Systematic Decision Making in Security Management: Modelling Password Usage and Support

Simon Arnell[1], Adam Beautement[2], Philip Inglesant[2], Brian Monahan[1], David Pym[3], and Angela Sasse[2]

[1] Hewlett–Packard simon.arnell@hp.com, brian.monahan@hp.com
[2] UCL a.beautement@cs.ucl.ac.uk, a.sasse@cs.ucl.ac.uk
[3] University of Aberdeen d.j.pym@abdn.ac.uk
[4] University of Edinburgh P.Inglesant@ed.ac.uk

Abstract. We demonstrate the use of a systematic decision-making methodology to support an informed choice of a password policy. Our approach uses an executable system model, grounded in empirical data, to compare, using simulations, two different policy options. The problem is framed economically, with the basis of the comparison being a notion of organizational utility. We quantify utility in this case by considering breaches of system security, users' productivity, and investment in support operations. Using our results, we are able to explore trade-offs between these factors and thus determine the optimal policy configuration given the initial conditions.

1 Introduction

Security managers in organizations have to routinely choose technologies and policies that protect the business-critical infrastructure of their organizations. These choices are constrained by both economic and regulatory circumstances and managers necessarily must make trade-offs between security and such constraints. When faced with analysing these trade-offs security managers currently have very limited tools to aid them in systematic decision making (for example, [7]). Consequently, managers rely on their own experience and intuition when selecting their implementation choices. While we do not suggest that this leads to poor decision making there are inherent drawbacks with this approach:

1. Decisions made in this fashion may be optimal but cannot be shown to be optimal. Their relative effectiveness cannot be established, because the manager has no rigorous means of comparison with other options;
2. Any decisions taken by the manager cannot be shared in a meaningful way with other stakeholders, in particular business and finance managers.

A 2009 article [18] indicated that the rise of cloud computing will make passwords of fewer than 12 characters vulnerable to a brute force attack. We envisage a scenario in which a senior business manager — concerned about the implications of this news — asks the organization's security manager to change the organization's security policy to mandate a minimum password length of 12 characters. The security manager would be wary of the impact such a change may have but without a systematic framework to support his intuition may struggle to communicate this concern to the business manager. Our solution draws upon the language of finance to approach the problem from an economic viewpoint.

Building on our previous work in this area [3, 14, 15] (and indeed on an earlier version of this work released as a technical report in 2011 [2]), we use the economic concept of utility, capturing the managers' objectives and preferences, to motivate the creation of an executable mathematical systems model representing the key processes, locations, and resources of the organization, as well as the environment within which they sit. Critically, we keep the model grounded in real-world empirical data. The combination of these approaches is at the core of our proposed methodology for systematic decision making and forms the key contribution of this paper. Using this model, we can compare the expected utility of the organization under two password policy conditions; the first mandating a minimum of 6 characters and the second a minimum of 12 characters.

In Section 2, we outline the economic concepts behind the development of the utility function. In Section 3, we discuss the design and implementation of the empirical data gathering process. Section 4

gives an explanation of our systems modelling approach. Section 5 provides a conceptual explanation of the executable mathematical model used. In Section 6, we analyze the results and draw the conclusion a 12-character password is desirable, providing that the helpdesk is sufficiently staffed to cope with the increased volume of password resets.

2 Utility (Loss) and Information Security

The fundamental concepts of information security are confidentiality, integrity, and availability. These concepts are *declarative*; that is, they express properties that systems may possess. Alongside these declarative, organizing concepts, sit various *operational* notions — such as access control, authentication, and possession of keys, and so on — that are used in order to establish the declarative properties.

Many authors (for an excellent distillation of the essentials, see [22]) have suggested that confidentiality, integrity, and availability provide an incomplete basis for information security analyses, and have suggested that various concepts be added to them (often confusing the declarative and the operational).

This confusion is problematic in the context of investment decision-making, where the concept of utility — sometimes confused with the declarative/operational above — is critical. Utility theory (see, for example, [17, 25]), particularly as developed in the contexts of macroeconomics and financial economics, provides a highly expressive framework for representing the preferences of the managers of a system.

