Jones, K. & Salako, K. (2013). Modeling Security Policy and the Effect for End-Users. Paper presented at the HCI International 2013: 15th International Conference on Human-Computer Interaction, 21 - 26 Jul 2013, Las Vegas, Nevada, US.

Modeling Security Policy and the Effect for End-Users

Kevin D. Jones and Kizito Salako

City University London
kevin.jones.1@city.ac.uk, kizito@csr.city.ac.uk

Abstract. Many "good practices" in computer security are based on assumptions and local evidence that do not generalize. There are few quantifiable methods of establishing or refuting the validity of these practices from a user perspective. We propose a formal model of security policies that allows us to evaluate the claimed benefits to the user of the system quantitatively. We illustrate the use of the model by looking at a security policy we all live with daily: The Password Policy.

1 Introduction

There are many myths in the field of computer security that have a daily effect on the users of systems. Since there is little in the way of quantitative evaluation to support or disprove claims about improved security, assumptions are taken as fact, and we accept discomfort in return for supposed benefit. In most other parts of the security space, we have strong systems for reasoning about the strength of the system, but this has not been the case for user level policy. We know that many security vulnerabilities are due to users not complying with stated policy, due to a lack of understanding of the value of that policy. A mathematical model of the security system allows a firm underpinning for discourse with users on the motivation for the policy.

We use a password policy as an example since it illustrates the concept. To address questions about the efficacy of such policies, we develop a probabilistic model which captures the way these policies constrain how people choose passwords, and what this means for system security. Our primary aims with this model are twofold: 1) clarifying and formalising concepts used when discussing password policies and the level of security they engender (some concepts we clarify include *password strength* and the *difficulty* an attacker might have in subverting password security), and 2) exploring new applications of probabilistic modelling approaches already applied in other contexts, with the aim of gaining insight into security evaluation.

The rest of this paper is structured as follows. We begin in Section 2 with details of how we model a password-based security system that is both subject to password policies and open to attack from adversaries. Section 3 develops this further with a discussion of what *password strength* means in our model. In Section 4, user implications of our model are explored.

2 Modelling Password Choices and Attacker Actions

In a *password-based security system*, access to computing resources are controlled as follows: each user of the system has preassigned access rights to the resources – we refer to these access rights as a *user–profile* – and anyone can gain the access granted in a given user–profile by submitting the password for the user–profile to the system for authentication. In practice, the system may be viewed as a collection of software tools that: 1) maintain a database of the one-to-one correspondence between a user-profile and its related password, and 2) grant an entity access to a user profile if and only if the entity submits the password associated with the user-profile.

Naturally, there is a limited amount of computer memory available to the system for storing each password chosen by its users, where these passwords are comprised of symbols from a finite character set. Consequently, the set of all possible passwords, $\mathcal{P}$, is finite and completely defined by the password security system.

A user of the password security system chooses a password, say π, from the typically large set $\mathcal{P}$ according to some probability distribution.

The system is open to attack by an attacker. Here, an ***attacker*** is any entity that seeks to subvert the password security system by correctly guessing the password for a legitimate user, submitting this guess to the password security system and, thereby, gaining access to those computer resources granted in a legitimate user–profile. An ***attack*** is the choice and submission of a password to the security system by an attacker. We refer to a collection of attacks by an attacker as an ***attack campaign***. In conducting an attack campaign, the attacker chooses a sequence of passwords from $\mathcal{P}$ at random. Suppose the number of attacks (that is, the length of such a sequence) is N. Then the sequence of passwords, which we may refer to as σ, is an ordered N*–tuple* of passwords, say $(\pi_1, \ldots, \pi_N)$. That is, an attack campaign is characterised by some sequence of passwords σ chosen by an attacker so that:

$$\sigma = (\pi_1, \ldots, \pi_N) \in \underbrace{\mathcal{P} \times \mathcal{P} \times \cdots \times \mathcal{P}}_{N \text{ times}},$$

We accept the view that an identical amount of time and effort is expended by the attacker in carrying out each attack in an attack campaign. After all, attackers have no notion of how much closer they have gotten to correctly guessing a password after a succession of failed attacks, and the act of guessing and submitting a password to the system does not significantly change from attack to attack. Formally, therefore, if an attack campaign occurs over regular time-intervals in calendar time, then the length N of an attack campaign is a measure of the time and effort spent by the attacker. For the sake of brevity we shall write the cartesian product above, $\mathcal{P} \times \mathcal{P} \times \cdots \times \mathcal{P}$, as $\mathcal{P}^N$.

