OF CONTRASEÑAS, סיסמאות AND 密码:
CHARACTER ENCODING ISSUES FOR WEB PASSWORDS

Joseph Bonneau
Rubin Xu
jcb82@cl.cam.ac.uk

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Web 2.0 Security & Privacy
San Francisco, CA, USA
May 24, 2012
How passwords get created

correct horse battery staple

0x636F7272656374...

UTF-8? ASCII?

ISO-8859-1?

Bonneau & Xu (University of Cambridge)  Character encoding & web passwords  May 24, 2012  2 / 26
Surprisingly little variation in (weak) passwords!

For top 1000 passwords, greatest efficiency loss is only 4.8 (fr/vi)
Research questions

- why is there so little language variation?
- how do non-English speakers choose passwords?
- how do websites fail for non-English characters?
- how do users cope with an English-dominated world?
Character encoding: a mercifully brief history

- ASCII (ca 1960)
  - English subset of Latin alphabet only
  - \( \approx 128 \) code points defined
  - high-order bit preserved for parity checking

- ASCII extensions
  - use high-order bits for extra characters
  - proprietary schemes (Windows code sheets)
  - 1988: ISO 8859 series (16 subsets)

- multi-byte encoding schemes
  - defined for Chinese, Japanese, Korean, and others
  - most use 2 bytes per character

- the dawn of the Internet
  - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
  - DNS: ASCII subset
Character encoding: a mercifully brief history

- **ASCII (ca 1960)**
  - English subset of Latin alphabet only
  - \( \approx 128 \) code points defined
  - high-order bit preserved for parity checking

- **ASCII extensions**
  - use high-order bits for extra characters
  - proprietary schemes (Windows code sheets)
  - 1988: ISO 8859 series (16 subsets)

- **Multi-byte encoding schemes**
  - defined for Chinese, Japanese, Korean, and others
  - most use 2 bytes per character

- **The dawn of the Internet**
  - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
  - DNS: ASCII subset
Character encoding: a mercifully brief history

- ASCII (ca 1960)
  - English subset of Latin alphabet only
  - \( \approx 128 \) code points defined
  - high-order bit preserved for parity checking

- ASCII extensions
  - use high-order bits for extra characters
  - proprietary schemes (Windows code sheets)
  - 1988: ISO 8859 series (16 subsets)

- Multi-byte encoding schemes
  - defined for Chinese, Japanese, Korean, and others
  - most use 2 bytes per character

- The dawn of the Internet
  - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
  - DNS: ASCII subset
Character encoding: a mercifully brief history

- **ASCII (ca 1960)**
  - English subset of Latin alphabet only
  - \( \approx 128 \) code points defined
  - high-order bit preserved for parity checking

- **ASCII extensions**
  - use high-order bits for extra characters
  - proprietary schemes (Windows code sheets)
  - 1988: ISO 8859 series (16 subsets)

- **multi-byte encoding schemes**
  - defined for Chinese, Japanese, Korean, and others
  - most use 2 bytes per character

- **the dawn of the Internet**
  - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
  - DNS: ASCII subset
Unicode and UTF-8

- **Unicode**
  - assigns a *code point* to every character in human writing systems
  - e.g. \( \ddot{a} \rightarrow 241 \)
  - **many** other features
  - over 1 M code points defined

- **UTF-8**
  - assigns code point to a variable number of bytes
  - e.g. 241 (\( \ddot{a} \)) → 0xc3b1
  - never allows 0x00 to appear outside code point 0
Unicode and UTF-8

- **Unicode**
  - assigns a *code point* to every character in human writing systems
  - e.g. ֱ → 241
  - **many** other features
  - over 1 M code points defined

- **UTF-8**
  - assigns code point to a variable number of bytes
  - e.g. 241 (ֱ) → 0xc3b1
  - never allows 0x00 to appear outside code point 0
Frequency of character encoding schemes today

<table>
<thead>
<tr>
<th>Encoding Scheme</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF-8</td>
<td>70.5%</td>
</tr>
<tr>
<td>ISO-8859-1</td>
<td>15.6%</td>
</tr>
<tr>
<td>GB2312</td>
<td>3.3%</td>
</tr>
<tr>
<td>Windows-1251</td>
<td>3.0%</td>
</tr>
<tr>
<td>Shift JIS</td>
<td>1.7%</td>
</tr>
<tr>
<td>Windows-1252</td>
<td>1.5%</td>
</tr>
<tr>
<td>GBK</td>
<td>0.9%</td>
</tr>
<tr>
<td>Windows-1256</td>
<td>0.6%</td>
</tr>
<tr>
<td>ISO-8859-2</td>
<td>0.5%</td>
</tr>
<tr>
<td>EUC-JP</td>
<td>0.4%</td>
</tr>
<tr>
<td>ISO-8859-15</td>
<td>0.4%</td>
</tr>
<tr>
<td>ISO-8859-9</td>
<td>0.3%</td>
</tr>
<tr>
<td>EUC-KR</td>
<td>0.2%</td>
</tr>
<tr>
<td>Windows-1250</td>
<td>0.2%</td>
</tr>
<tr>
<td>Windows-1254</td>
<td>0.2%</td>
</tr>
<tr>
<td>Big5</td>
<td>0.2%</td>
</tr>
<tr>
<td>Windows-874</td>
<td>0.1%</td>
</tr>
<tr>
<td>US-ASCII</td>
<td>0.1%</td>
</tr>
<tr>
<td>ISO-8859-7</td>
<td>0.1%</td>
</tr>
<tr>
<td>TIS-620</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

