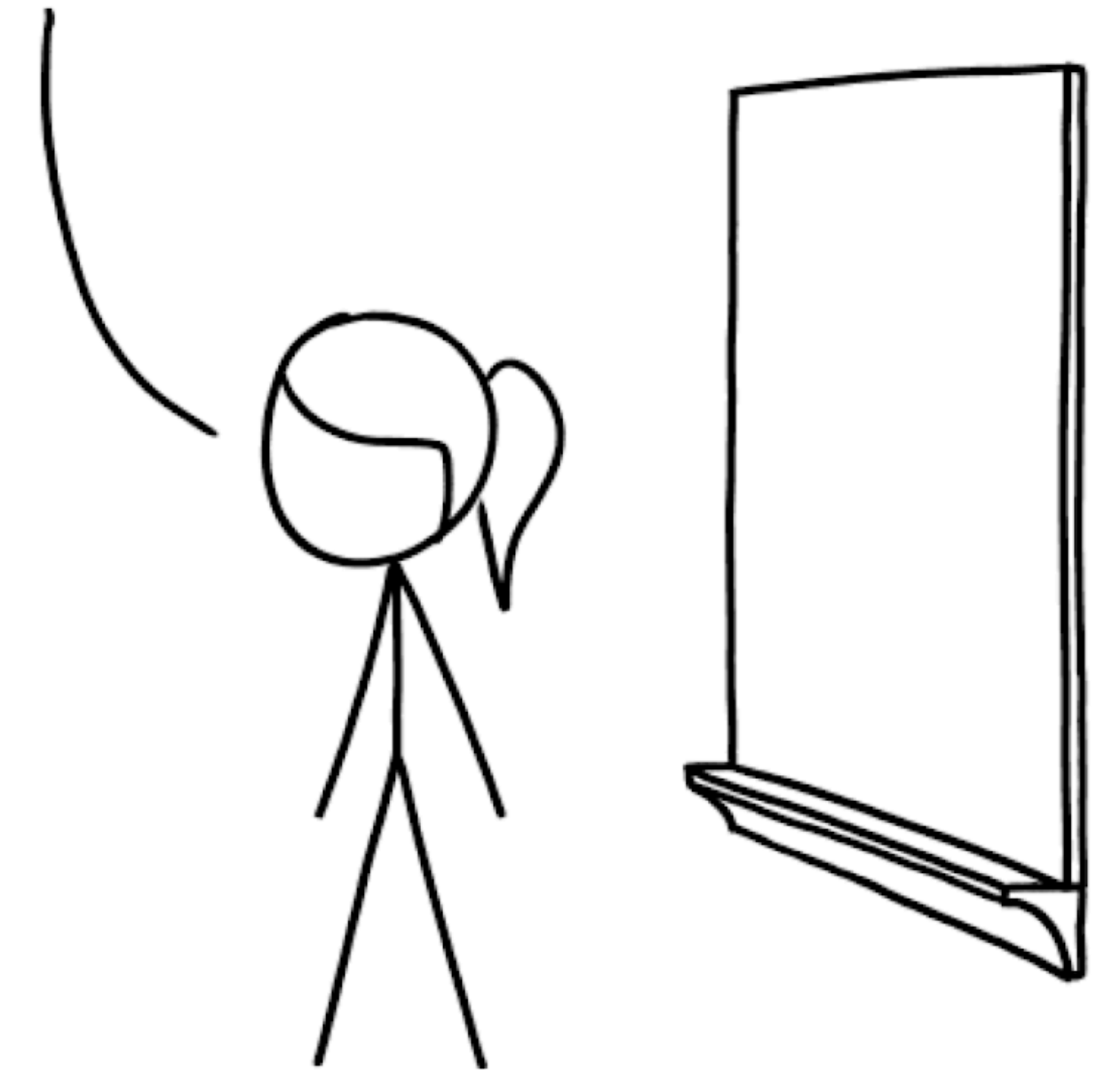


# PeoplePath

## Web Application Security

Ondrej Esler, December 1, 2021

WELCOME TO YOUR FINAL EXAM.  
THE EXAM IS NOW OVER.  
I'M AFRAID ALL OF YOU FAILED.  
YOUR GRADES HAVE BEEN STORED  
ON OUR DEPARTMENT SERVER AND  
WILL BE SUBMITTED TOMORROW.  
CLASS DISMISSED.