An attack is deterministic in its outcome: a correctly guessed password by an attacker compromises the system, while an incorrect guess will not. Therefore, for each pair of user-password π and choice of password an attacker submits

to the security system π_i, the indicator function $\nu(\ ,\ )$ tells us if an attack is successful; it is defined as follows

$$\nu(\pi, \pi_i) = \begin{cases} 1, & \text{if } \pi = \pi_i \\ \\ 0, & \text{otherwise} \end{cases}$$

A ***user policy*** is the set of rules and constraints, imposed on a user of a password-based security system, which limit both how the user chooses a password and the length of time for which such a user-password is valid. So, for instance, a password policy that gives guidance to a legitimate user about how to choose a "*strong*" password would be part of the user policy for any user that follows such guidance. In effect, a user-policy defines a partition of $\mathcal{P}$ into two disjoint subsets – those passwords that a user may choose from, and those passwords that the user will not choose from. A user policy is partly the result of system imposed rules for password choices – imposed on the user to restrict which passwords she can choose – and system imposed time limits for the validity of a chosen password. In addition, user policy is also the result of the preferences a user has about how to construct a password, such as using a combination of letters from the lyrics of a favourite song or the sum of family members birthdays.

In the same vein, there exist (possibly self-imposed) constraints on an attacker which determine both how the attacker chooses passwords and how long the attack campaign may last for. We loosely refer to these constraints as an ***attacker policy***. An attacker-policy partitions $\mathcal{P}$ into two disjoint subsets of passwords: those that an attacker may choose from when conducting a campaign of attacks, and those the attacker will not choose from. The partition arises naturally because an attacker policy is partly defined by the attacker's beliefs and preferences: for instance, he might take the view that certain passwords will never be chosen by the user. This limits the potential passwords considered by the attacker. However, this alone may be insufficient for the attacker's needs, given that the typical number of attacks an attacker can carry out is exceedingly small compared with the number of possible passwords a user may have choose from, so that the attacker's beliefs still result in a relatively large subset of passwords to be considered by the attacker. Compound this limitation with the fact that for many practical systems there is a finite amount of time for which a user's password might be valid – so, an attacker has a finite number of attempts to guess the user's password – and we see that an attacker would seek to maximize his probability of correctly guessing a user's password by focusing his attacks only on those passwords that he deems a user most likely to choose. In summary, the attacker policy defines the subset of passwords an attacker might try, but the attacker still needs to optimize which of these passwords he should try – an optimization problem that results in a probability distribution for which passwords the attacker chooses, as we shall see shortly.

A user of the system is required to choose a password, at random, in accordance with the user policy. For a given user and user policy, let Π be the

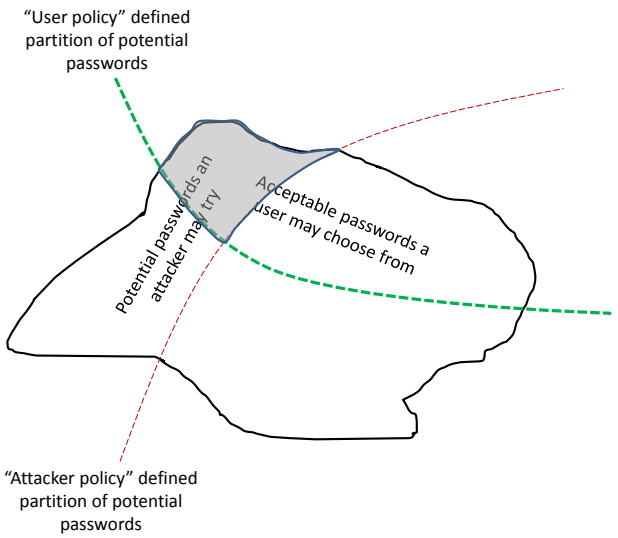

Fig. 1. *The overlap of those passwords that may be chosen by a user and those that may be chosen as part of an attack campaign.*

random variable that models the random choice of a password by the user. Then, $P(\Pi = \pi)$ is the probability that a given password π is chosen by the user from $\mathcal{P}$. This defines the *distribution of user-password choice*, an example of which is depicted in Fig. 2.

