Web Vulnerabilities

Agenda

- Why hack the web?
- SQL injection (SQLi)
- Cross-site scripting (XSS)
- Cross-site request forgery (CSRF)
- Concurrency vulnerabilities/race conditions
- A couple other fun things

But first…

Don’t try this on any site where you don’t have explicit permission to do so!!!

- Consult site policies for major sites
  - E.g. you can test Microsoft’s sites as long as there is no impact to real users
    - Your account may get auto-banned though
  - Ask the security team if you’re not sure

Why hack the web?

- Because it’s where the data is
- Attackers want to:
  - Steal confidential information like credit card #s and “secret plans”
  - Vandalize for protest or notoriety
- Websites often have lots of this stuff all in one place
- More functionality moving to the web = more reasons to hack it

Another reason to hack the web

- It’s easy
- Websites are:
  - Always online
  - Rarely monitored by a person
  - Usually running a mix of commodity software and handwritten code
    - Web security is not widely understood by developers
  - Often the shared responsibility of many people
    - More cracks to fall through

SQL Quick Primer

- SQL: Standard Query Language = how you access databases
- SELECT: Fetch data
  - `SELECT valuesYouWant FROM databaseTable WHERE conditions`
- INSERT: Add data
  - `INSERT INTO databaseTable(columnList) VALUES(valuesToInsert)`
- --: comment i.e. ignore everything after this
- ;: end statement (just like C)

SQL Injection (SQLi)

- Run your own SQL queries on the back end server
  - Steal data
  - Make yourself admin
  - Run code
    - MS SQL Server has the ability to run commands on Windows
- Happens when developers concatenate user data into a literal statement
- The main way that websites get hacked

SQLi Example

- URL: http://example.com/getCreditCard.php?username=David
  - Function is to display the user’s credit card number
- Code:
  - SELECT Number FROM CreditCards WHERE Username = ‘$username’
    - In PHP, $username is replaced by the actual data before execution

SQLi Example: Attack

- Attack URL:

  http://example.com/getCreditCard.php?username=**‘ or 1=1;--**
- What gets executed?
  - SELECT Number FROM CreditCards WHERE Username = ‘**‘ or 1=1;--’;**
- All credit card numbers are displayed!
- Another attack using the same URL:
  - http://example.com/getCreditCard.php?username=**‘; INSERT INTO Administrators(Name, Password) VALUES(‘AttackerName’, ‘AttackerPassword’);--**
  - SELECT Number FROM CreditCards WHERE Username = ‘**‘; INSERT INTO Administrators(Name, Password) VALUES(‘AttackerName’, ‘AttackerPassword’);--’;**

SQLi: Detecting and Preventing

- To find SQLi, replace parameters in URLs with ‘, ‘--, or other SQL delimiters.
  - If this causes an error, you might be looking at a SQL Injection bug
- To prevent SQLi, **use stored procedures**.
  - Stored procedures avoid the need to convert user input to text
  - Do *not* try to strip dangerous characters out of user input – too easy to miss something
    - If you have to, use a whitelist – only allow good characters e.g. alphanumeric

Javascript Security Primer

- Javascript is code for web pages
    - `<script>code();</script>`
- The entire page and all its functionality are accessible to the script
- Can also add new content and replace existing content

Javascript Security Primer Continued

- Same-domain only
  - If Javascript tries to access data from a different domain, the browser asks the user for permission
  - https:// pages are considered a separate domain and there are additional restrictions
  - Some exceptions:
    - Fetching images and similar common, normally safe operations
    - A site can explicitly give permission to access specific other domains
- <iframe>
  - Allows embedding another web page, which can then be controlled by the script in the parent frame
  - The iframe can fill the screen with the parent frame invisible

Cross-Site Scripting (XSS)

- Run attacker script on some other site e.g. Hotmail
  - This script can do anything that the user can do
- Happens when attacker data is displayed to a victim from a page in the target domain
- Two types:
  - Persistent/stored XSS: attacker script is *stored* on the server e.g. forum comment
  - Reflected XSS: attacker script is *incorporated into a URL* that the victim is lured to click e.g. http://example.com/target.php?value=<script>alert(“xss”);</script>

XSS Example: Stored XSS

- In our example website, when you type a comment, it gets added to the site and displayed to every user as text
- Attacker submits the comment:
  - <script>commentsField="I love Windows Vista!";form.submit();</script>
- Now when you visit the page, your browser sees:

```
<HTML><BODY>…stuff…<script>commentsField="I love Windows
Vista!";form.submit();</script>…
```

And executes it on your behalf!

