Protect Your System Against Shellshock

This informative article explains how the newly discovered Shellshock bug operates. It gives simple, easy-to-implement methods to reduce your system's vulnerability to it as well as minimise the harmful effects.

Shellshock is a security bug, which allows the attacker to execute arbitrary commands in the UNIX bash shell, a command line interface commonly used in UNIX and Linux distributions. This bug was first discovered by Stéphane Chazelas on September 12, 2014, and he initially called it 'bashdoor'. It was assigned the CVE (common vulnerability and exposures) identifier 'CVE-2014-6271', and publicly disclosed on September 24, 2014. This article will discuss the effects of the bug, ways to test your system's vulnerability to it and also explore ways to protect yourself against attackers. Similar vulnerabilities are being discovered continuously and patches are being updated regularly.

Test your system

To test if your system is vulnerable to the Shellshock bug or not, open up a terminal (*Terminal.App* for Mac OS X) and type the following command:

```
env x='() { :;}; echo vulnerable' bash -c 'echo hello'
env x='() { :;}; echo vulnerable' sh -c 'echo hello'
```

Here, we are setting an environment variable 'x'. The part *echo vulnerable* is the arbitrary command that is being executed before the actual bash command, i.e., *bash -c 'actual command'*, hence giving the attacker a chance to run arbitrary commands on your system.

You can further test your system for the vulnerability (as reported by Tavis Ormandy - CVE-2014-7169) by running the following command:

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

You will notice that the date is displayed and you can also find a file named 'echo' in the current directory. This means that your system is still vulnerable.

Here is another test that you can run, which will print out *'not patched'* if your system is not yet patched or is still vulnerable to the bug:

```
env foo='() { echo not patched; }' bash -c foo
```

The widespread effects of Shellshock

The Shellshock bug potentially affects all the systems that

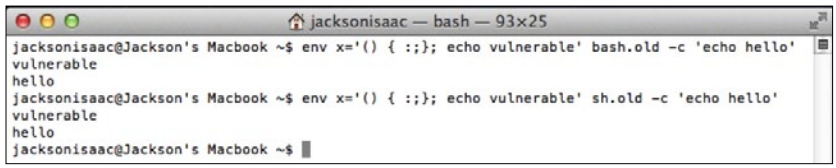

Figure 1: The Shellshock vulnerability test

have bash running on them. Bash is installed as the system's default command line interface on many Linux and UNIX-based systems including Mac OS X. Based on the source code analysis of bash, about *25 years' worth of bash versions* since the early 1990s through version 4.3 (in Linux systems) and version 3.2.48 (in Mac OS X) seem to be vulnerable. The vulnerability might exist beyond these versions as well, based on recent reports about the patches not being entirely effective.

It would be very difficult to patch all the systems that are affected by this bug, as bash is used in almost all devices like modems, home routers, Internet of Things (IoT) with embedded Linux, Web servers and devices connected to the Internet. Although bash is not directly exposed to the Internet, software or an application that is connected to the Internet can be used to run commands through bash internally on the vulnerable systems.

Operating systems that bash supports allow environment variables to be set, which are dynamic in nature. The attacker can attach some malicious code to this environment variable that will be executed once the variable is received. The attacker can force an application to send this specially crafted environment variable to bash, and it could be used to create a self-replicating worm.

According to a survey by Netcraft, we can conclude that almost half the online servers are running on Apache, which in turn runs on Linux machines and would have bash installed. As a result of this, half the Internet is vulnerable to the Shellshock bug.

The most likely method to attack systems is through Web servers using CGI (Common Gateway Interface), which is widely used to generate dynamic Web content. CGI scripts can be used to execute bash commands without the need for any authentication.

The bash continues processing commands after the function definition, resulting in what is generally termed as 'code injection attack'. The problem is that the auto-import function parser runs past the end of the function definition and keeps executing the codes. An attacker can gain access through this, and can then compromise and infect other systems on the network. FreeBSD and NetBSD have disabled auto import functions in bash version 3.2.54 onwards, by default, to prevent future vulnerabilities.