CYBERSECURITY FINAL EXAMS

We are a **global provider** of cloud-based solutions for **talent relationship management**.

Since 2002, we have empowered our clients to establish, track and develop **personal relationships** with talent throughout their entire **career lifecycle**.

We offer the most powerful and flexible **candidate engagement** technology in the marketplace.



**Seattle**  
USA



**New York**  
USA



**Munich**  
Germany



**Pilsen**  
Czech Republic

**60+**

employees

**100+**

clients in all  
major sectors

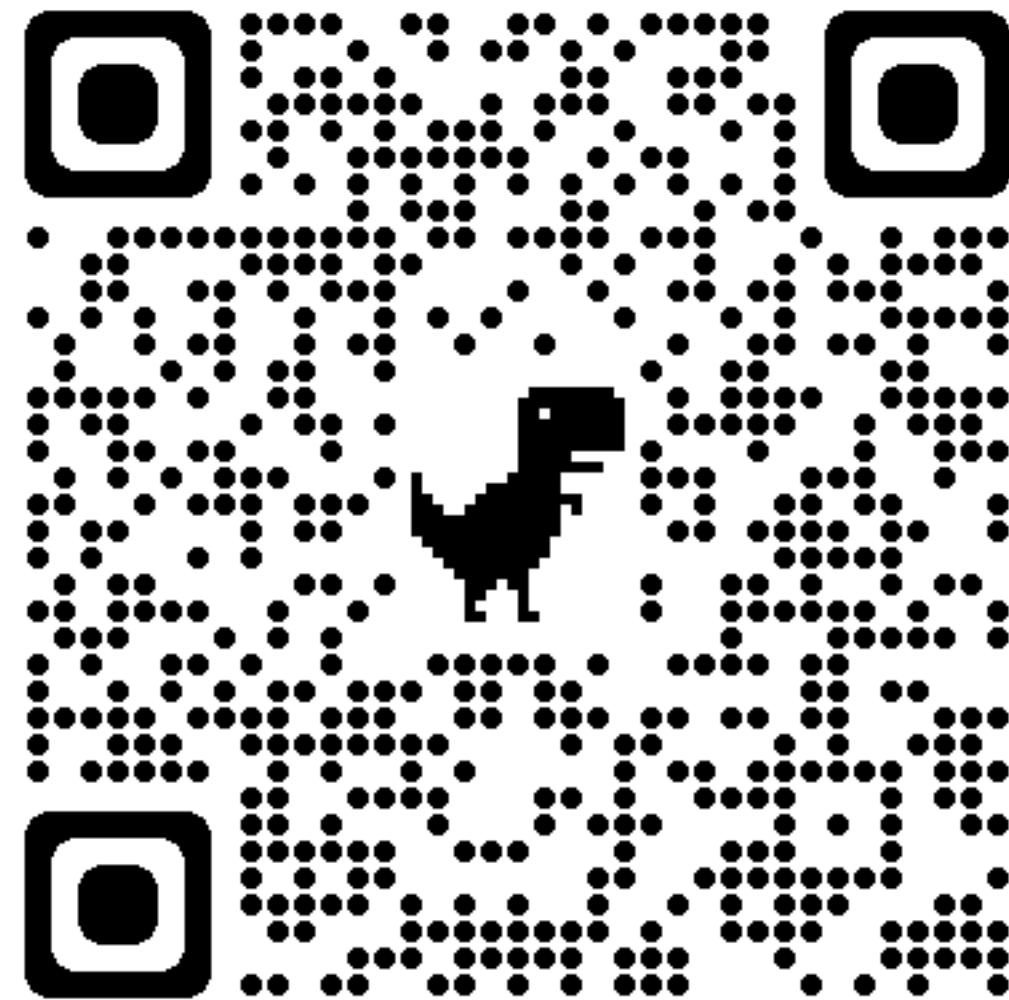
**250+**

years of combined  
talent engagement  
experience

**1,500,000+**

talent profiles across  
clients globally

<https://github.com/peoplepath/workshop-web-security>



# **Why's security important?**

# ISO 27001

INFORMATION SECURITY MANAGEMENT SYSTEM

# ISO 27001

ISMS

Human Resources

Access control

Cryptography

Physical and environmental security

Communications security

...