On the other hand, an attacker typically has a limited number of tries to guess a password, so he chooses each password in an attack campaign according to his attacker policy and some probability distribution. For a given attacker and attacker policy, let Π_i be the random variable that models the *ith* random choice of password made by the attacker for an attack campaign. Then, the probability that a given potential password, π_i, is used in an attack is $P(\Pi_i = \pi_i)$. This is a *distribution of attacker-password choice*, an example of which is depicted in Fig. 3. Actually, more interesting is the probability that a given sequence of potential passwords, say $\sigma = (\pi_1, \ldots, \pi_N)$, is chosen by the attacker to use in a campaign. In general, this is given by the discrete joint probability distribution of the random vector $\Sigma = (\Pi_1, \ldots, \Pi_N)$,

$$P(\Sigma = \sigma) = P(\Pi_1 = \pi_1, \ldots, \Pi_N = \pi_N) \; . \tag{1}$$

This defines the *distribution of the attacker's attack-campaign choice.*

For the campaign to be successful it is sufficient that at least one of the passwords tried by the attacker is a correct guess; where exactly in the sequence such a correct guess occurs is unimportant for now. In particular, the probability that an attack-campaign successfully compromises the system by guessing the legitimate password π being used by a given user

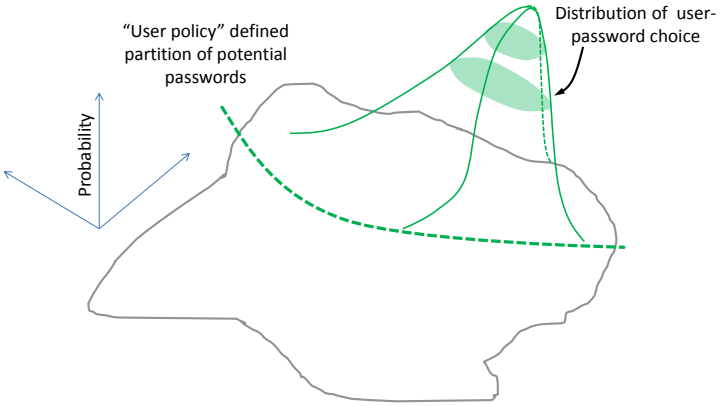

Fig. 2. *A distribution that defines, for each password, the probability that the password is chosen by the user. The random variable is defined on the space of passwords $\mathcal{P}$, and those passwords that a user will never choose from (according to the user-policy) will have zero probability associated with them.*

is given as:

$$P\Big(\nu(\pi,\Sigma)=1\Big)=\mathbb{E}\Big[\nu(\pi,\Sigma)\Big]=\sum_{\sigma\in\mathcal{P}^N}\nu(\pi,\sigma)P\left(\Sigma=\sigma\right)\,. \tag{2}$$

In practice, the precise form of Eq. (2) is influenced by different factors. For instance, many password-based systems have a maximum placed on the number of consecutive failed authentication attempts on a user profile. When this number is reached the user profile is made unavailable for authentication for a period of time. This limits the length of an attack-campaign carried out by an attacker. Another example is how some websites use so-called "Captchas" to determine if a human being is attempting to access the material they contain, as these also hinder an attackers ability to perform a brute-force attack, thus significantly reducing the length of an attack-campaign and increasing the effort required by an attacker.

While our focus is on passwords, the model does not ignore the utility of a *username* which, in conjunction with a password, may be viewed as a unique identifier for a legitimate user of the security system.