For example (e.g., [23]), in the macroeconomic management of market economies, central banks play a key rôle. The managers of a central bank may be given, by their national governments, targets for certain key economic indicators, such as unemployment (u_t) and inflation (π_t) at time t (time can be either discrete or continuous here). Their task is then to set a (e.g., monthly) sequence of controls, such as their base (interest) rates (i_t) so that the key indicators are sufficiently close to their targets, $\overline{u_t}$ and $\overline{\pi_t}$, respectively. Typically, using this example, the managers' policy is expressed as a utility function

$$U_t = w_1 f_1(u_t - \overline{u_t}) + w_2 f_2(\pi_t - \overline{\pi_t}) \tag{1}$$

together with system equations, $u_t = s_1(i_t)$ and $\pi_t = s_2(i_t)$, expressing the dependency (among other things) of u and π on interest rates in terms of functions s_1 and s_2 that describe the (macro) dynamics of the economy. Two key components of this set-up are the following:

- The weights w_1 and w_2 (typically, values between 0 and 1) that express the managers' preference between the components of the utility function — that is, which they care about more; and
- The functions f_1 and f_2 that express how utility depends on deviation from target. A simple version of this set-up would take the f_is to be quadratic. Quadratics conveniently express diminishing marginal returns as the indicators approach target, but make utility symmetric around target. More realistically, Linex functions [26, 27, 23], usually expressed in the form $g(z) = (exp(\alpha z) - \alpha z - 1)/\alpha^2$ are used to capture a degree of asymmetry that is parametrized by α.

The managers' task, then, is to set a sequence of interest rates i_t such that the *expected* utility, $E[U_t]$, remains with an acceptable range, as u_t and π_t vary, and trade-off against each other, as the sequence of rates i_t evolves. In general, there can of course be as many components as required in a utility function.

This economic framework can be deployed in the context of information security (see, for example, [3, 15, 16, 5]), where concepts — such as confidentiality, integrity, and availability — that lie within competing declarative categories can be seen to trade-off against one another as the relevant controls — such as system configurations or investments in people, process, and technology system configurations — vary.

In this paper — where are concerned with the use of passwords to access a system and the issues associated with resetting them — the declaractive information security concepts of interest are the following:

- *Breaches*, B, which may be understood as a particular aspect of confidentiality: passwords become known to unauthorized individuals;
- *Productivity*, P, which may be understood as a particular aspect of availability: the user's ability to access the system and perform work-tasks; and
- *Investment*, K, which here is simplified to be the provision of IT Help Desk effort.

We thus end up with a utility (loss) function of the form

$$U(k,l) = w_1 f_1(B - \overline{B}) + w_2 f_2(P - \overline{P}) + w_3 f_3(K - \overline{K}) \quad (2)$$

where the parameters (αs) of the Linex functions f_i and the weights w_i are determined by the managers' preferences, and the control variables: k, the level of investment in help-desk staff; and l, the length of password (here we explore six and twelve characters). These are the parameters explored experimentally using an executable mathematical system model (explained in Section 5). The executable mathematical system model replaces the rôle performed by the system functions, s_1 and s_2, in the discussion above.

We can summarize the approach as follows:

- The organization that deploys information security measures exists in an economic and/or regulatory environment. This environment places constraints upon the systems and security architectures available to the organization's managers;
- The managers formulate a utility function that expresses their policy preferences, which will depend upon the nature of their organization. For example, state intelligence agencies and online retailers will have quite different priorities among confidentiality, integrity, and availability; see, for example, [15];
- In a highly complex situation, such as a security architecture, it is typically not possible to formulate system equations (as functions s_1 and s_2) in the way that is usually possible in, say, macroeconomic modelling. Typically, though, the key control variables, such as system interconnectivity or investment in various aspects (people, process, and technology) of security operations, will be identifiable;
- Instead, however, an executable system model [8], using the key control variables, can be used in order to simulate the dynamics of the utility function.

Finally, note that for the purposes of our results, as presented in Section 6, we work *not* with utility, which one seeks to maximize, but rather with its dual, *loss*, which one seeks to *minimize*.