And this happens for everyone who visits the site!

XSS Example: Reflected XSS

- Same website, different page: this one lets you view comments specified in the URL
  - http://example.com/viewComment.php?comment=Hi everyone!
  - `<HTML><BODY>…Hi everyone!…</BODY></HTML>`
- New attack:
  - http://example.com/viewComment.php?comment=<script>page.location="http://example.com/addComment.php?comment=I love Windows Vista!";</script>
  - `<HTML><BODY>…<script>page.location="http://example.com/addComment.php?comment=I love Windows Vista!";</script>…</BODY></HTML>`
- If I can get you to click this link, the script will execute, causing you to add a comment proclaiming your love for Vista

XSS: Detecting and Preventing

- To find XSS, try putting <script>alert(“xss”);</script> in URL parameters and form fields.
- To prevent XSS, turn all special characters into encoded equivalents
  - E.g. < and > into &lt; and &gt;
  - Or use a sanitizing library if some tags like <b> are needed
- Script in browsers **runs by default**: the website has to take active measures to remove script from user-submitted content
  - Some frameworks like ASP.Net do this semi-automatically

XSS Variations

- Encoding bugs
  - URL encoding: %20
  - Hex encoding: &#20
  - Did the script filter check for these?
- Javascript injection
  - Eval("alert('this has a ' + crossSiteScript + ' flaw');")
  - Eval("alert('this has a '**);alert('XSS');doEvilStuff();//** flaw');")
  - alert('this has a '**);** **alert('XSS'); doEvilStuff();** **//** flaw');

Cross-Site Image Overlay (XSIO)

- Like XSS, but instead of running a script, overlay other stuff on the page
- E.g.
  - Put an attacker-controlled text box over the password box
  - On a “Grant permissions to attacker?” page, display a picture of the “Deny” button over the real “Approve” button
- Useful when you can XSS an HTTP site but the password is protected by SSL
- Sometimes you can do XSIO when you can’t do XSS

Cross-Site Request Forgery

- A URL that does something bad in a single click
- http://bank.com/transferMoney.php?fromAccount=yours&toAccount=mine&amount=100billiondollars
  - Then send this link to the user in an email claiming they've won the lottery and click here to claim their prize…
- Preventing CSRF:
  - Embed a random or secret token in every page that calls a sensitive function, include the token in the form submission/URL, and check it before processing the transaction
  - Check the referrer header for sensitive pages: the click should have come from your own site
    - Not ideal as a standalone measure: e.g. the link could be posted in a forum on your site

Preventing CSRF example

- http://bank.com/prepareMoneyTransfer.php:

