IEEE S&P 2021 Home

Help · Sign in · Sign out

#232 DALock: Password Distribution-Aware Throttling

Your submissions (All) Search

PC conflicts

Aniket Kate
Antonio Bianchi
Brendan Saltaformaggio
Mathias Payer
Nicolas Christin

Rejected

Submission (3.1MB) 4 Dec 2020 2:51:07am EST · 07ab1c27

▶ Abstract

Large-scale online password guessing attacks are widespread and continuously qualified as one of the top cyber-security risks. The common method for mitigating the risk of online cracking is to lock out the user after a fixed number (K) of

[more]

▶ Authors (blind)

J. Blocki, W. Zhang [details]

Prior Reviews (109kB)

Submission can be reviewed by Shadow PC

Yes

▶ Topics

To edit this submission, sign in using your email and password.

	OveMer	RevExp	RevCon	Rev
Review #232A	2	4	3	2
Review #232B	3	3	3	1
Review #232C	2	3	3	2
Review #232D	2	4	2	2

Reviews in plain text

Review #232A

Overall merit

2. Weak reject - The paper has flaws, but I will

Reviewer expertise

4. Expert - Historically an area of primary not argue against it.

focus, or an area I have done recent, significant work in.

Reviewer confidence

3. High

Paper summary

The paper proposes a account lock out threshold based on the probability of the passwords submitted. (Traditionally this is done based on just the number of incorrect password submission, count-based locking.) The authors show that the optimal attack against such an authentication server is NP-hard; and heuristic attacks provide significantly less advantage than against purely count-based locking.

Strengths

+ Distribution-aware locking was proposed earlier [17] (though in a slightly different form), however in this paper, authors analyze the attack against a distribution-aware locking mechanism.

Weaknesses

- The evaluation of the system is done in simulated data, which is not very convincing.
- The contribution on top of the prior work is quite small.
- the choice for simulation is not very clear

Detailed comments for author

The paper is well written, and addresses an interesting problem. The problem this paper addresses is very important: we need a more usable lockout mechanism. However, two biggest concern I have about the paper are the following:
1. The paper has small delta over the prior work, StopGuessing, by Schechter et al. [17]. The novel contribution in this paper is the evaluation of security. The evaluation however, is not done extensively. The security simulations are not clear why they are done in this way. For example, I would be interested in knowing the difference in the efficacy of DLock in preventing targeted and untargeted attacks. Or, how will the attack perform for different attacker's estimation of the password distribution (for example, using PCFG or NN)?

2. The evaluation is done on simulated data. Although the authors put a lot of thought into the simulation process, it is still unclear how this new mechanism will playout in practice. Especially how are the different thresholds chosen for security simulation is not explained.

"$$\Psi_u$$ never resets, unlike the consecutive strike parameter $$K_u$$". This will lead to straightforward DoS attack against users? Also, I am not quite sure what is the intuition behind having two different counters, especially the one that never resets. It would be helpful to have some explanation of this construction.

# Other editorial comments
Usability Figures are hard to read, especially, Fig 3, 4, and 5. The text fonts in Fig 4 & 5 are too small to read.

The naming convention for different experiment setups seems quite hard to understand. It will be good to create and explain the naming structure and follow it strictly.

Typo:
1. Table 1: The last row $$K_U$$ should be $$K_u$$.
2. "Both StopGuessing and DALock exploit
differences between the distribution of user passwords and sses."
3. "In an independent line of work, Tian et al. [17] developed" --> This is Schecter et al.

Revision

2. No

Review #232B

Overall merit

3. Weak accept - While flawed, the paper has merit and we should consider accepting it.

Reviewer expertise

3. Knowledgeable - I know the area well (key related work is quite familiar to me).

Reviewer confidence

3. High

Paper summary

This paper proposes a dynamic password guessing throttling mechanism that takes into account the popularity of a password input in determining when to cut off additional login attempts, as opposed to fixed counting schemes that allow 3 or 10 failures before lockout. In order to securely house a distribution of popular passwords, the authors also propose a differentially private password counter, comparing its efficacy against popular password strength meters such as ZXCVBN and Neural Network models. The authors show that in simulations, 5.8% of accounts can be compromised in a 3-strike mechanism, compared to 1.4% with the proposed system (or 4.6% with ZXCVBN). This can be further tuned by prohibiting the top N most popular passwords, reducing the compromise rate to 0.08%, while also limiting lockout to 0.08% (versus 4% for a 3-strike system).