W3Techs.com, 23 May 2012
user types password
- managed by OS/browser
- code point and encoding known
user types password

- managed by OS/browser
- code point and encoding known
The password submission process-step 2

Password

.............

Retype Password

browser transcodes password to page encoding

- many places for page to specify
  - HTTP header, HTML header, form attribute
- replace with HTML numeric character reference
- undefined behavior if character entity reference also available!
  - IE: ñ → &ntilde;
  - FF/Chrome: ñ → &#241;
all characters outside of limited ASCII range are URL-encoded
  - also called percent encoding
double encoding possible if characters already transcoded
direct encoding possible for multipart/formdata form action
The password submission process—step 3

- All characters outside of limited ASCII range are URL-encoded, also called percent encoding.
- Double encoding possible if characters already transcoded.
- Direct encoding possible for multipart/formdata form action.

**Encoding of **ami (love)

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Submission</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB2312</td>
<td>%B0%AE</td>
<td>6</td>
</tr>
<tr>
<td>UTF-8</td>
<td>%E7%88%B1</td>
<td>9</td>
</tr>
<tr>
<td>ISO 8859-1</td>
<td>%26%2329233%3B</td>
<td>14</td>
</tr>
</tbody>
</table>
What sites need to do to support UTF-8 passwords
What sites need to do to support UTF-8 passwords

NOTHING
Part 1: what can go wrong

Test of 22 sites:

- **English/UTF-8**: Google, Facebook, Microsoft Live, Twitter, Wikipedia, Yahoo!
- **English/ISO-8859-1**: Amazon, DeviantArt, Gawker, IMDB, Walmart
- **Chinese/UTF-8**: CSDN, Renren, Kaixin001, Sina Weibo, Tianya, Mop, Gamer.com.tw
- **Chinese/GB2312**: QQ, Taobao, Baidu, Youku
Correctly supporting sites

Facebook, Twitter, Wikipedia, DeviantArt\textsuperscript{1}, CSDN, Renren, Kaixin001

\textsuperscript{1}Only non-UTF-8 site
Explicit ban on non-ASCII passwords

UTF-8: Google, Microsoft Live, Yahoo!, Sina Weibo, Tianya
other: Amazon, Taobao, Baidu, Youku
Forgotten Password
Please choose a new password.

Password: 

Password too long (max. 64 characters)

Confirm Password: 

Change

Need help?

IMDB, Walmart
Code point truncation

Weibo, QQ call charcodeat() in JavaScript
Weibo, QQ call `charCodeAt()` in JavaScript

```
aaaaaa
    =
ŁŁŁŁŁŁŁŁ
    =
ssssssss
    =
≁≁≁≁≁≁≁≁
    =
屁屁屁屁屁屁屁屁
    =
```

屁屁屁屁屁屁屁屁
DES-crypt() truncation

- Truncation to 8 characters per specification
- Gamer.com.tw: 我的中文得很好
- underlying bug discovered: `ACEMOMENT` accepted for `ÀLAPLAGE`
  - À → 192 → 0xC380
- present in BSD, PHP, PostgresSQL…
DES-crypt() truncation

- Truncation to 8 characters per specification
- Gamer.com.tw: 我的中 accepted for 我的中文得很好
- Underlying bug discovered: ÀCEMOMENT accepted for ÀLAPLAGE
  - À $\rightarrow$ 192 $\rightarrow$ 0xC380
- Present in BSD, PHP, PostgresSQL...
DES-crypt() truncation

- Truncation to 8 characters per specification
- Gamer.com.tw: 我的中文 accepted for 我的中文得很好
- underlying bug discovered: ÀCEMOMENT accepted for ÀLAPLAGE
  - À → 192 → 0xC380
- present in BSD, PHP, PostgresSQL…
Truncation to 8 characters per specification

Gamer.com.tw: 我的中文 accepted for 我的中文得很好

underlying bug discovered: ÀCEMOMENT accepted for ÀLAPLAGE
  À → 192 → 0xC380

present in BSD, PHP, PostgresSQL...
Down-conversion in `jcrypt()`

- buggy version of Java implementation of `bcrypt()`
- Gawker, Mop: ?????????? accepted for 我的中文得很好
Down-conversion in `jcrypt()`