  <HTML><BODY>…<FORM>

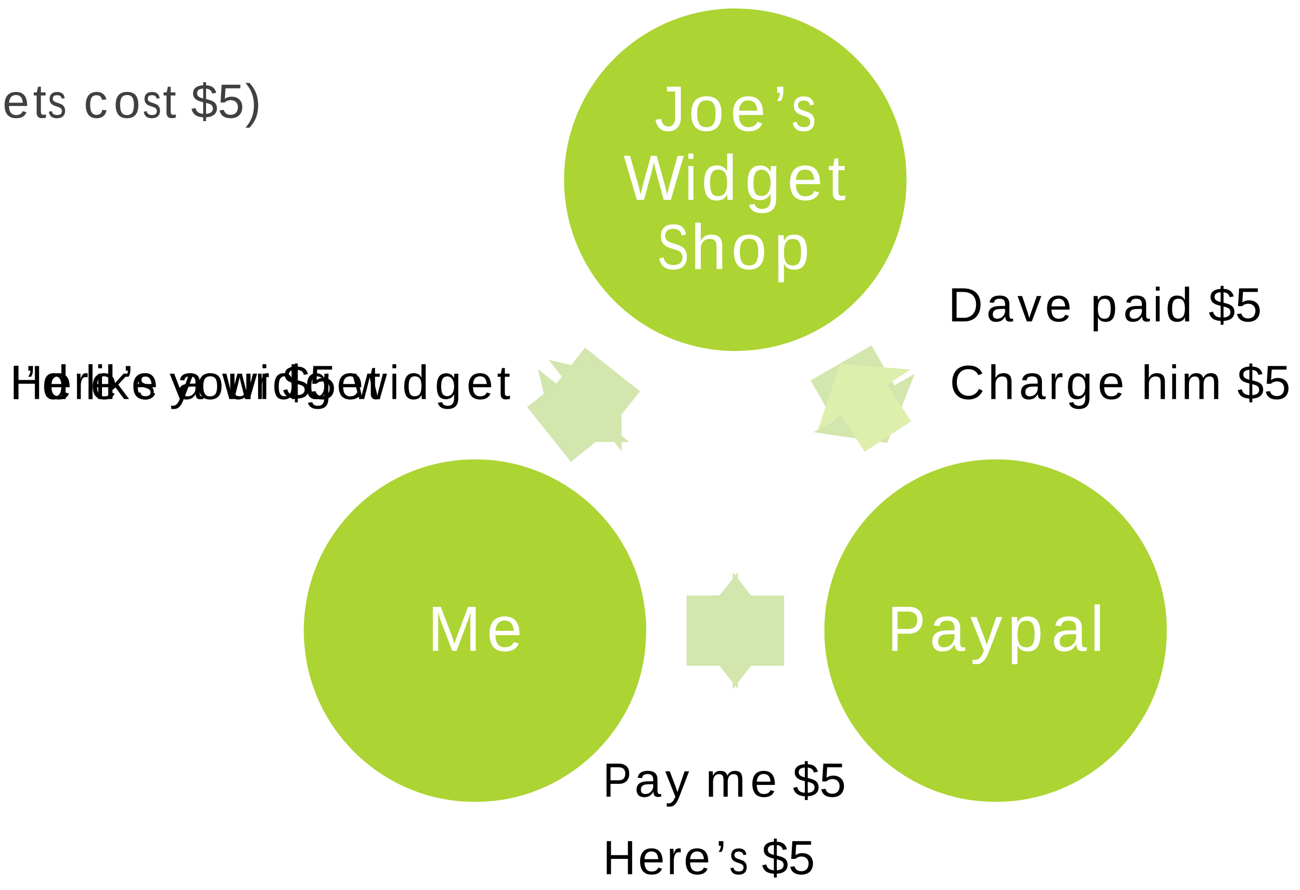
  <input type="hidden" name="secretToken" value="13245"/>…
- http://bank.com/transferMoney.php?fromAccount=yours&toAccount=mine&amount=100billiondollars**&secretToken=13245**
  - Server validates that token matches
- Needs some randomness, attacker can try brute force
  - Lure you to their own web page
  - Run a script loop trying guesses

Consistency Errors/Race Conditions

- Asynchronous processing
  - Outsourced payment processing (e.g. Paypal)
  - Outsourced authentication/login (e.g. Facebook Connect)
  - Sites that let you log in before validating your email address
- The part that “gives” needs to make sure the part that “takes” is happy before allowing the user to proceed
  - Before giving access, make sure you got paid, got a real password, etc.

