Security issues in Perl web apps

Viacheslav Tykhanovskyi

February 17, 2013

Common web security issues

Validate input data!

SQL injections

```
use DBI;
use DBIx::Class;
use Rose::DB::Object;
use ObjectDB;
```

XSS

Blind escaping <, >, ’, " and & is not enough!

XSS

Blind escaping <, >, ’, " and & is not enough!

- Various HTML attributes (href, refresh meta tag, ...)

XSS

Blind escaping <, >, ’, " and & is not enough!

- Various HTML attributes (href, refresh meta tag, ...)
- Not validated JSON response

XSS

Blind escaping <, >, ’, " and & is not enough!

- Various HTML attributes (href, refresh meta tag, ...)
- Not validated JSON response
- Using template variables in JavaScript code

XSS

Blind escaping <, >, ’, " and & is not enough!

- Various HTML attributes (href, refresh meta tag, ...)
- Not validated JSON response
- Using template variables in JavaScript code

Escape taking context into account.

Cookies

- Sign cookies

Cookies

- Sign cookies
- XSS preventing is hard. Set `HttpOnly` cookie flag for better protection.

CSRF

```
Plack::Middleware::CSRF
```

Path traversal

```
../../../../../../etc/passwd
```

Path traversal

```
../../../../../../etc/passwd
```

- Detect `..`

Path traversal

```
../../../../../../etc/passwd
```

- Detect `..`
- `File::Spec->no_upwards(@paths);`

Perl-specific security issues

- No buffer overflow

- No buffer overflow
- Most system commands are embedded

- No buffer overflow
- Most system commands are embedded
- Written by smart people

```
use strict;
use warnings;
```

Tainting

```
-T
```

```
sub is_tainted {
    return !eval {
        eval("#"
             . substr(join("", @_), 0, 0)
        );
        1
    };
}
```

system()

```
system("program $arg");
```

vs

```
system('program', $arg);
```

open()

```
open my $fh, ">$file";
```

vs

```
open my $fh, '>', $file;
```

eval()

```
eval "require $class";
```

eval()

```
eval "require $class";
```

```
load_class("Foo;print 'nice feature!'")
```

0

\o

0

```
\0
```

```
$file = "/bin/ls\0 /etc|";
if (-e $file) {
    open my $fh, $file;
}
```

CGI & ARGV

```
script.pl?foo
```

CGI & ARGV

```
script.pl?foo
```

```
...
$app->run(@ARGV);
...
```

Regular expressions

```
if ($string =~ m/$user_supplied_re/) {
    ...
}
```

vs

```
if ($string =~ m/\Q$user_supplied_re\E/) {
    ...
}
```

Unicode

`utf8`

vs

`UTF-8`

rand()

"rand()" is not cryptographically secure

How to make life easier?

How to make life easier?

- Use modules from CPAN. Many of them are time-proved

How to make life easier?

- Use modules from CPAN. Many of them are time-proved
- Google "OWASP"

How to make life easier?

- Use modules from CPAN. Many of them are time-proved
- Google "OWASP"
- Follow Best Practices

How to make life easier?

- Use modules from CPAN. Many of them are time-proved
- Google "OWASP"
- Follow Best Practices
- Use scanners
  - nikto http://cirt.net/nikto2
  - skipfish http://code.google.com/p/skipfish/
  - w3af http://w3af.sourceforge.net/

Questions?