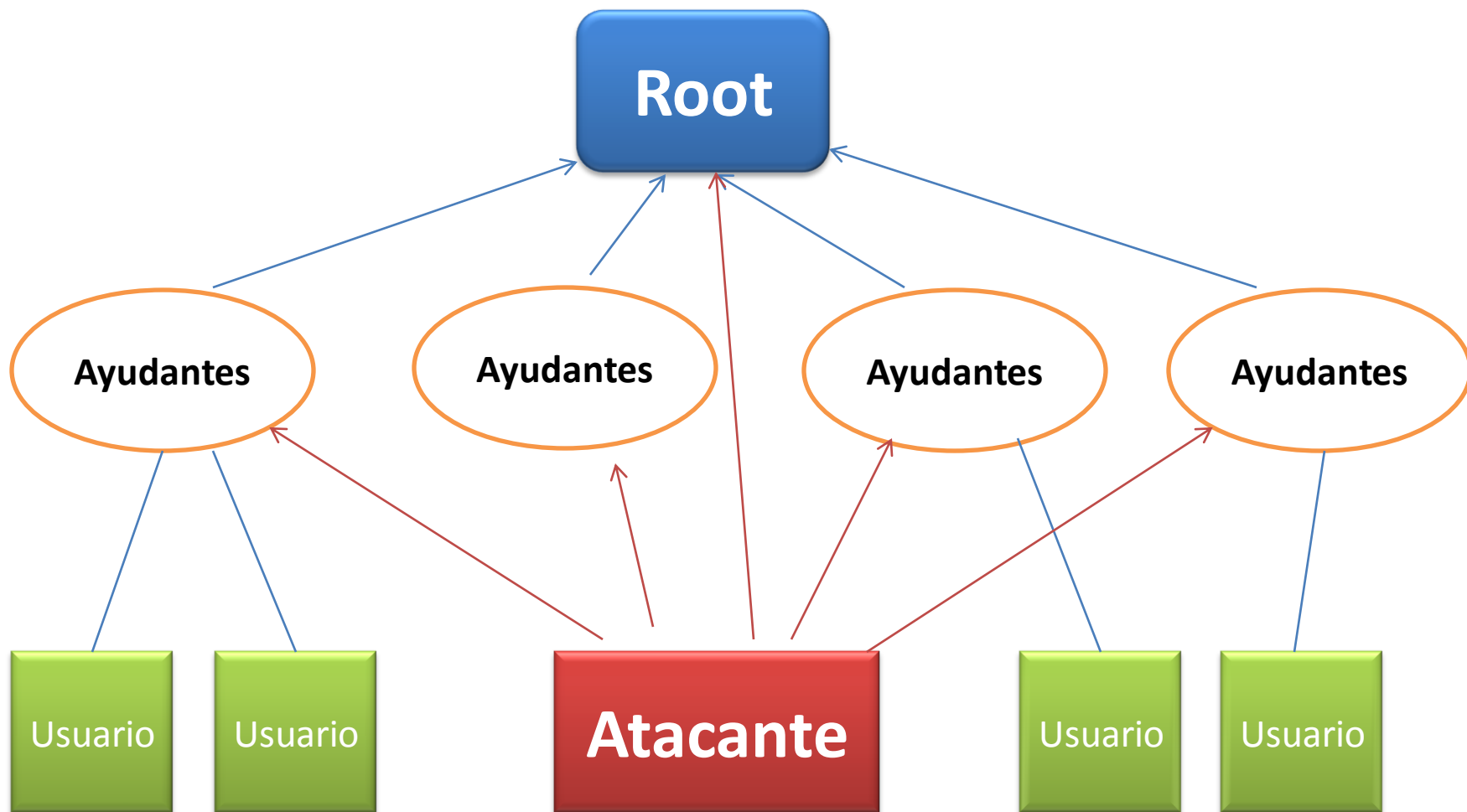
Claudio Salazar

Security Research Team

Una estructura común



La OTRA estructura común



Vamos por las cookies

- **Root ?**
 - Local kernel exploit
 - Local exploit sobre binarios suid
 - Exploit sobre proceso corriendo como root (remoto/local).
 - Otras posibilidades que veremos más adelante.

Pero hay un objetivo más fácil ..

Vamos por las cookies

Ayudantes

- Son **N, N** posibilidades de lograr mayores privilegios en la topología del lab.
- Si se renuevan constantemente, todos los novatos cometen los mismos errores.
- Un poco de ingeniería social ..

Vamos por ellos

Atacante : Hey Ayudante, se me quedo pegado X.

Ayudante : Oka

Ayudante loguea como root en tty1

Ayudante /etc/init.d/gdm restart

Atacante lo distrae, le pregunta porque en su sesión gráfica se le ve la letra así, ese ícono se muestra así, bla bla o a veces no existe dialogo.

Ayudante se va ..

Atacante cambia a tty1 : ROOT !

Vamos por ellos

- Estas cosas pasan.
- A veces no es necesario que sea en mi maquina :)
 - Una lista de los nodos, más un pequeño script.
 - Revisando nodos en busca de la cookie.
 - Tenemos la galleta y nos cambiamos de maquina :D

```
USER  TTY  FROM          LOGIN@  IDLE  JCPU  PCPU
root  tty1  -             01:07  7.00s 0.27s 0.01s -bash
```

- Pero que este logueado en un nodo, sin saber la pass, de que le sirve ?