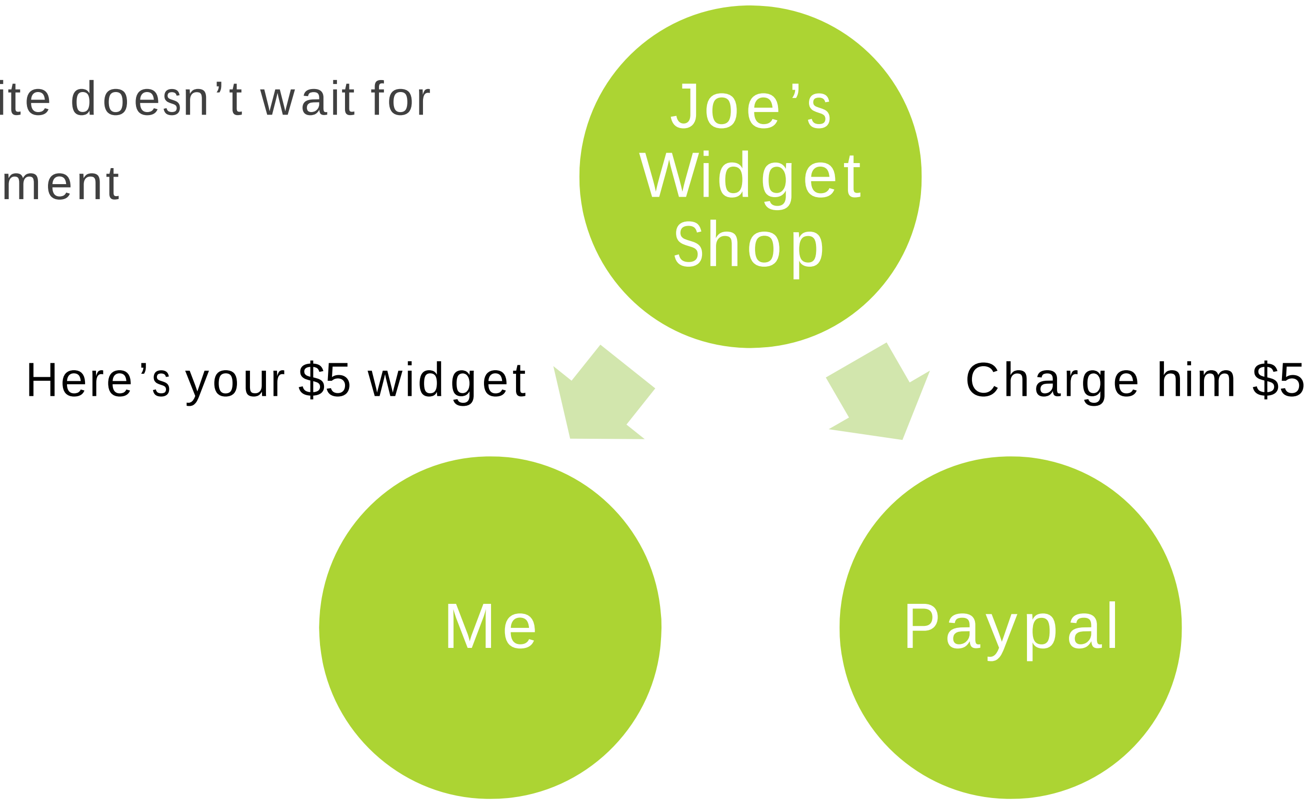
How it should work

- (widgets cost $5)



