Discovering & disclosing a privacy leak

Giorgos Zervas
Joint work with John Byers & Michael Mitzenmacher
My disclaimer...

- This is by no means advice on what you should do in a similar case.
- It is just our story.
What next?

Question 1:
Do you disclose the leak?

Question 2:
If yes, how?
If not, why?
<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Be ethical</td>
<td>Who cares?</td>
</tr>
<tr>
<td>Protect user data</td>
<td>“Shooting the messenger”</td>
</tr>
<tr>
<td>CS giving back to the world</td>
<td>Computer Fraud and Abuse Act</td>
</tr>
</tbody>
</table>
“Shooting the messenger”

Something as simple as:

- path traversal
- invalid form input

...can be illegal under the CFAA
Andrew Auernheimer, security researcher under attack.

- Security researcher (founder of Goatse Security)
- Recipient of TechCrunch "Crunchie" award for public service
- Free speech advocate (former sysadmin for Encyclopedia Dramatica)
- Famous Internet troll and all-around fun-loving jackass
- Weev on Wikipedia

Andrew Auernheimer (aka "Weev") is currently released on bail after being arrested by federal agents. His only crime? Embarrassing a Fortune 500 company.

This month, federal prosecutors have charged him with several serious felonies (including identity theft) because he blew the whistle on AT&T cutting corners with their security. They'd published the email addresses of thousands of their own subscribers on the public web through negligence. Andrew's team (Goatse Security) immediately alerted the media.

This sort of reporting of vulnerabilities is in the best interest of the public, and the US attorneys are attempting to set a dangerous precedent for security researchers who publish vital information that could embarrass large companies.

Out on bail, he is prohibited from using a computer (his only source of livelihood) despite not yet having even been charged with a crime. His Arkansas home was raided, his equipment was stolen, and his person was dragged half-way across the country to face trial in New Jersey. Now he's facing tens of thousands of dollars of legal expenses.

He was arraigned on the 21st of July 2011

He urgently needs your help! Please donate now!

NEW: show your support! For every donation to Andrew's legal fund of USD$25 or more, we'll send you a FREEWEEV.INFO t-shirt (sizes S/M/L/XL), postage paid to any continental USA shipping address. See what the shirts look like here.

Why Is This Important?

This case will set a precedent, both legal and social, for the types of retribution that corporate and government interests can enact against whistleblowers such as independent researchers and journalists.
The Case

In June, 2010, Andrew's ragtag band of researchers at Goatse Security discovered that, due to cutting security corners, AT&T (NYSE:T) was publicly divulging the email addresses of their subscribers using Apple's (NASDAQ:AAPL) iPad 3G tablet computing device.

His team successfully downloaded over 100,000 subscriber email addresses from AT&T's public website, including those belonging to Fortune 500 CEOs, members of the military, and federal government officials. After realizing the vast potential impact this data could have in criminal hands, he immediately alerted the media. The email addresses were never publicized. This wasn't "stealing" in any sense of the term - it was the same type of request made of a server that Google makes thousands of times a day of every web page on the Internet.

AT&T had taken no security measures whatsoever to protect their customers' email addresses, serving them out on the public web to any request made with a valid serial number of an iPad 3G's SIM chip. The problem? These serial numbers are sequential integers - not passwords.

The U.S. Attorney prosecuting the case (Paul Fishman) has confirmed to the media that there is no evidence that the addresses were disseminated for criminal purposes.

In fact, other than to ONE reporter, the email addresses were never used or divulged in any way.

Important Points

- Subscriber data was placed on the public web by AT&T
- No access controls were in place to protect the data
- The information accessed: a list of subscriber email addresses
- No criminal intent, as confirmed by the US Attorney
- The media was immediately contacted to alert the public of the danger

Despite these important facts, the DOJ is currently seeking a conviction for the following charges:

- Conspiracy to commit unauthorized access to a computer system (18 USC §1030)
- Aggravated identity theft (18 USC §1028A)

His arraignment on these charges is the 21st of July - next week. His immediate legal expenses are over $30,000 USD.

He urgently needs your help! Please donate now!

Donate to the legal defense fund
Another scary story

Super bad: First State set police on man who showed them how 770,000 accounts could be ripped off

Asher Moses
October 18, 2011

He confirmed that the vulnerability affected the firm's full customer database. It took me 30 seconds to write. They're probably of the opinion that I spent several hours putting together something to attack them.

Private security consultant Patrick Webster.

When Patrick Webster told First State Super he found a flaw exposing the personal details of its 770,000 members - including NSW Police officers, politicians and magistrates - he thought he was doing a good deed.

But before long, Webster, a private security consultant, received a phone call from the police informing him that his actions were illegal.
What would you do?
What would you do?

We decided that we should disclose the vulnerability.
Next question, how?

- Shoot Yelp an email?
- Contact CERT and let them deal with it?
- Anonymous public disclosure?
Next question, *how?*

<table>
<thead>
<tr>
<th></th>
<th>Email</th>
<th>CERT</th>
<th>Anonymous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get credit?</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Put ourselves on the line?</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Maintain control?</td>
<td>Y</td>
<td>Less so</td>
<td>N</td>
</tr>
<tr>
<td>Serve public?</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
So, what do we tell Yelp?

