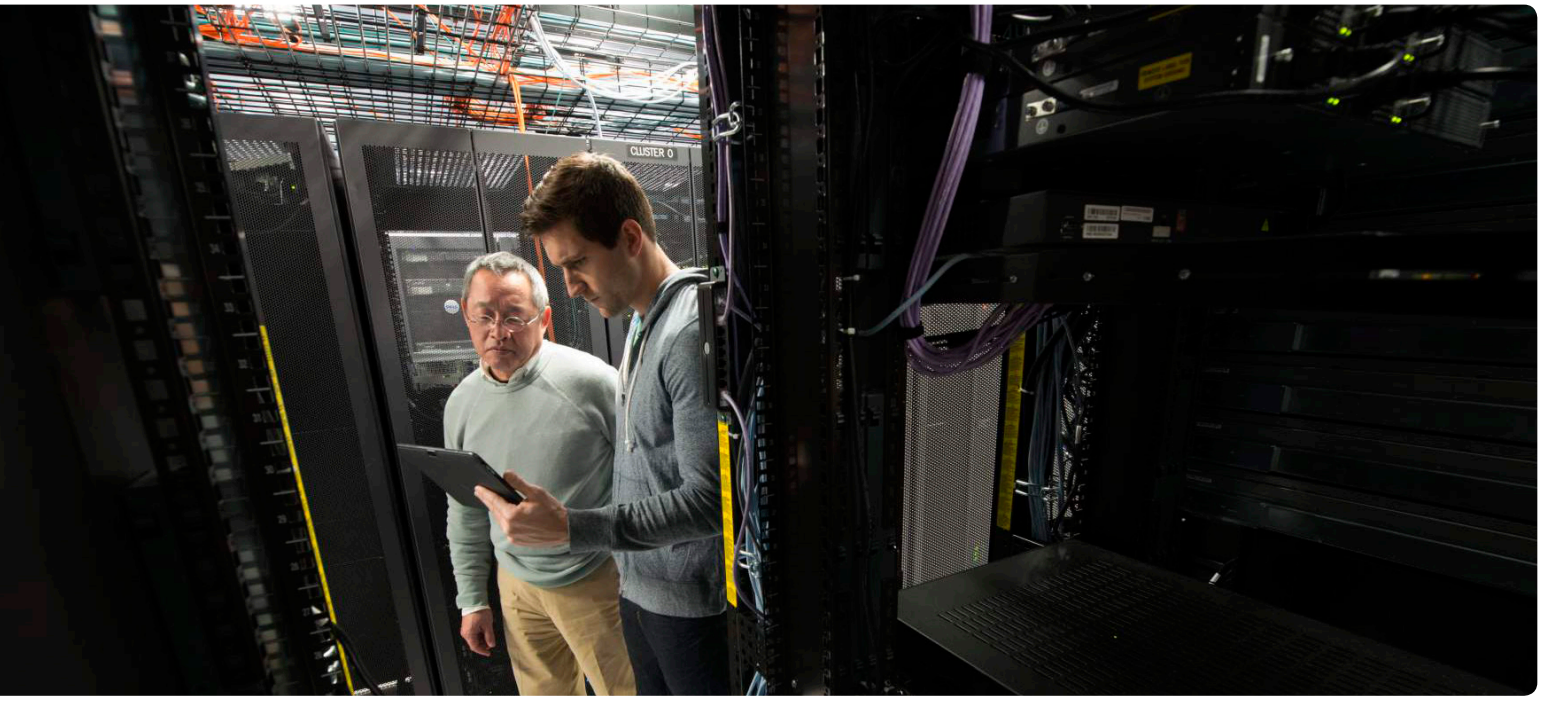
Shellshock and Future Vulnerabilities—Buying Precious Time



Introduction

The GNU Bash Code Injection Vulnerability, or Shellshock, affects Linux/UNIX systems by permitting execution of arbitrary commands in the context of Bash shell. Shellshock is relatively easy to exploit yet very high in severity, particularly through HTTP.

The best protection against Shellshock and similar vulnerabilities is a next-generation firewall (NGFW) (such as from Dell SonicWALL) that does not run Linux or UNIX. The signatures (such as the ones addressing Shellshock) immediately update NGFWs with Intrusion Prevention Service (IPS) to prevent sophisticated exploits early in the kill chain while buying enterprises valuable time to properly review and execute their software patching and update plans.

Shellshock was not the first widespread network vulnerability and it will not be the last. As an example of the Dell SonicWALL approach, this white paper provides IT managers and network administrators with background and insights about similar vulnerabilities:

- Why vulnerabilities like Shellshock are so dangerous
- How Shellshock works
- How attackers exploit network vulnerabilities like Shellshock
- How to buy time and protect networks from exploits
- What to do about future vulnerabilities

Unpatched Bash is vulnerable to arbitrary command execution because it can process strings that follow function definitions in environment variables.