The model allows for an asymmetry between a system administrator and an attacker in what each can learn from a failed login attempt. Usually, failed attempted logins are not informative in indicating to an attacker just how "close"

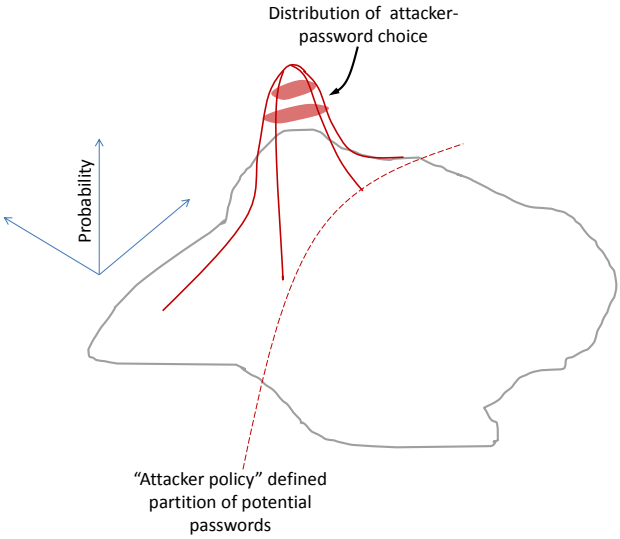

Fig. 3. *A distribution that defines, for each password, the probability that the password is chosen by an attacker in an attack. The random variable is defined on the space of passwords $\mathcal{P}$, and those passwords that an attacker will never choose from (according to the attacker-policy) will have zero probability associated with them.*

the attacker's wrongly guessed password is to the true password. On the other hand, a system administrator can compare the incorrect password that was submitted with the actual password to determine if the two passwords differ by an insignificant number of characters.

For clarity it has been indicated that a single attacker carries out an attack-campaign. However, this simplification is not necessary and the model generalizes easily to cater for multiple attackers.

3 Password Simplicity

When is a password chosen by a user a *simple password*? A user-password could be simple for any of the following reasons:

1. The password is chosen from a set of related passwords, where an attacker armed with some knowledge about this set can deduce what the underlying relationship between the passwords is. For instance, a symmetrical relationship between the passwords (e.g. they all comprise of some permutation of the same vowels, consonants and symbols), or a set of simple transformation rules that gives another password in the set once one of the passwords in the set is known (e.g. a rule that replaces all 's' characters with the '$' symbol).

2. The password is one of many common passwords known to attackers (e.g. common phrases such as "*password*" or "*123456*").

A user choosing simple passwords potentially increases the probability of an attacker correctly guessing the user's password. We can make this notion precise as follows. For an attacker carrying out a randomly chosen attack-campaign aimed at accessing a given user-profile, we define *the simplicity of a given user-password* for the attacker as the probability that the attacker correctly guesses the password in a random attack-campaign, if the password is the one associated with the user-profile. That is, for a user-password π and a random variable Σ that models an attacker's choice of attacks in an attack-campaign (where the length of the campaign is N), the simplicity of π for the attacker is computed by Eq. (2) Let us consider a particular case of this formula. Suppose further that the attacks

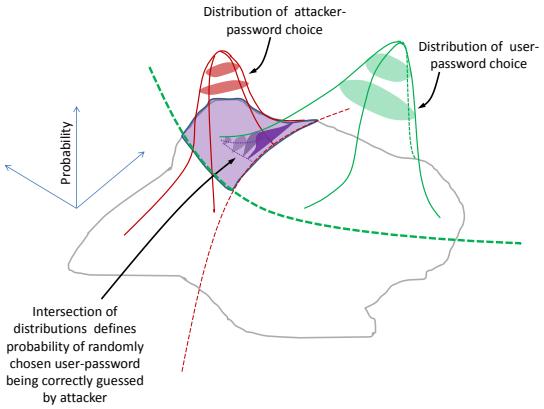

Fig. 4. *The overlap of the distributions in Fig.'s 2 and 3 defines the probability that a randomly chosen user-password is correctly guessed by an attacker. A simple password would be one for which such an overlap implies an attacker is likely to guess the password.*

in an attack-campaign are independently and identically distributed password choices made by the attacker, where Π_a is the random variable that models the choice of an attack by an attacker. Then, the simplicity of the password π for this attacker is:

$$\begin{aligned}
\mathbb{E}\left[\nu\left(\pi,\Sigma\right)\right] &= \sum_{\sigma\in\mathcal{P}^N} \nu\left(\pi,\sigma\right) P\left(\Sigma=\sigma\right) \\
&= \sum_{(\pi_1,\ldots,\pi_N)\in\mathcal{P}^N} \left(1-\prod_{i=1}^{N}\left(1-\nu\left(\pi,\pi_i\right)\right)\right) P\left(\Pi_a=\pi_1,\ldots,\Pi_a=\pi_N\right) \\
&= \sum_{\pi_N\in\mathcal{P}} \cdots \sum_{\pi_1\in\mathcal{P}} \left(1-\prod_{i=1}^{N}\left(1-\nu\left(\pi,\pi_i\right)\right)\right) P\left(\Pi_a=\pi_1\right)\ldots P\left(\Pi_a=\pi_N\right)
\end{aligned} \tag{3}$$

Note that we can make the following expansion:

$$1-\prod_{i=1}^{N}\left(1-\nu\left(\pi,\pi_i\right)\right) \;=\; \sum_{i=1}^{N}\nu\left(\pi,\pi_i\right) \;-\; \sum_{i_1<i_2}^{N}\nu\left(\pi,\pi_{i_1}\right)\nu\left(\pi,\pi_{i_2}\right) \;+\; \ldots$$
$$+\;(-1)^{N}\sum_{i_1<\ldots<i_{N-1}}^{N}\prod_{j=1}^{N-1}\nu\left(\pi,\pi_{i_j}\right) \;+\; (-1)^{N+1}\prod_{i=1}^{N}\nu\left(\pi,\pi_i\right).$$

Therefore, upon using this expansion in Eq. (3), the linearity of mathematical expectation $\mathbb{E}\left[\bullet\right]$, and the notation $\phi_a\left(\pi\right) := \mathbb{E}\left[\nu\left(\pi,\Pi_a\right)\right]$, we have:

$$\begin{aligned}
&\mathbb{E}\left[\nu\left(\pi,\Sigma\right)\right]\\
&=\binom{N}{1}\mathbb{E}\left[\nu\left(\pi,\Pi_a\right)\right] \;-\; \binom{N}{2}\mathbb{E}\left[\nu\left(\pi,\Pi_a\right)\right]^2 \;+\; \ldots \;+\; (-1)^{N+1}\,\mathbb{E}\left[\nu\left(\pi,\Pi_a\right)\right]^N\\
&=\binom{N}{1}\phi_a\left(\pi\right) \;-\; \binom{N}{2}\left(\phi_a\left(\pi\right)\right)^2 \;+\; \ldots \;+\; (-1)^{N+1}\left(\phi_a\left(\pi\right)\right)^N\\
&=1-\Big(1-\phi_a\left(\pi\right)\Big)^N
\end{aligned} \tag{4}$$

as the expression for the simplicity of the password π for the attacker. By examining this form of password simplicity we make the following observations:

1. *Password simplicity is affected by the length of the attack-campaign*: for $0<\phi_a\left(\pi\right)<1$ the longer the attack-campaign the more likely the attacker is in being successful. This follows from the function $1-\Big(1-\phi_a\left(\pi\right)\Big)^N$ being a strictly monotonically increasing function of N, and the following limit.

$$1-\Big(1-\phi_a\left(\pi\right)\Big)^N \to 1 \qquad \text{as} \qquad N\to\infty\,.$$

   Consequently, if the security system limits the number of tries an attacker has to validate password guesses this, in turn, limits the length of an attacker's attack-campaign. Many online password-based security systems implement this sort of limitation, but a significant number still do not.
2. *The probability of an attack succeeding affects password simplicity*: the more likely an attacker is at guessing the password in a single attack, the more likely the attacker is at being successful in a random attack-campaign. For, if $1\geq x\geq\phi_a\left(\pi\right)\geq 0$, then the following inequality holds.

$$1-\Big(1-x\Big)^N \geq 1-\Big(1-\phi_a\left(\pi\right)\Big)^N.$$

The probability $\phi_a(\pi)$ takes into account the knowledge an attacker has, as well as the method by which the attacker chooses a password to be used in an attack.