- Clearly, tell them about the bug.
- But we also want to make sure that they fix it quickly.
- Tell Yelp we give them $X$ days to fix bug, then we make a public disclosure.
Our strategy: Responsible Disclosure

- Collect data?

- Contact Yelp (not so easy)
  - No public email address, just a web form
    *We completed the form*
  - No phone number with a live human behind
    *We left voicemail*

- Contact NYT (very easy!)
  - We had contact details of David Streitfeld
    *Left him voicemail and emailed him*

- Wait...
About 24 hours later...

Both Yelp and NYT call back and both want to speak at 5pm EST

JB gives confidential ROFR to NYT...

...gets lukewarm response.

MM, GZ talk with Yelp team.

Thankful that we went to them first.

Not so happy that we want to go public but appreciative of our intentions.
Resolution

- Yelp takes down mobile site in minutes
- Fixes bug in hours
- Notifies us
- We suggest joint blog statement...
  ...they agree
Output Filtering Failure
Posted by Michael, VP of Engineering

About a month ago, we launched m.yelp.com specifically targeting iPhone, Android, and WebKit-based smart phones. Our engineering team pushes code live on average three times a day. Moving fast means we need to have sound engineering practices internally — such as code reviews by peers and automated testing tools such as unit-tests and static analysis — to catch mistakes before they happen. In this blog post we will detail a misstep that we made and the response that followed.

On the morning of October 27, 2011, we were alerted by a team of researchers at Harvard, Yale, and Boston University that they had found a servlet on m.yelp.com that could expose private user information. Working jointly with this team, our engineers gained a full understanding of the exposure. The leak allowed clients to see a JSON dictionary with user-specific fields, including email address, birth date, gender, and full names. No financially sensitive information was exposed.

Once we understood the risk to our users we immediately took the mobile site down. We resolved the issue within an hour, but kept the site down for 12 more while we double- and triple-checked for other issues (none were found). We analyzed the servlet’s access logs to see if anyone exploited the hole, but we did not find any evidence that user information had actually been collected. We also created an automated system to detect sensibly named fields (last_name, birthdate, etc.) being sent to clients. Following this work, we felt comfortable that the risk of a future exposure of this type had been mitigated; so we turned the mobile site back on.

The servlet at issue was using an ORM to retrieve information from the database. An ORM (object-to-relational mapper) system that automatically generated but wasn’t inspected by engineering used its database access method to expose user data to clients.
Our python logic for the biz details servlet looks like the following:

```python
def reviews(self, business_id):
    """unsafe version of reviews()"
    reviews = self.get_business_reviews(business_id)
    return json.write(
        # UNSAFE CODE: unsanitized data going to the
        # client DANGER!
        {'reviews': reviews}
    )
```

In the code above, the call `get_business_reviews()` under the covers returns data about reviews, including details of the users who wrote the reviews. This is where the private information on the user is requested.

To illustrate, here is an example JSON response from the offending servlet:

```json
{
    'reviews': [...
        {'user': {'birthdate': '1971,01,01',
            'display_name': 'Art G.',
            'first_name': 'Art',
            'gender': 'm',
            'last_name': 'Goldfin',
            ...}, ...
    }
}
```

*Note: birthdate, gender, and last_name are all private fields that shouldn’t have been returned in the JSON response.*

The above servlet function `reviews()` was rewritten to (approximately this):

```python
def reviews(self, business_id):
    """output filtered version of reviews()"
    reviews = self.get_business_reviews(business_id)
    # Filter reviews to contain only data necessary
    # for frontend
```
In the second implementation of the `reviews()` function you can see that the reviews object is transformed by `filter_for_frontend()` before being written into the JSON response. We’ve also modified `json.write()` to have a list of sensitive fields that will throw an exception if an engineer tries to pass a field with a restricted name like last_name, birthdate, etc.. With these protections in place, we’re well protected from this type of exposure in the future.

We’d like to thank researchers at Harvard, Yale, and Boston University:

- Georgios Zervas (a postdoc at Yale/Harvard)
- Michael Mitzenmacher (Harvard) (bio)
- John Byers (Boston University) (bio)

We appreciate the team’s diligence in finding and notifying us about this important problem; their thoughtful handling of a sensitive and tricky security situation is commendable. If you do find any security-related issues on Yelp, please send an email to security-abuse@yelp.com.

Yelp’s Engineering team is committed to excellence in engineering and data security; this incident was responded to with full force. Keeping user data safe is a top priority for Yelp, and we’ve taken concrete steps in response to this incident to make sure that it will not happen again.
Did Yelp have to disclose the leak?

California was the first state to enact a data breach notification law. (SB1386)

(e) For purposes of this section, "personal information" means an individual's first name or first initial and last name in combination with any one or more of the following data elements, when either the name or the data elements are not encrypted: (1) Social security number. (2) Driver's license number or California Identification Card number. (3) Account number, credit or debit card number, in combination with any required security code, access code, or password that would permit access to an individual's financial account. (f) For purposes of this section, "personal information" does not include publicly available information that is lawfully made available to the general public from federal, state, or local government records.
Thanks

- To Yelp’s team for their exemplary response.
- To all the researchers from BU, Harvard, and Berkeley who gave us valuable advice.