pASSWORD tyYPOS
and How to Correct Them Securely

R. Chatterjee, A. Athalye, D. Akhawe, A. Juels, T. Ristenpart

To typo is human; to tolerate, divine.
LOGIN

rahul

· · · · · · · · ·

LOGIN
Password-based authentication systems

Password459!

Salted, slow cryptographic hash

$H(\text{Password459!}) = \text{"a5idoiaU7p.."}$
Password-based authentication systems

Any typo is rejected

password459!
Password459!

Salted, slow cryptographic hash

$H(\text{Password459!}) = \text{“a5idoiaU7p..”}$
$H(\text{password459!}) = \text{“a5idoiaU7p..”}$

Typo-tolerant password checking
Allow registered password or typos of it
Facebook passwords are not case sensitive (update)

If you have characters in your Facebook password, there's a second password that you can log in to the social network with.

By Emil Protalinski for Friending Facebook | September 13, 2011 -- 12:26 GMT (05:26 PDT) | Topic: Security

Password459! pASSWORD459! password459!
We know little about password typos

Lots of work on usability of passwords...

[Ur et al. 2012], [Shay et al. 2012, 2014], [Mazurek et al. 2013],
[Bonneau, Schechter 2014] [Keith et al. 2007, 2009],
[Bard 2007], [Jakobsson et al. 2012]

... but nothing on typo-tolerant password checking.

1. How can we build a typo-tolerant systems?

2. How much would tolerating typos help users?

3. Does it endanger security?
Our work

We measure password typos at Dropbox and show they are a huge problem for both users and service providers.

We develop approaches to typo-tolerant checking, and show they improve utility with minimal security impact.

“Have your cake and eat it too”
How to do typo-tolerant password checking?
We focus on relaxed checkers

password459!
Password459!

H(password459!) = "a5idoiaU7p.."
H(PASSWORD459!) = "a5idoiaU7p.."
H(Password459!) = "a5idoiaU7p.."

Apply caps lock corrector
Apply first case flip corrector

No change in password hash database

Can we find a small but useful set of typo correctors?
MTurk password transcription study

100,000+ passwords typed by 4,300 workers

- Capslock 11%
- Flip first letter case 4.5%
- Add char. at end 4.6%
- Add char. at front 1%
- Last digit to symbol 0.2%
- Other 78.8%

Top 3 20%

% of all typos

MTurk password transcription study

100,000+ passwords typed by 4,300 workers

- Capslock 11%
- Flip first letter case 4.5%
- Add char. at end 4.6%
- Add char. at front 1%
- Last digit to symbol 0.2%
- Other 78.8%

Top 3 20%

% of all typos
Impact of top-3 typos in the real world

Instrumented production login of Dropbox to quantify typos

**NOTE:** We did not change authentication policy.

24 hour period:

- **3% of all users** failed to login because one of top 3 typos
- **20%** of users who made a typo would have saved at least 1 minute in logging into Dropbox if top 3 typos are corrected.

Allowing typos in password will add several person-months of login time every day.
Typo-tolerance will significantly enhance usability of passwords.

Can it be secure?
Threat #1: Server compromise

password459!
Password459!

No change in password hash database

H(password459!) = “a5idoiaU7p..”
H(PASSWORD459!) = “a5idoiaU7p..”
H(Password459!) = “a5idoiaU7p..”

No change in security in case of server compromise
Threat #2: Remote guessing attack

Web service should lock account after \( q \) wrong guesses.

Get 3 free checks with every query.

\[ \Rightarrow q \text{ queries result in } 3q \text{ free password guesses.} \]

\[ \Rightarrow \text{Previously, } q \text{ queries result in no free guesses} \]

\[ \Rightarrow \text{Attacker’s success increases by 300\%} \]

Apply **extra char. at end** corrector

Apply **flip corrector**

\[ H(\text{Password}) = \text{a5idoiaU7p..} \]

\[ H(\text{password}) ? = \text{“a5idoiaU7p..”} \]
Passwords are not uniformly distributed!

300% improvement, only if all checked passwords are equally probable.

BUT, humans do not chose random passwords.
Attack simulation using password leaks

Adversary knows:
- Distribution of passwords, and the set of correctors (Top 3)

**Exact checking**
Query most probable $q$ passwords

**Typo-tolerant checking**
Query $q$ passwords that maximizes success. Computed using greedy algo.

$q = 10$

![Bar chart showing success probability comparison between Exact checking and Typo-tolerant checking for two different values of $q$. The chart shows a comparison between $q = 10$ for both types of checking methods.](chart.png)
Security-sensitive typo correction

Don’t check a correction if the resulting password is too popular.

Password

Free Correction Theorem
For any non-uniform password distribution, set of correctors, and adversarial query budget \( q \), there exists a typo correction scheme that corrects typos with no degradation in security.
Security of checkers with filtering

Correct typo ensuring that total probability of all checked password is less than $\Pr[pw_q]$.

Estimated password distribution with rockyou

Change in success: 0.02%
Typo-tolerant checking can enhance users’ experience for essentially no degradation in security.
1. Introduce typo-tolerant password checkers
   • Compatible with existing password databases, easy to deploy

2. Study password typos empirically
   • 3% of users fail to login due to correctable, top-3 typos

3. Analyze security of typo-tolerant checkers
   • “Free” correction theorem (In theory)
   • With heuristic, works in practice too

GitHub/rchatterjee/mistypography

Thanks!
rahul@cs.cornell.edu