3 Empirical Data Gathering

3.1 Populating the Model from Empirical Password Studies

To obtain an empirical basis for our model for the case study described in Section 1, we researched the impact of user authentication through passwords on organizational productivity. We chose password authentication because it is the most widely deployed security mechanism in commercial and public sector systems today. Previous research on password practices has established that users prefer simple passwords that are not very secure (e.g., [10]). Most organizations try to counteract this preference through security policies and mechanisms that force users to make passwords more secure — by mandating a minimum length and complexity of passwords, and by requiring regular password resets. However, most users cannot cope with the cognitive demands that result from those policies, leading to an increase in:

1. The number of forgotten passwords [24] and therefore a commensurate increase in the demand for the helpdesk resources required to securely reset them;
2. The number of users who employ 'workarounds', such as writing passwords down [1]. These workarounds often create new vulnerabilities which increase the likelihood of an organizational security breach.

Previous research did not, however, provide a set of data that would allow us to determine the impact of different password policies on organizational productivity in a systematic way. For instance, we were not able to predict by what proportion helpdesk calls would increase if an organization forced users to change from a simple 6 character password to a complex 12-character one. Similarly, determining the risks resulting from workarounds depend very much on the type of users, and the context in which they and the organization operate. The security implications of a password being captured depend on what the user has access to — some users have access to sensitive systems and data, others do not. The risk of a password being captured is very different for users who only access systems from the relatively controlled environment inside an organization's perimeter, compared to mobile users who spend much of their time accessing systems while in public areas, and using 3rd-party wireless access networks.

To create an empirical basis for our model, we conducted a set of studies to obtain data on the impact of different password policies:

1. We estimate the time to generate a password of specific length and complexity, based on observations of a group of participants in a laboratory study;
2. We estimate the time to enter a password of a specific length and complexity, and the number of failed attempts, based on a set performance tests under laboratory conditions;
3. We estimate the frequency of forgetting a password, based on a diary study [14] conducted with employees in two organizations, supplemented by reset statistics from two organizations.

Study 1: Time to GeneratePasswords We measured the time it took the 19 participants (those in cohort G1 in Table 1 below) to generate passwords in response to a specific password policy: the password had to be 7 or 8 characters long, contain 3 out of 4 character types (lower case letter, upper case letters, numbers and symbols), and could not consist of a dictionary words or simple derivatives. Participants took on average 1 min 34 seconds, (standard deviation 86.4) to produce a password that complied with the policy.

Study 2: Time to Type in Passwords We carried out a set of lab-based experiments, in which we measured the time it took participants to enter passwords of varying length and complexity (see Table 1). The passwords used conformed to a range of policies that varied the required minmum length and password composition. In two of the cohorts passwords were chosen by the participants. In the third participants were given a password to use that had been generated by the experimenter.

Table 1. Password Allocations

Cohort	No. of Participants	Policies used by the cohort	Success rate on first attempt
C1	22	Min 6 chars, 3 of 4 character sets	219/220 = 99.55%
		Min 8 chars, at least 1 numeric, 1 capital	215/220 = 97.72%
G1	19	7 letters, numbers, and non-alphanumerics	185/190 = 97.37%
		8 letters and numbers	186/190 = 97.89%
C2	21	Min 12 letters, numbers and non-alphanumerics	178/190 = 93.68%
		Min 10 letters, numbers and non-alphanumerics	182/190 = 95.79%

Participants entered passwords corresponding to two different policies 10 times each. In total, we measured 1200 password entries by 62 participants; 2 of these participants (both in Cohort C2) were unable to complete the experiment because they had completely forgotten the password, and 3 others required password reminders. The mean time overall for a single password attempt was 4.53 seconds. In 1165 of these entries, the password was successful on the first attempt.

Study 3: Frequency of Forgetting Passwords We captured the likelihood of forgetting passwords through diary studies with 32 participants in two organizations — a university and an investment bank — who kept a diary of their password interactions for one week. In total, 144 unique passwords were recorded in the study, used in 982 password entries, of which users reported 5 forgotten passwords, resulting in the need for a new password or helpdesk call. We also obtained statistics for password resets as recorded by helpdesks (the more secure but expensive option) and self-service resets through portals (cheaper, arguably less secure) in these organizations. These are shown in Table 2; in comparison, Florencio and Herley [10] report that the passwords of only 1.5% of Yahoo users have to be reset every month; unlike the organizations in our study, Yahoo does not mandate strong passwords, and most users had weak passwords. The password diary study also revealed that, in addition to the time and effort required to carry out the reset, there are further implications for productivity because of the time it takes for a password reset to propagate through the organization's infrastructure — in one of our organizations, it took on average 2 hours. This means users were 'locked out' from many services for that period of time, and had to re-arrange their primary tasks.