and many more

# Development security

Coding standards, Security principles, Penetration testing



# What's penetration testing?

# OWASP

Open Web Application Security Project

<https://www.owasp.org>

# OWASP top 10

List of most common security issues

# **Most common attacks on the Web?**

# SQL injection

# (SQL) injection

# SQL injection

```
$sql = 'SELECT * FROM user WHERE id = ' . $_GET['id'];
```

# Demo



# SQL injection

```
$sql = 'SELECT * FROM user WHERE id = :id';
```

```
$stmt = $pdo->prepare($sql);
```

```
$stmt->execute([':id' => $_GET['id']]);
```

# Cross-Site Scripting

# XSS

# XSS

```
echo 'You are searching for ' . $_GET['query'];
```

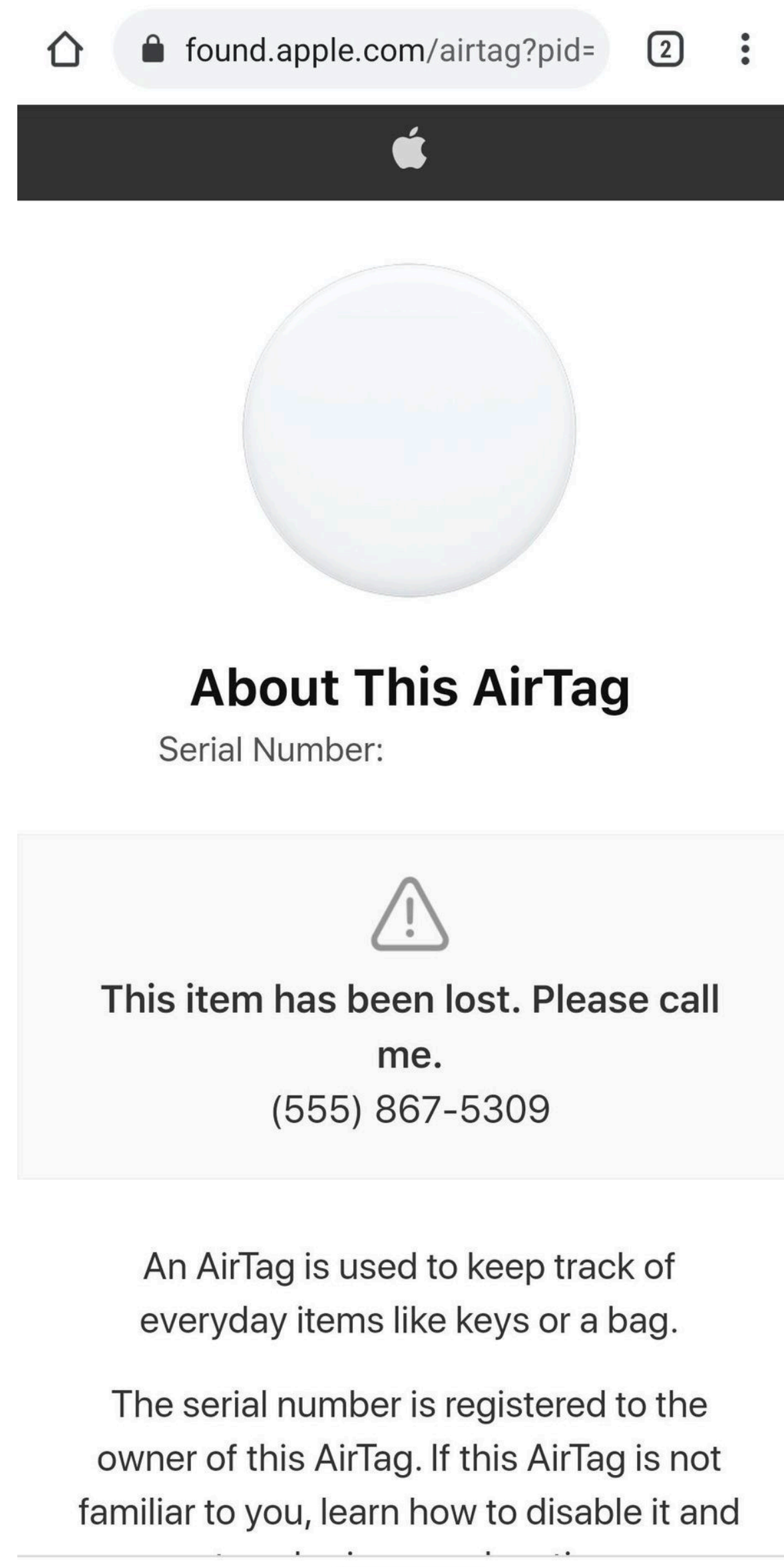
# Demo



# XSS







# XSS

```
<script>window.location='https://malicious.com';</script>
```



# XSS



**\*andy**  
@derGeruhn



 Follow

`<script  
class="xss">$('.xss').parents().eq(1).find('a')  
.eq(1).click();$('[data-  
action=retweet]').click();alert('XSS in  
Tweetdeck')</script>` ❤️