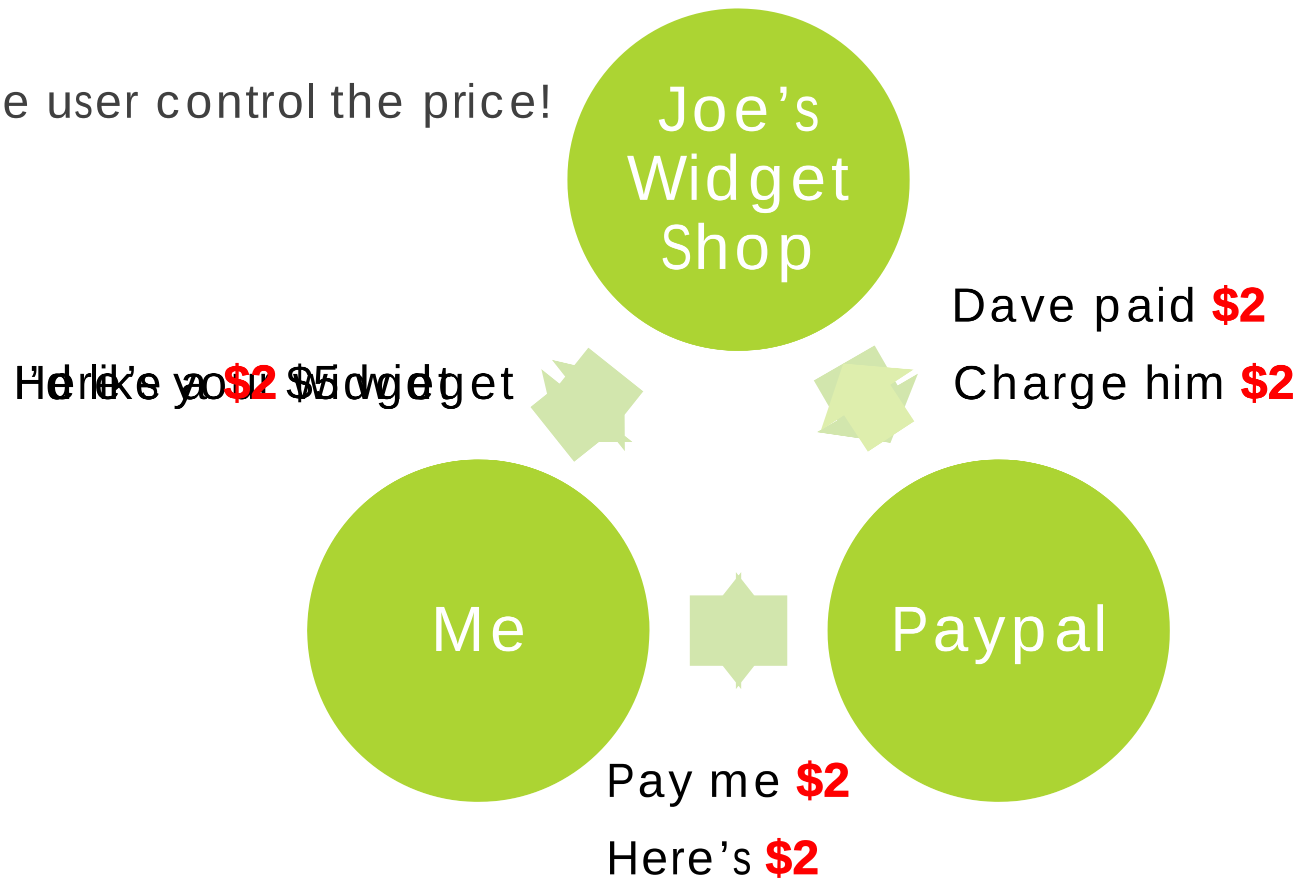
How it can go wrong

- The site doesn't wait for payment



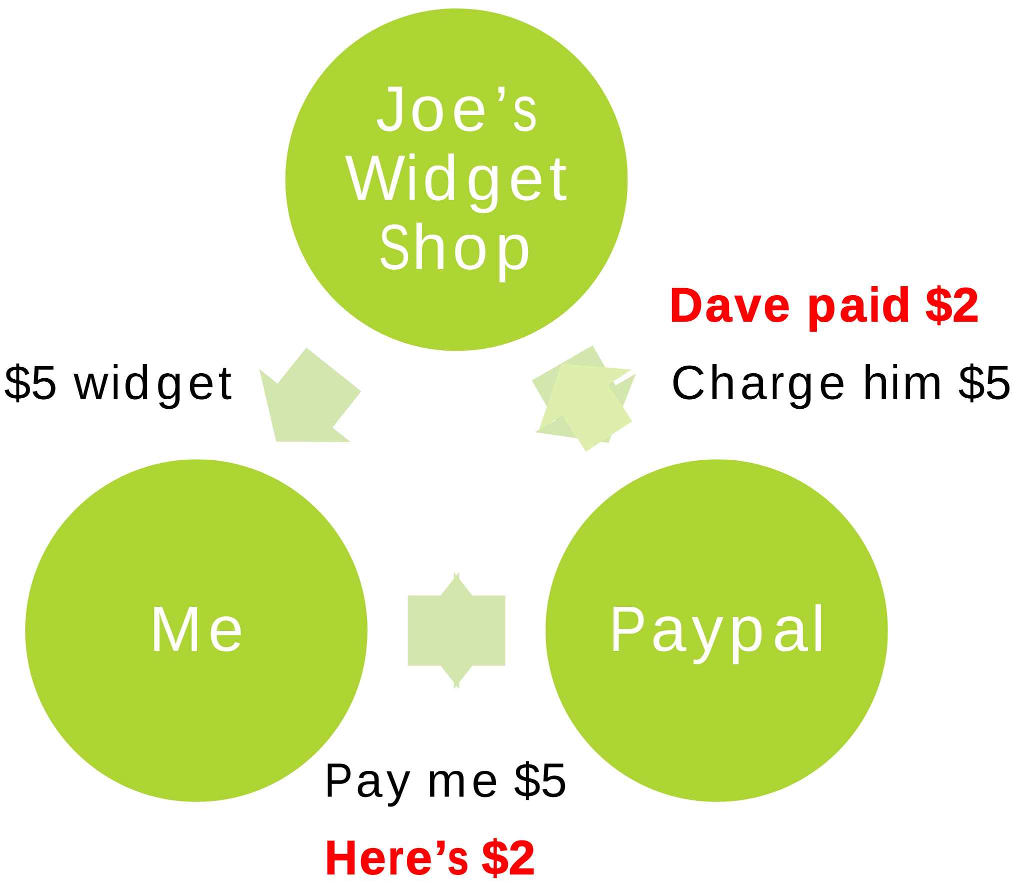
How it can go wrong

- Let the user control the price!