3. *Password Strength*: Given the distribution for an attacker's password-choice, the Shannon-entropy

$$-N \log_2 [1 - \phi_a(\pi)] .$$

By definition, each password π chosen by a user has an associated simplicity, $\mathbb{E}[\nu(\pi, \Sigma)]$, for a given attacker. Consequently, there is a natural notion of a *simplicity function* for the attacker: for each password π, the simplicity function $\theta(\pi)$ gives the probability that the attacker successfully guesses the password in a random attack-campaign.

$$\theta(\pi) := \mathbb{E}[\nu(\pi, \Sigma)] \tag{5}$$

The simplicity function makes apparent the idea that some passwords are easier to guess than others for a given attacker. There is also a dual relationship: a given password chosen by a user could be easy for some attackers to guess but difficult for other attackers. So, different attackers have different simplicities.

Since $\theta(\pi)$ is defined on $\mathcal{P}$, the user-policy and attacker-policy have the following implications for password simplicity:

1. Those passwords that a user will not choose from will be of no use to the attacker, even if those passwords are very simple for an attacker to guess had they been chosen by the user.
2. Those passwords that an attacker will not choose from have associated simplicity of 0, and hence are the best passwords for a user to choose from.
3. Simple passwords that a user may choose from hold the most potential for the attacker being successful.

4 Password Policy and its Effects on Users

Password policies have a clear effect on users. For example, some password-based security systems implement a user-policy that puts a time limit on the validity of a password chosen by a user. In effect, for a user to continue to access computing resources, the user has to choose a new password after a fixed amount of calender time has elapsed. Rationale for such a policy include:

1. Limit the time an attacker has to penetrate the system
2. Limiting the time a successful attacker has "unauthorised" access to computing resources.

We all live with such expiration policies, and many of us complain about having to change our passwords "too often". We are generally told that we have to live with this for "added security" but there is rarely any quantitative evidence that security is added. The models we have proposed can be instantiated

for any given situation to establish whether or not there is value in such a policy. Unsurprisingly, whether there is additional security or not depends on the assumptions made about the relationship between the user and attacker profiles. To illustrate this point consider the following counterexample to claims of password expiration being beneficial. Let Φ_i be the random variable $(1 - \phi_a(\Pi_i))^N$ that models an attacker failing to guess a user password Π_i in N independent consecutive attacks (we defined the function $\phi_a(\pi)$ when proving Eq. (4)). Suppose that upon being required to renew her password a user is increasingly more likely to choose a simpler password with each new choice. So formally, given $m-1$ successive password choices $\pi_{m-1}, \ldots, \pi_1$ made by a user, the user being more likely to choose a weak password in her m^{th} choice, compared with her 1^{st} choice, is modelled by the following inequality.

$$\mathbb{E}[\Phi_m | \pi_{m-1}, \ldots, \pi_1] \ll \mathbb{E}[\Phi_1] .$$

The rationale for this inequality is that in such a situation an attacker can expect the passwords chosen to become simpler. As a consequence of this, the **probability that an attacker will fail to guess a users password after m password choices by the user** is much less than it would be if the user chose only one password over the same time period. That is

$$\mathbb{E}[\Phi_m \Phi_m \ldots \Phi_2 \Phi_1] \ll (\mathbb{E}[\Phi_1])^m \leq \mathbb{E}[{\Phi_1}^m] .$$

5 Conclusion

Much of what we assume to be true in security is not, particularly when we involve users! Our assumptions about policy improving security often turn out to be untrue. We need a system of quantifiable reasoning, that allows us to make informed decisions about the validity of policy choice, to ensure that we do get the right balance of security and convenience. In particular, we have shown that policies based only on one dimensional assumptions are likely to be erroneous. For our simple case, we show it is necessary to take both user and attacker choices into account to get a true understanding of vulnerability. We have proposed a probabilistic framework for reasoning about policy, which allows such discourse. In the future, we propose to extend this work to allow reasoning about more aspects of security policy and the environment, including multiple attacker profiles. We also intend to develop user studies, allowing accurate instantiation of the user based variables, leading to better overall understanding of the validity of security policy change.

The next time your IT department says "we have changed our password policy to increase security", you can legitimately say "prove it".