 Reply  Retweet  Favorite ... More

RETWEETS

83,502

FAVORITES

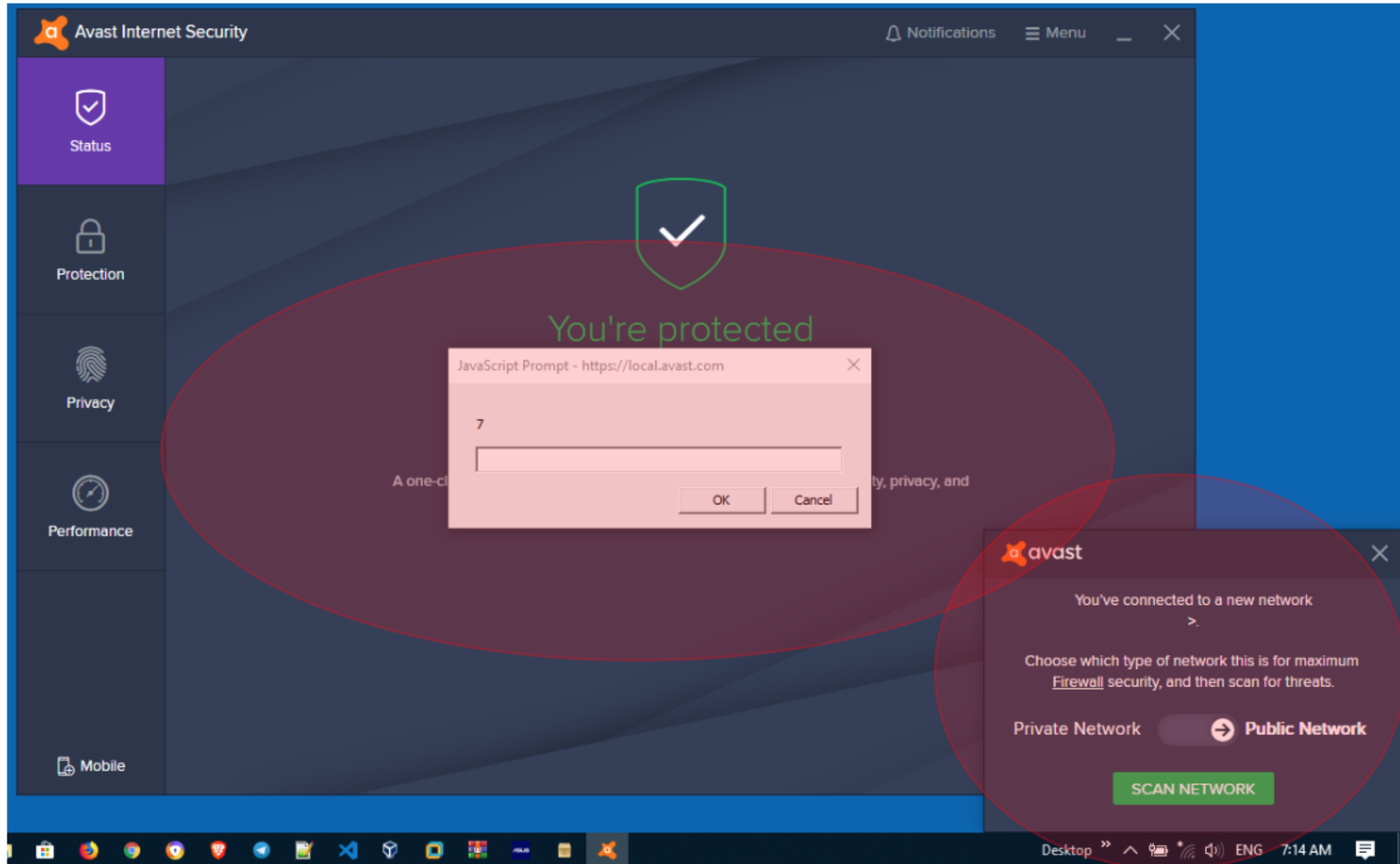
9,709



9:36 AM - 11 Jun 2014

source: <https://www.zdnet.com/article/tweetdeck-wasnt-actually-hacked-and-everyone-was-silly/>

# XSS



source: <https://medium.com/bugbountywriteup/5-000-usd-xss-issue-at-avast-desktop-antivirus-for-windows-yes-desktop-1e99375f0968>

# XSS

```
echo 'You are searching for ' . htmlspecialchars($_GET['query']);
```

XSS

CSP

XSS

# Content-Security-Policy

# XSS

**Content-Security-Policy:** default-src 'self'; img-src https://\*; child-src 'none';

source: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>

# CSRF

# Cross-site request forgery



# CSRF

```
if ($isAuthenticated) payTo($_POST['user_id']);
```

# Demo

# CSRF

```
$csrfMatches = $_SESSION['csrf'] == $_POST['csrf'];
```

```
if ($isAuthenticated && $csrfMatches)  
payTo($_POST['user_id']);
```

```
$_SESSION['csrf'] = bin2hex(random_bytes(16));
```

# Cross-site request forgery

# SameSite cookie

# CSRF

```
; php.ini configuration  
session.cookie_samesite = 1
```

source: <https://www.php.net/manual/en/session.configuration.php#ini.session.cookie-samesite>

# CSRF

Set-Cookie: PHPSESSID=123...; SameSite=Strict

source: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite>

# Directory Traversal



# Directory Traversal

`https://example.com?download=../../../etc/passwd`

# Demo

# Directory Traversal

```
; php.ini configuration  
open_basedir = /var/www;/data/uploads
```

source: <https://www.php.net/manual/en/ini.core.php#ini.open-basedir>