- buggy version of Java implementation of `bcrypt()`
- Gawker, Mop: ????????? accepted for 我的中文得很好
Down-conversion in `jcrypt()`

- majority of sites don’t support UTF-8 passwords correctly
- many bugs left to find...
Case study: Chinese

Large leaked data sets now available
- 70yx-gaming site, 10 M users
- CSDN-forum site, 6 M users
(nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
<15% of users enter valid Pinyin passwords
45% numeric only, 90% contain some digits
  - compare to 15%, 45% for RockYou passwords
11% adjacent keyboard patterns
  - compare to 3% for RockYou passwords
Case study: Chinese

Large leaked data sets now available
- 70yx-gaming site, 10 M users
- CSDN-forum site, 6 M users
(nearly) all data in ASCII
- graphical Pinyin input disabled for password field

<15% of users enter valid Pinyin passwords
45% numeric only, 90% contain some digits
- compare to 15%, 45% for RockYou passwords
11% adjacent keyboard patterns
- compare to 3% for RockYou passwords
Case study: Chinese

- Large leaked data sets now available
  - 70yx-gaming site, 10 M users
  - CSDN-forum site, 6 M users
- (nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
- <15% of users enter valid Pinyin passwords
  - 45% numeric only, 90% contain some digits
    - compare to 15%, 45% for RockYou passwords
  - 11% adjacent keyboard patterns
    - compare to 3% for RockYou passwords
Large leaked data sets now available
- 70yx-gaming site, 10 M users
- CSDN-forum site, 6 M users

(nearly) all data in ASCII
- graphical Pinyin input disabled for password field

<15% of users enter valid Pinyin passwords
45% numeric only, 90% contain some digits
- compare to 15%, 45% for RockYou passwords

11% adjacent keyboard patterns
- compare to 3% for RockYou passwords
Case study: Chinese

- Large leaked data sets now available
  - 70yx-gaming site, 10 M users
  - CSDN-forum site, 6 M users
- (nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
- <15% of users enter valid Pinyin passwords
- 45% numeric only, 90% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 11% adjacent keyboard patterns
  - compare to 3% for RockYou passwords
Case study: Hebrew

Small leaked data set used
- Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords
Case study: Hebrew

Small leaked data set used
- Wondertree-spiritual site, 1K users

2.5% of passwords included Hebrew characters
- over 90% of usernames did...

- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords

- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords
Small leaked data set used
- Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords
Small leaked data set used
- Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords
Small leaked data set used
- Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords
Hebrew transliteration strategies

- Phonetic transliteration
  - אהבה → ahava (love)

- Keyboard transliteration
  - יִתְנָהָלָר → thigusnkcsu (There is no one else but him)
Case study: Spanish

- Spanish alphabet: mostly English/Latin
  - ñ considered a letter proper
  - á, é, í, ó, ú used to indicate stress

- Tens or hundreds of thousands of Spanish passwords at RockYou
  - impossible to compute due to cognates
## Spanish transliteration strategies

<table>
<thead>
<tr>
<th>password</th>
<th>meaning</th>
<th>proper</th>
<th>transliterated</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>contraseña</td>
<td>password</td>
<td>408</td>
<td>218</td>
<td>34.8%</td>
</tr>
<tr>
<td>muñeca</td>
<td>doll</td>
<td>197</td>
<td>354</td>
<td>64.2%</td>
</tr>
<tr>
<td>cariño</td>
<td>affection, dear</td>
<td>104</td>
<td>153</td>
<td>59.5%</td>
</tr>
<tr>
<td>pequeña</td>
<td>little (girl)</td>
<td>87</td>
<td>72</td>
<td>45.2%</td>
</tr>
<tr>
<td>teextraño</td>
<td>I miss you</td>
<td>65</td>
<td>27</td>
<td>29.3%</td>
</tr>
<tr>
<td>teamomamá</td>
<td>I love you mom</td>
<td>2</td>
<td>151</td>
<td>98.7%</td>
</tr>
<tr>
<td>código</td>
<td>code</td>
<td>5</td>
<td>110</td>
<td>95.7%</td>
</tr>
<tr>
<td>música</td>
<td>music</td>
<td>2</td>
<td>1447</td>
<td>99.9%</td>
</tr>
</tbody>
</table>
- ñ transliterated about half of the time
  - varies by password-strongly significant!
- stress accents almost always dropped
  - likely greater than 99% including examples like pájaro (bird)
Summary

- multilingual passwords are poorly supported
- users rarely make use when they are
- evidence that security is being harmed
Future directions

- can users enter Chinese passwords securely?
- how will we cope with mobile devices?
- more data needed to study linguistic trends
  - Russian, Arabic, Japanese, Korean, Greek, Hindi, Bengali, etc.
Thank you

jcb82@cl.cam.ac.uk

major thanks to:

- Noam Szpiro
- Elsa Monica Treviño Ramírez
- Claudia Diaz (Claudia Maria Díaz Martínez)