Strengths

+ Paper is well-written and thoroughly evaluated. Authors demonstrate the security and usability tradeoffs of DALock, including how password blocklists improve security and usability (apart from the initial usability cost of blocked passwords).

+ Comparison of differentially private password storage mechanism versus popular password strength meters is novel and helps to demonstrate that DALock can be deployed without having to implement a new storage system, though with some decrease in performance.

+ Simulations suggest real-world usability and security gains over 3-strike policies that would improve account lockout.

Weaknesses

- Nominal variations from StopGuessing significantly reduces the novelty of the work. The authors do not sufficiently explain the complexity that arises from targeting accounts over IP addresses. (If StopGuessing was re-targeted to accounts, would the assumptions it makes somehow fail, where DALock succeeds?)

- Comparison with password strength meters suggests that absent banning passwords, HashCat and ZXCVBN actually provide better usability for users (though result in more compromises). This undercuts in part the contribution of a privacy preserving password storage mechanism.

Detailed comments for author

Overall, this paper strikes on an interesting topic of how to improve the usability of password throttling mechanisms. Absent the recent publication of StopGuessing, this work would have a clear statement of novelty and technical contribution. Unfortunately, DALock bares many resemblances to StopGuessing: (1) a privacy-preserving storage mechanism for passwords; (2) a dynamic penalty function for failed login attempts; and (3) the same evaluation criteria, illustrating the same problem is being addressed (though this is less critical).

While the authors point out that StopGuessing is IP-based and DALock account-based, the underlying techniques are nominally identical. Likewise, the difference between modeling password guessing strategies (StopGuessing) versus password distributions (DALock) is unlikely to yield a practical difference in performance, given the tools for guessing are built from password distributions. The authors need to make a better case for why, were StopGuessing to be applied to usernames rather than IP addresses, it would degrade in performance. (Or alternatively, that large IP addresses like mobile gateways exist, which reduces the protection provided by StopGuessing if attackers rely on proxies.)

While there are similarities, the ability for off-the-shelf password strength meters like ZXCVBN to serve as a drop-in for a differentially private storage system *is* a novel insight, and likely would help the deployability of this scheme. On this front, it would be helpful to have an additional experiment or two. Some suggestions:

- As stands, Fig 4 and Fig 5 make it hard to evaluate the tradeoffs of one password strength meter versus another. For example, HashCat has the best usability, but worse security bounds than 0.1-CS-all. Is it possible to fix the usability bound, to evaluate exactly how much security one provides over the other?

- Another potential artifact of using popular breaches is that some were part of the construction of the password strength meters evaluated, while some are the construction of the password strength meters evaluated, while some are

not. Differentiating these scenarios would better help articulate how inaccurate modeling of a password distribution can reduce the security (and potentially usability) bounds in simulations. This nuance would also help determine when a differentially private storage mechanism becomes more critical.

Revision

1. Yes

Requested changes

- Better explain contribution compared to StopGuessing, either in the problem construction, evaluation, or a fundamental failure in translating StopGuessing from IP addresses to usernames. Absent this framing and highlighting the key novel contributions, this may be an unfortunate case of getting beaten at the races.

Review #232C

Overall merit

2. Weak reject - The paper has flaws, but I will not argue against it.

Reviewer expertise

3. Knowledgeable - I know the area well (key related work is quite familiar to me).

Reviewer confidence

3. High

Paper summary

This paper proposes a password rate-limiting policy which takes into account the popularity of the passwords being guessed, allowing more guesses of unpopular passwords

Strengths

Overall idea of distribution-aware rate-limiting is good

Discussion raises a number of interesting points

Weaknesses

This paper's evaluation model is extremely complex

The overall model may not match reality and in fact be much less realistic than prior work

Detailed comments for author

The basic idea of taking into account the popularity of passwords being guessed is a good one, though it is not really novel, being proposed by Tian/Schechter/Herley. In the introduction this paper draws a distinction in that this work focuses on locking out specific user accounts rather than IP addresses, but there's a reason that Tian et al. focused on the former: it's how rate-limiting actually works in practice at modern large-scale services (which dropped simple "k-strikes-and-you're-out" policies years ago).