Vamos por ellos

```
cp /bin/login /bin/login2
```

```
cp /tmp/mylogin /bin/login
```

```
if (user == "root" || user == "ayudante")  
    enviar(pass);  
/bin/login2 (user, pass)
```

A esperar que llegue la pass y bingo !

- Esto es lo más sencillo con lo que nos podemos encontrar.

No es lo único ..

- Cuidado con los permisos de directorios donde se maneja información sensible del laboratorio.
- No guardar las pass del lab en programas.
- Especial cuidado con cualquier fichero sensible (.bash_history, .mysql_history, .ssh/id_rsa, etc)

De visita a la casa ..

- Nos encontramos con un ayudante logueado desde su casa.
- Identificamos su sistema operativo.
- Linux? Si no es fuerza bruta a SSH, o revisar las apps web que pueda estar sirviendo, difícil conseguir algo ..
- Pero esta el otro lado de la moneda, Windows.

De visita a la casa ..

- Windows pirata no tiene acceso a updates de seguridad.
- Buscamos remote exploit para Windows +/- recientes (existen hartos) y probamos suerte.
- Boom! Es nuestro.

Y si de sofisticación se habla ..

From : atacante@inf.utfsm.cl

To : seguridad@labcomp.inf.utfsm.cl

Msg : Hey! Ha salido una nueva vulnerabilidad en el demonio X, se publico este exploit adjunto y lo probe en tal maquina y funciona. Parchen a la brevedad.

Reply : Muchas gracias, revisaremos el asunto.

Y si de sofisticación se habla ..

Seguridad revisa el código del exploit y lo compila. Efectivamente, al ejecutar el exploit consigue explotar exitosamente la vulnerabilidad del demonio X y entrega una shell, pero que más?

- El exploit puede cumplir su papel y además agregar una puerta trasera, enviar información, etc, a través del ocultamiento de estas acciones dentro de la shellcode.

Y si de sofisticación se habla ..

- Una shellcode :)

```
char shellcode[] =  
    "\x31\xc0\x31\xdb\xb0\x17xcd\x80\x31\xc0\x50\x68\  
x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\  
3\x89\xe1\x99\xb0\x0bxcd\x80";
```

- Que hace ?

```
n1c@spect:/tmp$ ndisasm -b 32 shellcode  
00000000  31C0          xor  eax,eax  
00000002  31DB          xor  ebx,ebx  
00000004  B017          mov  al,0x17  
00000006  CD80          int  0x80  
  
.....  ...  .  ...  
...  ...  ...
```

Las otras amenazas



Apache/PHP

- Usuarios con `public_html`
- `safe_mode` on ? No del todo seguro.
- Cada cierto tiempo, se encuentran formas de esquivarlo
- También existen `buffer overflows` en PHP
- Lo principal no es prohibir, sino estar atentos.

Apache/PHP

- Ya pasando a apps del laboratorio, revisar continuamente el bugtracker para anticiparse a posibles vulnerabilidades.
- A veces pueden ser solucionadas sigilosamente, por lo tanto en las nuevas versiones del software revisar el ChangeLog en busca de parches a problemas de seguridad.

MySQL/Postgresql

- En caso de una inyección SQL explotada satisfactoriamente, una configuración de los permisos de usuario correcta aminora el impacto.
- Si sólo se realizan conexiones locales, bloquear las conexiones remotas.

CUPS

- Un usuario me pide que le imprima un doc.
- Ingreso mi user/pass y se lo imprimo.
- Que pasa si alguien estaba escuchando de por medio, o en la misma maquina ?
- Uso de CUPS con SSL.

Cosas de la vida

- Intentar no tener servicios ejecutándose como root.
- Un Servidor Web (Apache/Tomcat/etc) como root, si existe LFI se pueden leer archivos como el /etc/shadow. Ejecución remota de comandos aún peor.
- Un motor DB como root, lectura/escritura de archivos importantes. Si es LAMP aún peor. Sino, técnicas sutiles i.e. la utilizada para el deface de apache.org (2000)

Los otros servicios

- Restringir permisos para copia de repositorios.
- Configurar permisos para cada trac.
- Una ayudita a Apache, un WAF (Web Application Firewall)
- Una ayuda al nodo, un IDS (Intrusion Detection System)
- Una ayuda al kernel, parches grsecurity o SELinux.

A close-up portrait of an elderly man with dark skin, wearing glasses and having a grey beard. He is looking slightly to the right with a thoughtful expression. The background is a blurred green, suggesting an outdoor setting.

La misión

La misión

- Como ayudantes, la misión es prestar servicios a los usuarios.
- Y lo más recurrente en el último tiempo, velar por la privacidad e integridad de su información.
- Esto refiere tanto a los datos personales como los documentos que manejan.

El antídoto

- Concientizar que seguridad no es tener corriendo unos cuantos programas creados para eso.
- Es educar a los ayudantes de la responsabilidad que tienen al cuidar los datos de los demás.
- Junto con medidas globales para evitar fugas de información.

Q&A