Background

The GNU Bash Code Injection Vulnerability, or Shellshock, affects Bash shell, the command-line user interface in Linux, UNIX and their variants. (These variants include OS X, the operating system running on all Apple Macintosh computers sold today, because of its origins in the BSD distribution of UNIX.) All unpatched versions of Bash between 1.03 and 4.3 are vulnerable.

The innocent code behind Shellshock was first written in 1989 and lurked as a vulnerability in the Bash shell for a quarter-century. It was discovered and publicly disclosed in September 2014, and cited in six Common Vulnerabilities and Exposures (CVE) entries in the National Vulnerability Database maintained by the National Institute of Standards and Technology (NIST).[1]

Unpatched Bash is vulnerable to arbitrary command execution because it can process strings that follow function definitions in environment variables. Some of the known ways to exploit this vulnerability are through common, application-level protocols like HTTP, SSH, DHCP and SMTP.

NIST rates Shellshock 10 out of 10 in severity, with low access complexity and no need for authentication.[2] If exploited, the vulnerability may allow unauthorized disclosure of information, unauthorized modification and disruption of service.

Shellshock is typical of network vulnerabilities in its effects on networks and IT teams. Its severity and the urgent steps required to deal with it offer a cautionary tale to network administrators preparing for vulnerabilities they will encounter in the future.

Why vulnerabilities like Shellshock are so dangerous

Shellshock and similar vulnerabilities are dangerous for several reasons:

- They arise from an innocent oversight in legitimate code.
- That code is present and running on millions of systems and devices – many of which are long forgotten or do not receive regular updates – on networks worldwide.
- Tens of thousands of bad actors around the world know the code well and can launch exploits against it in almost no time.
- It is relatively easy to access the code and exploit it, even from outside the network, because no authentication is required.
- The network protocols most vulnerable to exploitation are among the most common, carrying billions of packets per hour.

As an analogy for Shellshock, consider a trusted brand of subway turnstile in use at most of the world's busiest subway stations. For 25 years, the manufacturer, the repairmen, the subway authorities and the passengers have assumed that it permits only one passenger at a time to go through.

One day an installer discovers that any number of passengers can actually get through the turnstile at one time by crowding together. Worse yet, by crowding like this, criminals can potentially smuggle in weapons and contraband undetected. Once inside, they can use the weapons to hijack subway trains and smuggle larger quantities of contraband onto bigger transportation networks.

The installer announces the vulnerability and thieves everywhere start trying to exploit it. It will take every subway system a few days or weeks to dispatch a repairman, but in the meantime, it is not an option simply to block off the

[1]CVE-2014-6271, -6277, -6278, -7169, -7186 and -7187.
[2]"Vulnerability Summary for CVE-2014-6271", National Institute of Standards and Technology, last revised November 19, 2014.

turnstiles because all passenger traffic would grind to a halt. While subway authorities figure out and implement a fix, they post armed guards at each turnstile and throughout the system to ensure that any suspicious activity is promptly stopped.

No analogy is perfect, but in almost every similar situation, a normally trusted component becomes vulnerable. Bad actors then find ways to exploit the vulnerability rapidly and affect nearby systems. And, fortunately, NGFWs (such as SonicWALL) that are not subject to the vulnerability serve as armed guards to buy time until organizations can patch and update their systems.

How Shellshock works

The details of the Shellshock vulnerability are outlined in CVE-2014-6271.

Background—Environment variables and Bash shell

Bash shell supports global variables (or environment variables), which can be seen and accessed by the processes that create them and by child processes of the current shell. Programmers can use the export keyword to create any of these variables. For example, in

```
export varglobal ="value2" OR export varlocal ="value1";
```

"value2" is an environment variable.

Besides taking these environment variables, Bash also supports exporting functions (i.e., globally defining functions). Any environment variable definition starting with the characters () { is used to export functions.

Function exporting definition

The Shellshock vulnerability comes into play when Bash shell parses the logic of that function exporting definition. When Bash shell executes a command, it invokes a function called initialize_shell_variables. A for-loop parses the command given to Bash shell, and a conditional statement in that for-loop checks for function export definition; specifically, whether () { is present.

Inside that conditional statement is another function call, parse_and_execute, which starts executing commands in the context of Bash shell. The problem is that parse_and_execute does not check to see whether anything besides the function definition is being passed. As long as that is the case, it is possible to append arbitrary commands along with the () { text and have Bash shell execute them.

Arbitrary commands could include unintended, malicious commands that move files, modify passwords and delete everything on the computer. That is the main vulnerability at the heart of Shellshock.

