the Availability Digest

www.availabilitydigest.com
@availabilitydig

Shellshock – The Bash Vulnerability

October 2014

A bug in the popular Bash shell used in Linux and Unix operating systems was discovered at the end of September, 2014. Dubbed 'Shellshock,' the flaw has led to a code-injection vulnerability for applications running under Linux and Unix that use Bash as the shell. Some analysts are concerned that Shellshock may rival the recent Heartbleed vulnerability, discovered last April in OpenSSL.[1]

The Bash Vulnerability

Bash is a commonly used command interpreter for Unix and Linux systems. It converts typed or scripted command strings into operating system commands. Bash is a free software replacement for the Bourne shell.[2] Bash was released in 1989, and it has been distributed widely as the shell for the GNU operating system and as a default shell for Linux and Apple's Mac OS X.

Bash uses Common Gateway Interface (CGI) scripts to generate dynamic content for web pages and web applications. The Bash vulnerability is caused by its handling of environmental variables in CGI scripts. When a script assigns a CGI variable, not only is the specified function executed, but any malicious trailing code in the function definition also will be executed. Bash takes no precautions to protect against trailing code.

This opens the door for code-injection attacks. It is fairly easy to smuggle code into CGI environment variables, as the CGI scripts are sent over the Internet to display web pages on client systems. The malicious code can open a back door to welcome sophisticated malware that then can infect an enterprise's entire network. CGI scripts, including child processes spawned by the scripts, are all vulnerable.

The Bash vulnerability was discovered on September 24, 2014, by Stéphane Chazelas, a French IT manager working for a software maker in Scotland. However, the flaw has been around for twenty-two years. It first appeared in Bash Version 1.13 and continues through Bash Version 4.3.

Not only is Shellshock a critical security risk to Linux and Unix operating systems, it also is a threat to the entire Internet because of Bash's ubiquity among systems connected to the Internet. Even worse is the "Internet of Things," in which all kinds of controllable devices are attached to the Internet. Linux is now being used in cars, calculators, intelligent light bulbs and toasters, and even home wireless routers. It is doubtless that most intelligent appliances will ever be patched to correct the Bash defect. The Internet of Things opens the entire Internet to attack by Shellshock.

[1] Heartbleed – The Worst Vulnerability Ever, *Availability Digest*; April 2014.
http://www.availabilitydigest.com/public_articles/0904/heartbleed.pdf

[2] Bash' is an acronym standing for *Bourne-again shell*, referring to its objective as a free replacement for the Bourne shell.

NIST (the National Institute of Science and Technology) rates the Bash flaw 10 out of 10 in terms of severity. Nist has stated:

> GNU Bash through 4.3 processes trailing strings after function definitions in the values of environment variables, which allows remote attackers to execute arbitrary code via a crafted environment, as demonstrated by vectors involving the ForceCommand feature in OpenSSH sshd, the mod_cgi and mod_cgid modules in the Apache HTTP Server, scripts executed by unspecified DHCP clients, and other situations in which setting the environment occurs across a privilege boundary from Bash execution.
>
> Authentication: Not required to exploit.
>
> Impact Type: Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service.

Comparison to Heartbleed

Many analysts have compared the seriousness of Shellshock to Heartbleed. In fact, the similarities between them are striking. While Shellshock allows execution of malicious commands that have been inserted following an environmental variable in a CGI script, Heartbleed allows the reading of unauthorized data following a text string.

Heartbleed is a flaw in the OpenSSL (Open Secure Sockets Layer) cryptographic software library. The OpenSSL flaw that created the Heartbleed vulnerability was a missing-bounds check in handling heartbeats. Heartbeats were added to OpenSSL in Version 1.0.1, which was released on March 14, 2012. The intent of the heartbeat was to test and keep alive secure communication links without the need to renegotiate the connection.

Either side can send a heartbeat to the other side, and the initiator expects a copy of its heartbeat in response. The heartbeat is an arbitrary text string, up to 64K bytes in length, and a 16-bit integer indicating the length of the text string. For instance, the heartbeat message might be “bird, 4.”

Unfortunately, the flawed heartbeat logic in OpenSSL did not check the text message against the indicated length (the missing bounds check). If the heartbeat initiator sent a heartbeat comprising “bird, 64K,” it would receive the text “bird” followed by 64K bytes (less four bytes) of the following contents of memory.

That chunk of memory can then be mined for information nuggets. Typically, it is dynamically allocated memory used by OpenSSL and contains data such as private keys, user names, and passwords. Since the memory block returned as a heartbeat response is taken from dynamically allocated memory that has been returned to the memory pool, each heartbeat usually receives a different block of old memory that contains different information.

Though the Heartbeat flaw was introduced in 2012, it was not discovered until April, 2014.

Operating System Patches

There has been a rapid response to harden operating systems against Shellshock. Red Hat has released patches for its versions of Linux and Unix, and Apple has released patches for OS X. Oracle has released patches for Oracle Linux Versions 4 through 7 and for Solaris Versions 8 through 11. Apache web servers are at particular risk and should be carefully evaluated.

The GNU project has issued a series of patches for Bash. The Free Software Foundation has issued a call for financial help to improve Bash.

Oracle Vulnerabilities

Oracle has confirmed that thirty-two of its products are affected by Shellshock. They include some costly integrated hardware systems. Oracle is currently working on fixes, but it has given no schedule as to when the patches will be ready.

Some of its infected hardware products include:

Oracle Exalogic cluster appliances
Oracle Database Appliance
Big Data Appliance
SPARC SuperCluster
SUN ZFS Storage Appliance Kit

Affected software products include:

Oracle VM
Oracle Key Vault
Many Oracle communication products

Mitigating Shellshock

According to blogger Hernan Ochoa, you can check if you're vulnerable by running the following lines in your default shell, which on many systems will be Bash. If you see the words "busted," then you're at risk. If not, then either your Bash is fixed or your shell is using another interpreter.

```
env X="() { :;} ; echo busted" /bin/sh -c "echo completed"

env X="() { :;} ; echo busted" `which bash` -c "echo completed"
```

If you are vulnerable, you have several options:

- You can upgrade to a new, patched version of Bash.
- You can replace Bash with an alternative shell.
- You can limit access to vulnerable services until you have the vulnerability corrected.
- You can filter inputs to vulnerable services to ensure that there is no attached malicious code.

Summary

Shellshock is still too new to know if it will live up to analysts' fears. However, there already is proof-of-concept code on the Internet for exploiting Bash using CGI scripts to run code with the same privileges as the web server.

Acknowledgements

Material for this article was taken from the following sources:

Patch Bash NOW: 'Shellshock' bug blasts OS X, Linux systems wide open, *The Register*; September 24, 2014.
Environment Bashing, *Akamai*; September 24, 2014.
'Bash' bug could let hackers attack through a light bulb, *CNN*; September 25, 2014.

Oracle SHELLSHOCKER – data titan lists unpatchables, *The Register*; September 27, 2014.
SMASH the Bash bug! Red Hat, Apple scramble for patch batches, *The Register*; September 28, 2014.