The reason for this guess is that it is so easy to find large numbers of usernames (or registered emails) that it is not sufficient to allow an untargeted attacker to have a few guesses at each. IP addresses (or subnets etc.) are a more scarce resource for guessers than are known user accounts to attack. There are also positive factors associated with "known good" IP addresses where users frequently log in to limit the disruption of these policies to legitimate users.

Thus I fundamentally think the model in this paper is less realistic than the Tian et al. model which is how large services actually operate. Perhaps the proposal here is simpler to maintain than an IP-based system, but it is already quite complex and requires a large number of users to be effective. Beyond this difference of model the difference in contribution here is fairly slim to the prior paper.

Other comments:

Section III could be spaced out more throughout the paper, e.g. introduce the count-min sketch after explaining the context where it is needed.

One method to help manage user mistakes is to not add to the count for repeated wrong guesses of the same wrong value, that could be a nice component to add.

The discussion of the choice of $$\epsilon$$ is a bit misleading, comparing to other papers when the risks are totally different. For example the Blocki et al. paper only released counts of different passwords by rank, not plaintext passwords as this paper proposes. RAPPOR is used for less sensitive information. It's a general problem with differential privacy that it's very hard to translate parameter choices into a meaningful real-world property.

Revision

2. No

Review #232D

Overall merit

2. Weak reject - The paper has flaws, but I will not argue against it.

Reviewer expertise

4. Expert - Historically an area of primary focus, or an area I have done recent, significant work in.

Reviewer confidence

2. Medium

Paper summary

The paper considers a throttling mechanism to slow down (online) password guessing attacks. While "classical" methods count the number of login attempts with wrong password and block after, e.g., 10 wrong attempts, the presented approach takes the distribution into account. It accumulates the probability mass of wrong login attempts and blocks the account if enough probability mass has been spent on that account, presumably by an attacker.

The paper extensively evaluates this approach for several thresholds and password distribution approximations, and shows that it decreases accidental lockout for users and increases account security.

Strengths

- neat idea
- thorough evaluations

Weaknesses

- evaluation purely theoretical, unclear impact on practice

Detailed comments for author

The paper is overall written well, for some minor comments see below.

The analysis is extensive, but very mathematical. It uses complex models of user behavior which may or may not be accurate.

For example, the model for users making typos when entering passwords is modeled based on data from [14]. However, [14] measures password typos in a pretty artificial setting, requesting MTurkers to type a list of passwords that were sampled from RockYou. This seems to provide a weak model for real-world password typos. Specifically, and potentially most problematic for the current work, is that it will not show differences between users; I would expect that there are users with higher error rates (e.g., entering passwords faster, less frequently, or being less experienced with typing), and users with lower error rates (e.g., due to technical help such as password managers) etc. I couldn't find data on this in [13,14].

The assumed attacker is an ideal attacker with full knowledge about the system, which seems a plausible (yet pessimistic) assumption. The discussion on the Password Knapsack Problem, however, doesn't contribute much to the paper. It's a worst-case assumption and thus quite irrelevant. Also, as the paper mentions, approximation algorithms are highly successful in solving the problem.

It assumes a password distribution independent of the user (which is incorrect in practice). How does that influence the results?

The idea itself is neat, but relatively straight-forward. I'm missing an analysis more geared towards practice, testing with real users and observing potential acceptability issues with it.

The discussion is pretty hard to follow. Part of the reason is that the graphs containing the main results (Fig 4 and 5) are plainly unreadable when printed, and still hard to read at 300% magnification on a monitor, hindering in following the discussion. Furthermore, these graphs contain seemingly random combinations of parameters, making it harder to interpret results.

Minor stuff:

- lots of missing spaces before references and round brakets
- weird formatting of Definitions 1 and 2
- weird breaks in formulas on page 4

- page 3 left middle: DALock exploit*s*
- page 3 right middle model *by(Blocki
- page 10 right bottom: in Appendix *?* see Figure 7

Revision

2. No

HotCRP