According to Wikipedia, the security firm Incapsula noted 17,400 attacks on more than 1,800 Web domains, originating from 400 unique IP addresses on September 26, 2014; 55 per cent of the attacks originated from China and the United States. By September 30, the website performance firm CloudFlare said it was tracking approximately 1.5 million attacks and probes per day related to the Shellshock bug.

Updating bash

In order to reduce your system's vulnerability to Shellshock, you can update bash in the OS you are using. Here's how you can do so.

Build bash from source on Mac OS X

Here I'll show you how to rebuild bash from source on Mac OS X, as the patch released by Apple Inc is not entirely effective against some of the vulnerabilities.

Note: Make sure your Mac has *Xcode* and *Xcode command line* tools installed.

Open up *Terminal.App* and type in the following commands.

(i) To download the bash tarball and apply patches:

```
$ curl https://opensource.apple.com/tarballs/bash/bash-92.
tar.gz | tar zxf -

$ cd bash-92/bash-3.2

$ curl https://ftp.gnu.org/pub/gnu/bash/bash-3.2-patches/
bash32-052 | patch -p0

$ curl https://ftp.gnu.org/pub/gnu/bash/bash-3.2-patches/
bash32-053 | patch -p0

$ curl https://ftp.gnu.org/pub/gnu/bash/bash-3.2-patches/
bash32-054 | patch -p0

$ curl https://ftp.gnu.org/pub/gnu/bash/bash-3.2-patches/
bash32-055 | patch -p0

$ curl https://ftp.gnu.org/pub/gnu/bash/bash-3.2-patches/
bash32-056 | patch -p0

$ curl https://ftp.gnu.org/pub/gnu/bash/bash-3.2-patches/
bash32-057 | patch -p0
```

Here, we need to add patches starting from 052 and apply the later patches subsequently.

Note: Bash 3.2 patches 52, 53, and 54 correspond to Bash 4.3 patches 25, 26, 27 and so on.

(ii) Rebuild bash: After applying the patches in the bash-3.2 folder, we move up to the bash-92 folder and build the source:

```
$ cd ..
$ xcodebuild
```

(iii) Back up and update default bash: Now let us check the

version of the bash we have built, take a back-up of the old bash, and replace the latter with the built version of bash.

```
$ build/Release/bash –version

$ build/Release/sh --version

$ sudo cp /bin/bash /bin/bash.backup

$ sudo cp /bin/bash /bin/sh.backup

$ sudo cp build/Release/bash /bin

$ sudo cp build/Release/sh /bin
```

You can also add a *chmod a-x* to the backed-up version to prevent it from being used, instead of the newer version:

```
$ sudo chmod a-x /bin/bash.backup

$ sudo chmod a-x /bin/sh.backup
```

Update bash on Debian-based systems (Ubuntu, etc)

Ubuntu/Debian users can easily update bash through the official repositories with the help of 'apt-get':

```
$ sudo apt-get update

$ sudo apt-get install bash
```

You can also run the following command in a terminal:

```
$ sudo apt-get –only-upgrade install bash
```

Update bash on Fedora systems

Type the following command to update bash on Fedora systems:

```
$ sudo yum update bash
```

References

[1] *http://en.wikipedia.org/wiki/Shellshock_(software_bug)*
[2] *http://apple.stackexchange.com/questions/146849/how-do-i-recompile-bash-to-avoid-shellshock-the-remote-exploit-cve-2014-6271-an*
[3] *http://www.lynda.com/articles/shellshock-bash-exploit*
[4] *http://www.troyhunt.com/2014/09/everything-you-need-to-know-about.html*
[5] *http://www.symantec.com/connect/blogs/shellshock-all-you-need-know-about-bash-bug-vulnerability*
[6] *http://news.netcraft.com/archives/2014/09/24/september-2014-web-server-survey.html*

By: Jackson Isaac

The author is an active open source contributor to projects like GNOME-Music, Mozilla Firefox and Mozillians. You can follow him on *jacksonisaac.wordpress.com* or reach him by mail at *jacksonisaac2008@gmail.com*