How it can go wrong

- Pay less

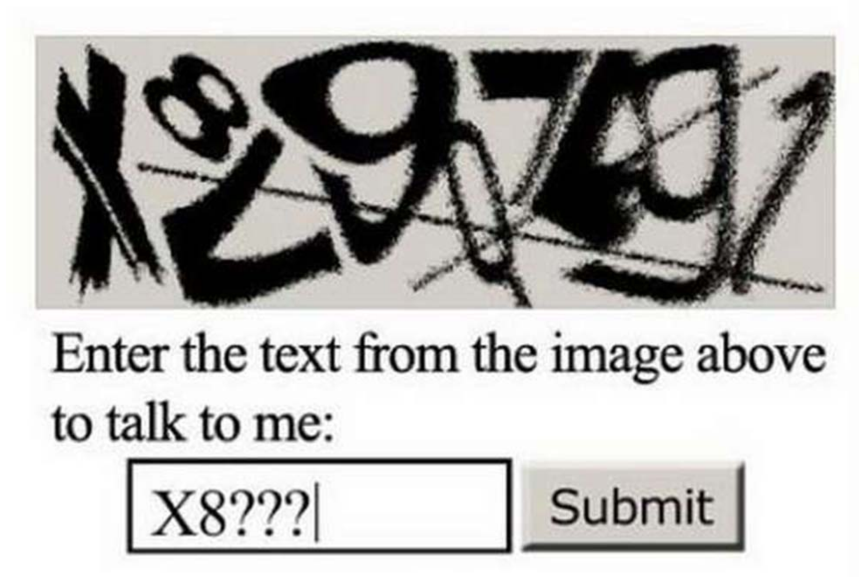


Fraud and Abuse

- Not a vulnerability: a way to use the site's intended functionality to do something bad
  - Spam and scams
  - Review fraud: post something for sale, then post lots of positive reviews
    - Recent example: "Shake Your Phone to Recharge It" app
  - SEO: create a site selling something, then post lots of links to it on other pages to trick search engines
  - Money/credit card laundering

CAPTCHA

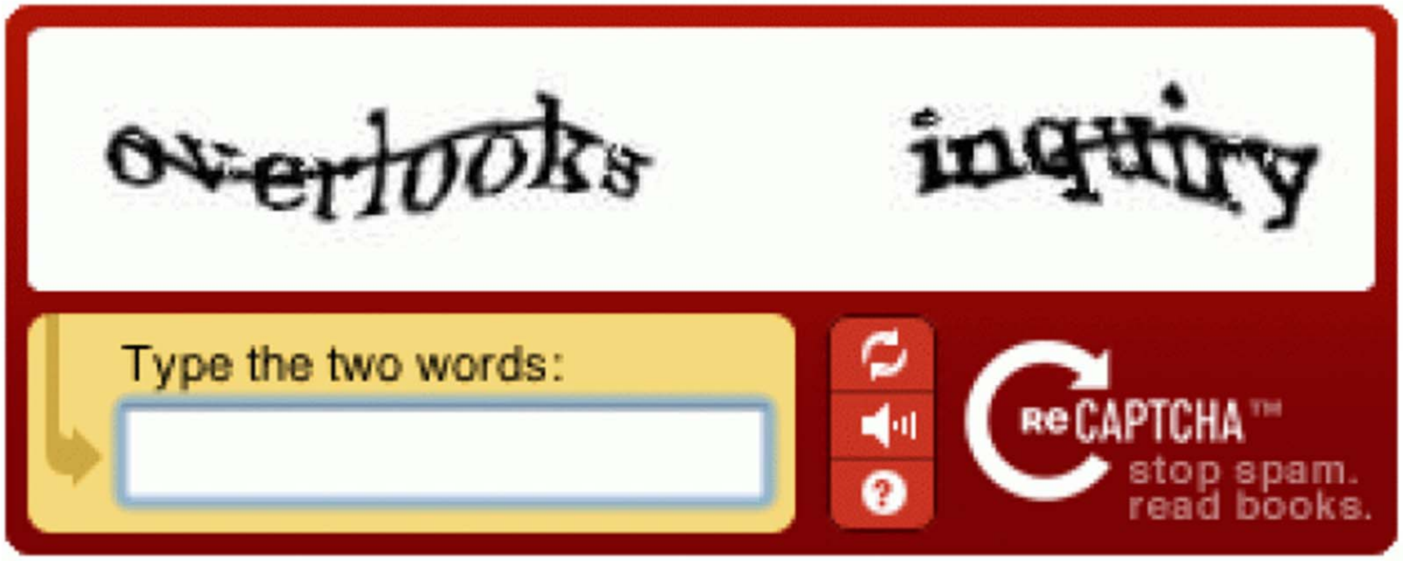
- Completely Automated Test to tell Computers and Humans Apart
- Prevent automated generation of accounts for spam, etc.



Enter the text from the image above to talk to me:

X8???

ReCAPTCHA


overlooks
inquiry
Type the two words:
ReCAPTCHA™
stop spam.
read books.

