The BASH Shellshock bug

Unintended consequences in reuse of software

1 Introduction

On 24 September, advisories were published concerning a bug in the Bourne Again SHell, bash. This software is a command line interpreter for UNIX-like systems, and exists on many platforms, including end-user systems. To make matters worse, bash is used by popular software such as the apache web server or OpenSSH, to provide extendability. To put things in perspective, the apache web server deploys about half of the web sites on the internet today.

Under certain conditions, attackers can force these legitimate services to execute arbitrary commands on the server. This started to happen very soon after publication of the vulnerability, and the bug is now used as vector to spread malware. At the time of writing, systems administrators and web site operators are rushing to patch their systems.

2 The problem

(This section makes use of technical language)

Bash was created 25 years ago to provide a versatile environment with which to administer UNIX systems. It provides scripting facilities that allow among other things the automation of common tasks. Parameters to scripts are often passed as environment variables. The content of these variables can be anything: the name of a file, a username, or even function definitions. In this particular case, the vulnerability lies in the fact that bash continued to process commands after the end of a function definition. More technical information about the problem is available online.

3 Why is this bad?

3.1 Widespread usage

The apache web server uses bash as engine for dynamic web pages, using the Common Gateway Interfaces protocol (CGI) as a means of communication. Other software, for example mail servers, also use bash for back-end operations. The apache web server runs on about 50% of the web servers on the internet, but that is not the full story! Derivatives of apache and bash exist on countless devices as back-end for the administration interface: home routers, media players, Small-Office/Home Office SAN appliances, etc. There are millions of these devices in the field, often operated by people with little to no expertise in systems administration, and who have no idea that their device is offering this kind of service and is vulnerable, and even less know how to fix the problem.

3.2 Inappropriate reuse

Bash was never intended to be reachable online. The intent was rather that it is used locally or over trusted links by knowledgeable people to perform for example systems administration tasks. However, as already stated, the apache web server uses this feature to provide CGI pages (see Glossary), and this is only one of the most common uses: mail servers and countless other devices can pass commands to bash and might therefore be used as components of attack vectors.

The result of this bundling is that a tool and in particular extended functionality that were meant to be used by trusted persons in trusted environments are reachable by anyone with an internet connection. The bug, which has little impact in the intended environment, can now have devastating effects.

Most likely, the choice of bash as a mechanism to provide dynamic web pages was one of convenience rather than a carefully studied selection. Bash was already ubiquitous on systems where the web server could run, but the authors used it without caring for unnecessary features.

3.3 Obsolete features

The idea of allowing a web server to execute CGI scripts dates back to 1993. In the early days of the internet, CGI was the only way for a web site to provide dynamic pages. Since then, alternatives to CGI exist like servlets or ASP. These are more efficient and better integrated in the web server, and make for a more comfortable development environment. Nevertheless, the feature remains and there are many reasons for this: fear of breaking popular sites; difficulty of removing old code; failure to review the feature list, etc.

4 Recommendations

4.1 Think before reusing

Reusing existing code and programs is a common development practice. Programmers are encouraged not to re-invent the wheel. However, they often reuse software without completely understanding the ramifications of the reuse, and without regard for unnecessary features in the reused software. Developers are under pressure to deliver faster, and have no time, no expertise, nor incentive to conduct proper review of reused code. This was also one of the underlying problems in the recent "Heartbleed" incident.

We recommend that any code reuse be the result of a thorough review and understanding of the reviewed code. Management must allow and encourage developers to take the time to perform this review, and developers must not only focus on speed and convenience. Unnecessary features should be stripped. Should that prove impossible or take too much time, then the code should not be reused, or any resulting risk consciously accepted.

4.2 Review features

The idea of allowing a web server to execute CGI scripts dates back to 1993. Since then, alternatives to CGI (such as servlets or ASP) exist. In general, these alternatives are more efficient and better integrated in the web server, and make for a more comfortable development environment. However the CGI feature still exists today, and there are many reasons for this: fear of breaking popular sites; difficulty of removing old code; failure to review the feature list, etc.

We recommend providers of online services or infrastructure to regularly review the need to keep existing features in deployed software, and to plan for their removal when better alternatives exist. Product teams should not be afraid to deprecate old features, and even more when they prove to be dangerous.

Glossary

ASP (Active Server Page): Technology created by Microsoft to facilitate the development of web applications on their web server software

CGI (Common Gateway Interface): The first mechanism that allowed a web server to return dynamic pages

Command-Line Interpreter: A program that acts as text user interface for an Operating System

Environment Variable: A named value that can be used in a script. Usage of the variable in a script will be replaced by its value.

Servlet: Mechanism to extend the functionality of a web server in Java.

Script: Small (usually) set of commands that automate tasks on a computer.

UNIX: Operating System originally developed in the 70's. Linux is one of the most popular variants.

Additional References

[1] NCSC-NL, "Beveiligingsadvies NCSC-2014-0595 [1.00] - Ernstige kwetsbaarheid in Bash verholpen," 25 09 2014. [Online]. Available: https://www.ncsc.nl/dienstverlening/response-op-dreigingen-en-incidenten/beveiligingsadviezen/NCSC-2014-0595+1.00+Ernstige+kwetsbaarheid+in+Bash+verholpen.html.

[2] CERT-SE, "BM14-001 - Allvarlig sårbarhet i bash," 26 09 2014. [Online]. Available: https://www.cert.se/2014/09/bm14-001-allvarlig-sarbarhet-i-bash.

[3] CERT.LV, "Bourne Again Shell (Bash) attālināta koda izpildes ievainojamība jeb "Shellshock"," 25 09 2014. [Online]. Available: https://cert.lv/resource/show/537.

[4] CERT Hungary, "GNU Bash kritikus sérülékenysége - frissült," 25 09 2014. [Online]. Available: http://www.cert-hungary.hu/node/275.

[5] CERT.be, "Severe Bash Vulnerability," 25 09 2014. [Online]. Available: https://www.cert.be/docs/severe-bash-vulnerability.

[6] NCSC-FI, "Shellshock-haavoittuvuutta hyödynnetään aktiivisesti," 25 09 2014. [Online]. Available: https://www.viestintavirasto.fi/tietoturva/tietoturvanyt/2014/09/ttn201409251726.html.

[7] CERT.SI, "SI-CERT 2014-05 / GNU bash ranljivost omogoča izvajanje ukazov na daljavo," 25 09 2014. [Online]. Available: https://www.cert.si/si-cert-2014-05/.

[8] CERT-FR, "Vulnérabilité dans GNU bash," 25 09 2014. [Online]. Available: http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ALE-006/index.html.

[9] CSIRT.SK, "Závažná zraniteľnosť v Bourne Again Shell (Bash)," 25 09 2014. [Online]. Available: https://www.csirt.gov.sk/oznamenia-a-varovania-803.html?id=118.

About "Info Notes" from ENISA

With the "Info Notes" series ENISA aims at giving the interested reader some background and recommendations about NIS related topics. The background and recommendations are derived from past experiences and common sense, and should be taken as starting points for discussions on possible course of action by relevant stakeholders. Feel free to get in touch with ENISA to discuss or inquire more information on the "Info Notes" series (cert-relations@enisa.europa.eu).