How attackers exploit network vulnerabilities like Shellshock

Network vulnerabilities affect several common networking protocols (exploit protocol vectors). Shellshock, for instance, is much more exploitable and carries more severe consequences for some protocols than for others.

HTTP

HTTP, the protocol used in transferring Web pages, is an exceptionally effective way to exploit a network vulnerability because of its prevalence on the Internet.

In the case of Shellshock, the main vulnerability lies with HTTP servers running some kind of dynamic parsing script in Linux environments. Shellshock takes advantage of any application services trying to parse or process values of HTTP Client request headers and execute commands in the context of Bash shell. For example, it could be used to replace values of common HTTP headers, such as:

```
User-Agent: () { :;}; <command to execute>
```

Passing those variables and values into a vulnerable version of Bash shell will

NGFWs (such as SonicWALL) that do not run on Linux/UNIX are not vulnerable to Shellshock or similar vulnerabilities.

The intrusion prevention system can buy precious time for IT teams to respond to widespread network vulnerability strategically instead of just tactically.

cause the destination computer to execute any command in angle brackets.

SSH

Exploiting SSH (Secure Socket Shell) is far less effective than exploiting HTTP for two reasons:

1. An attacker would need valid authentication credentials to try this vector.
2. Most common SSH server deployments already grant authenticated users permission to execute any arbitrary command they want anyway.

For example, an attack could take this form:

```
ssh –l someuser <remote-host> '() { :;}; <command to execute>'
```

Subject to the limitations above, Bash shell would cause any command in angle brackets to be executed.

Since Shellshock does not grant credentials and allows the execution of arbitrary commands only in the context of Bash shell, a Shellshock-based attack on SSH servers is less worrisome. The vulnerability depends on the user's shell access privileges, which can be managed through account restrictions, and it does not entail "user level privilege" escalation.

DHCP

In this exploit protocol vector, a client sends a DHCP Discover message requesting a lease from the DHCP server. The server replies with a DHCP Offer message, which is an opportunity to exploit the client. First, however, it is necessary to compromise the DHCP server, which is difficult to accomplish externally, and only slightly less difficult internally.

In the case of Shellshock, an attacker could send additional, optional headers in the Offer message, replacing values in those headers with Bash export function definitions followed by a command to execute. If a Linux/UNIX client started processing the headers, it could execute arbitrary commands; for example:

```
add "() { :;}; <command to execute>"
```

It is not easy to gain access to and compromise a DHCP server from the outside, so a successful exploit would likely be at least partly internal. But this could be an effective attack in an environment with many Linux clients.

SMTP

This exploit adds values to certain SMTP headers to get the server to execute specific commands in the context of Bash shell; for example, by appending

```
() { :;}; <command to execute>
```

to SMTP headers like MAIL FROM.

Vulnerability has been demonstrated in the Qmail server, and MIME headers may represent another area of vulnerability, depending on how the email system was customized. If somebody is processing MIME headers on a Linux/UNIX system and executing commands in the context of Bash shell, then that process could be vulnerable as well.

Use cases include a botnet attempting to exploit SMTP servers by attempting to download additional components to the email server. Upon success, the system initiates an IRC channel to a command-and-control server and listens for commands.

How to buy time and protect networks from exploits

The most effective way to prevent exploits from wreaking havoc on networks is also the most accessible: enable the intrusion prevention system (IPS) on security gateways and ensure that the IPS signatures are always up to date. The IPS can buy precious time for IT teams to respond to widespread network vulnerability strategically instead of just tactically.

Another common countermeasure is to install anti-virus software on the server. Once compromised, many computers promptly download and

run malware, so anti-virus products can detect and eliminate such programs before they have the chance to do harm or propagate.

As for Shellshock, IT managers and network administrators can identify potentially vulnerable systems—primarily Linux servers—on the network with one simple command line test:

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

If the output contains the word "vulnerable," then the system is vulnerable.

Since all Linux/UNIX distributions offer updates with patches for Shellshock, it is important to run them immediately on all systems, vulnerable or not. For example:

- for CentOS, Red Hat and Fedora, `yum update bash`
- for Ubuntu and Debian, `apt-get update && apt-get install --only-upgrade bash`

What to do about future vulnerabilities

Within a few hours of the public disclosure of Shellshock there were reports of several related exploits:

- Malware droppers downloaded additional components to vulnerable servers and launched them.
- Reverse shells and back doors opened specific ports to access vulnerable servers from outside.
- Vulnerable computers launched data exfiltration attempts to steal data and upload it to foreign servers.
- Compromised machines formed distributed denial of service (DDoS) botnets that attacked or caused attacks on other machines.