ReCAPTCHA

- Two words
  - One is known
  - One is unrecognizable text from a scanned document with historical merit
- First word provides CAPTCHA function
- Second word provides OCR function to help digitize books
  - Multiple samples per word filter out bad readers
- The known word may be first or second
  - Hack: if it has punctuation it’s always the unknown word
- Audio CAPTCHA for blind users

Problems with CAPTCHA

- OCR has gotten good: hard to make text that computers can’t read but humans can
  - Spammers might be okay with a 1% success rate since attempts are nearly free
- Audio recognition is even better: Audio CAPTCHA cracked more than text
  - Hard for computers to generate speech that computers can’t understand
- Even easier: use humans to solve it for you
  - Pay low-wage workers in poor countries
  - Require solving CAPTCHA to see valuable content, and instead of serving your own CAPTCHA, host one from the service you’re trying to abuse

Benefits of problems with CAPTCHA

- Massive increases in effectiveness of OCR and related machine learning techniques!
- Digitized books => Project Gutenberg (free public domain books)

Passwords

- 20% of all passwords are one of the 100 most common
- People will tell you their password for a pen
- Recommendation: a long, uncommon, non-plain-English password, using uppercase, lowercase, numbers and symbols, different for every site.
  - Good luck with that.
- Two-factor auth: something you know and something you have
  - Smart cards, cell phones, trusted PCs, dedicated keychain devices
  - Relatively expensive, hardware-specific
- Biometrics: really good until compromised once, then useless forever
- Implicit knowledge e.g. recognizing pictures of your friends

Social Engineering

- "There's no patch for human stupidity"
- Talk the phone rep into resetting the password
- Talk the victim into installing the backdoor
- Lots of fun techniques and ruses
  - Search engine hacking: secret questions are usually weak
  - People want to be helpful, so let them!
  - People fear authority figures, so be one.

Social Engineering Tricks

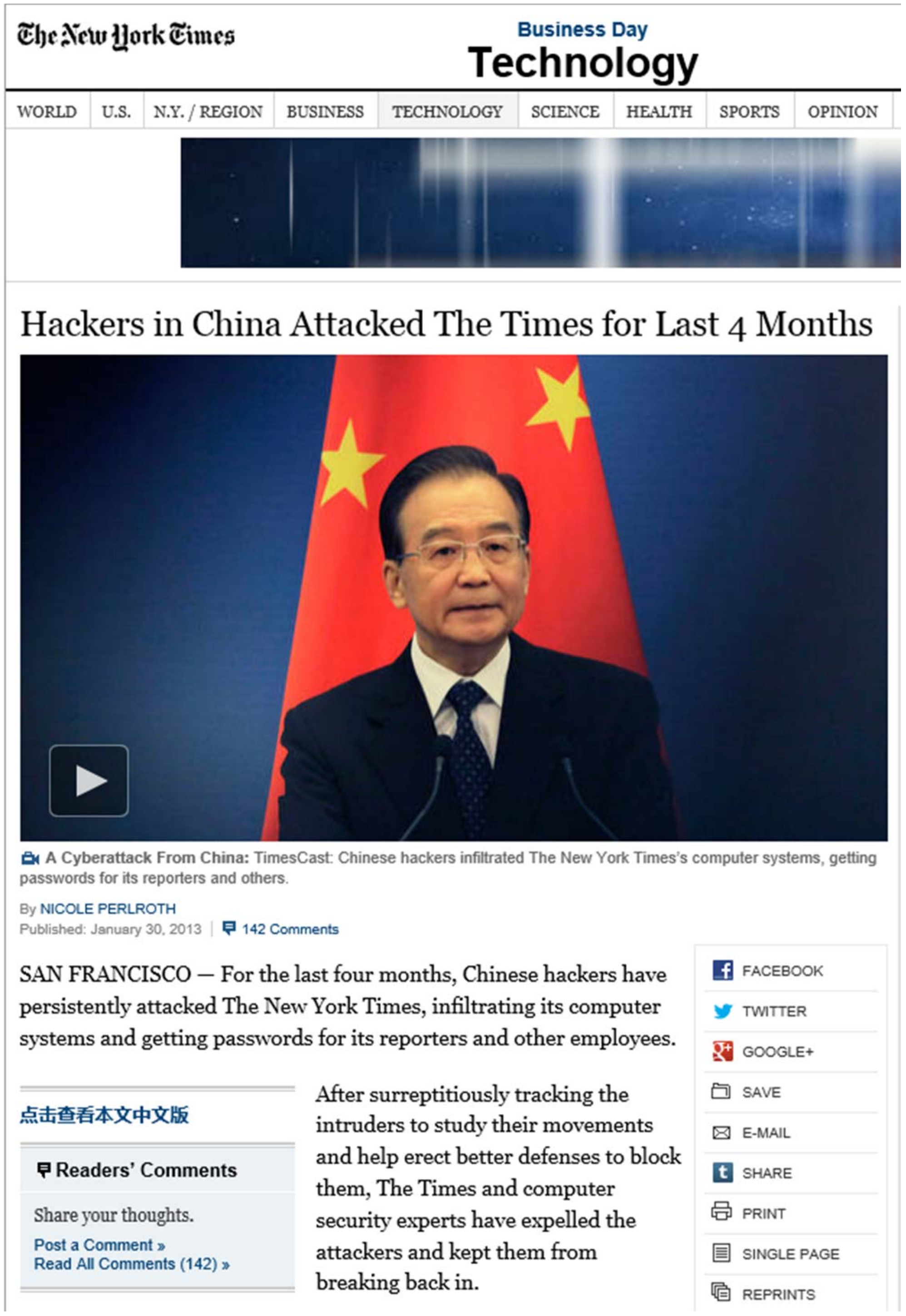
- Elicitation: build rapport and steer the conversation towards the topic at hand
  - Ideally, let the target steer the conversion there
  - Seed the conversation with keywords in innocuous contexts
- Pretexting: pretending to be someone else
  - Useful roles:
    - Repairman
    - The boss
    - The tech support guy
    - Interview candidate
  - Don’t claim knowledge you don’t have.