# Common web development mistakes?

# Sensitive data exposure

HTTP/1.1 200 OK

Server: Apache/2.4.51 (Unix)

X-Powered-By: PHP/8.1.0

sources: <https://httpd.apache.org/docs/current/mod/core.html#traceenable>, <https://httpd.apache.org/docs/current/mod/core.html#servertokens>, <https://www.php.net/manual/en/ini.core.php#ini.expose-php>

# Sensitive data exposure

```
$password = md5(`APm9FK7Yn`);
```

# Sensitive data exposure

```
$password = password_hash( `APm9FK7Yn`, PASSWORD_DEFAULT);
```

# Tabnabbing

```
<a href="example.com" target="_blank">
```



# Tabnabbing

```
<a href="example.com" target="_blank" rel="noopener">
```

# Tabnabbing

```
<a href="example.com" target="_blank" rel="noopener">
```

# Security misconfiguration

```
apt-get install mongodb
```

```
service mongod start
```

# **Weak authentication and session management**

# Using Components with known vulnerabilities

# Best practices

# **What's most important?**

(in web security)

# Everything is user input!

form data, files, headers, ...



# Implement with proven frameworks/libraries

(if you can)

basic security out-of-box

# Learn about secure configuration

or use Cloud

# **Add CAPTCHA** **on public forms**

keep that pesky robots away

# Use password manager

never reuse password

# Use 2FA

especially on github/gitlab, etc.

# Use strong cryptography

bcrypt for passwords, SHA256 for hashing

# Use strong cryptography

bcrypt for passwords, SHA256 for hashing

# Use strong cryptography

bcrypt for passwords, SHA256 for hashing



# Make a plan what to do in case of security attack

you'll be hacked soon or later

<https://www.peoplepath.cz/>