Enterprises with hundreds or thousands of servers have the greatest exposure to network vulnerabilities like Shellshock, yet they also take the most time to manage configuration changes and roll out updates to affected systems. Therefore, they must rely on NGFWs and network security gateways as their first line of defense.

As is typical for a vulnerability of this nature, Dell SonicWALL has deployed multiple IPS signatures against Shellshock (see table below).

These signatures, which were available within hours of the public announcement of each vulnerability, run on Dell SonicWALL NGFWs to protect networks from exploits. That protection bought IT teams the time they needed to plan and implement their own strategies for updating individual servers.

Enterprises with thousands of servers have the greatest exposure to network vulnerabilities like Shellshock, yet they also take the most time to roll out updates to affected systems.

IPS Signature	Corresponding CVE	Signature Number
5665	CVE-2014-6271	1
10529	CVE-2014-6271	1
5603	CVE-2014-6271	2
5605	CVE-2014-6271	3
5667	CVE-2014-6277, CVE-2014-7186	1
5661	CVE-2014-6278, CVE-2014-7169	1
5669	CVE-2014-7187	1

Well over a month after disclosure, bad actors were still launching millions of attempts per day to exploit Shellshock.

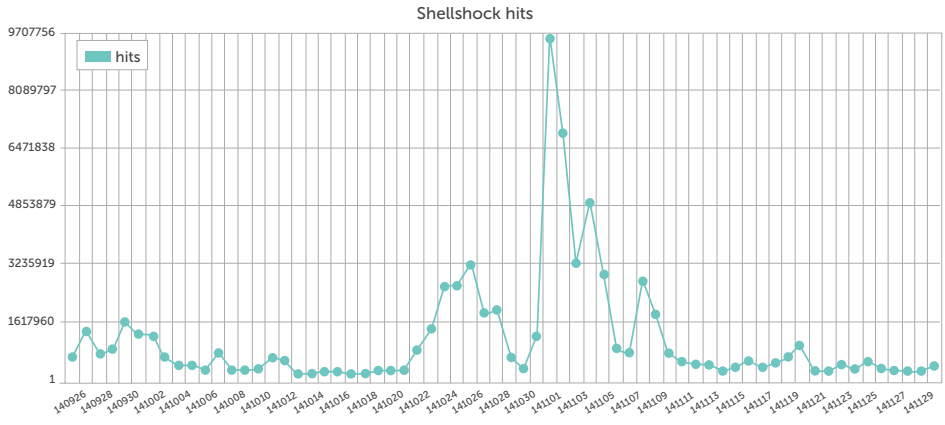


Figure 1: Shellshock attacks attempted against networks protected by Dell SonicWALL

Furthermore, attempts to exploit network vulnerabilities do not simply cease after a few days. Figure 1 is a chart of data collected by the IPS module on Dell SonicWALL NGFWs installed all over the world.[3] Peaks in malicious activity occurred in the first few days after the vulnerability was disclosed, but the millions of attempts per day in late October and early November — well over a month after the disclosure — underscore the wisdom of protecting and patching rather than trying to ride out the initial storm.

NGFWs that do not run on Linux/UNIX (including Dell SonicWALL) are not vulnerable to this or future vulnerabilities of this type.

Conclusion

The Shellshock vulnerability affects Linux/UNIX systems by allowing attackers to execute arbitrary commands appended to function definitions of environment variables in the Bash shell context. The most commonly exploited protocol vectors are HTTP, SSH, DHCP and SMTP. Dell SonicWALL NGFWs are not vulnerable because they do not run Linux or UNIX; however, machines on the same network that have not yet been patched are still vulnerable.

IT managers and network administrators can protect themselves against Shellshock and future network vulnerabilities by installing updates for all their Linux/UNIX distributions and installing anti-virus software on their servers. In the meantime, Dell SonicWALL continuously blocks exploits and buys precious time.

To learn more about Dell SonicWALL and its product offerings, please visit dell.com/sonicwall.

[3]The x-axis shows date; that is, 140926 equals September 26, 2014. The y-axis shows number of attempts.

For More Information

About Dell Software

Dell Software helps customers unlock greater potential through the power of technology—delivering scalable, affordable and simple-to-use solutions that simplify IT and mitigate risk. The Dell Software portfolio addresses five key areas of customer needs: data center and cloud management, information management, mobile workforce management, security and data protection. This software, when combined with Dell hardware and services, drives unmatched efficiency and productivity to accelerate business results. www.dellsoftware.com.

If you have any questions regarding your potential use of this material, contact:

Dell Software
5 Polaris Way
Aliso Viejo, CA 92656
www.dellsoftware.com
Refer to our Web site for regional and international office information.