DNS hacks/site defacements

- Old old way: hack the registrar
- Old new way: hack end users to use the wrong IP address
  - Kaminsky’s DNS attack
- New way: SE the registrar into handing you control of the domain
  - A favorite of Anonymous

Reminder: Ethics

- Don’t do this stuff if you don’t have permission
  - It is very often illegal (I am not an attorney)
- You are physically present = if things go wrong they can go *very* wrong
  - Law enforcement officers are experts in social engineering
- If you want to practice, do it in a situation where no harm is involved
  - E.g. try to get secret but harmless information from your friends, talk your way to the front of the line at a nightclub, etc.



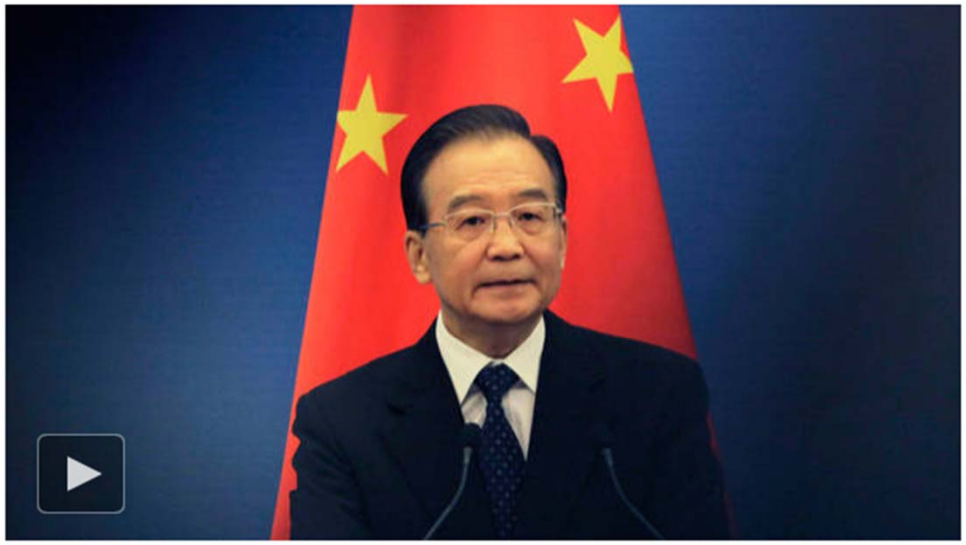
The New York Times

Business Day

Technology

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

Hackers in China Attacked The Times for Last 4 Months



A Cyberattack From China: TimesCast: Chinese hackers infiltrated The New York Times's computer systems, getting passwords for its reporters and others.

By NICOLE PERLROTH

Published: January 30, 2013 | 142 Comments

SAN FRANCISCO — For the last four months, Chinese hackers have persistently attacked The New York Times, infiltrating its computer systems and getting passwords for its reporters and other employees.

After surreptitiously tracking the intruders to study their movements and help erect better defenses to block them, The Times and computer security experts have expelled the attackers and kept them from breaking back in.

点击查看本文中文版

Readers' Comments

Share your thoughts.

Post a Comment »

Read All Comments (142) »

FACEBOOK

TWITTER

GOOGLE+

SAVE

E-MAIL

SHARE

PRINT

SINGLE PAGE

REPRINTS

Open Floor

- Ask Me Anything!