4 System Modelling

Our approach, the mathematical basis of which is presented in [8, 9], is grounded firmly in mathematical logic, computation theory, and probability theory, but employs well-developed, implemented tools. Our approach views a system as having the following key conceptual components:

Table 2. Actual password resets in two organizations

Organization	Type of password	Period of data	Mean password resets per head, month
Investment bank	Single sign-on	January – October 2008	11.4%
University	Single password but not full SSO	December 2009 – January 2010	13.4%

- Environment: All systems exist within an external environment, which is typically treated as a source of events that are incident upon the system rather than being explicitly stated;
- Location: Relevant places connected by (directed) links. Locations may be abstracted and refined provided the connectivity of the links and the placement of resources is respected. Mathematically, various graphical structures capture this notion [8, 9];
- Resource: The notion of resource captures the components of the system that are manipulated by its processes (see below). Resources include things like the components used by a production line, the tools on a production line, computer memory, system operating staff, or system users, as well as money. Mathematically, we capture this notion using certain algebraic structures [21, 8, 9];
- Process: The notion of process captures the (operational) dynamics of the system. Processes manipulate resources in order to deliver the system's intended services. Mathematically, we use algebraic representation of processes based on the ideas in [20], integrated with the notions of resource and location [21, 8, 9].

This framework is explored in detail in [8, 9, 12], with related work in, for example, [11, 13]. The Gnosis modelling tool, which we use to implement this framework, is described in [8, 12]. The basic idea is to implement the process component as (concurrent combinations of) sequences of actions, together with a range of control constructs. Resources are declared at locations, and may be shared by processes, which may manipulate the quantity and location of a resource. Events incident upon the system from the environment are represented by the results of the sampling of declared probability distributions by processes. Here the environment is intended to include not only things outside the system of interest, but also any internal parts of the system not modelled in detail.

The experimental methodology deployed in a study of the kind described in this paper must be tailored to the constraints of the experimental environment. The key point is the classic mathematical modelling cycle — of observation, model construction/refinement, model analysis, real-world interpretation of analysis, and back to observation — iterated until the modeller is satisfied with the viability of the model [9].

In the context of the study of password usage, as described in this paper, some methodological details are, owing to the relative sparsity of the data, important. In the presence of sufficiently large data sets, classical statistical clustering techniques could be applied. Our approach is to examine the individual executions of the system model and compute the utility function of the resulting clusters of points. Thus we are sampling the utility function rather than analyzing it statistically, witnessing the formation of clusters, and observing local utility maxima. A thorough statistical analysis is further work.

4.1 From Empirical Data to the Model

We chose to model two different password lengths, utilising three different characters types. The model assumes probability distributions for mis-typing – related to the length, variety of character types, and newness of the password – and for time to enter a password, from a normal distribution with fixed mean and standard deviation. We include a representation of the threat environment through the risks of a partial or complete password capture. The model assumes that these risks parameters are related directly to the location of the user, and for this purpose we modelled users as three categories of user in four locations, making the assumption that the proportion of time in each type of location is related to the user category. We have validated our assumptions about the three types of users in four locations by conducting in-depth interviews with three senior information security managers.The password data we gathered is used to provide support underlying our choice of the *transfer function* we use in the model. This function essentially defines a way of mediating from various characteristics (e.g., password length, number of retries) into a

probability of successful authentication — see the initial comments to Section 5 (and also 5.4) explaining the rôle of the transfer function in our model.

5 The Conceptual Model

In this section, we explain our executable mathematical model, constructed according to the general methodology outlined in Section 4 using the Gnosis modelling language [8, 12] and structure explained in Section 4. Rather than try to explain the Gnosis code in detail, we instead explain conceptually its various components and their interconnection.

We begin by imagining our scenario consists of a corporate work situation with a number of types of corporate user: ROAD_WARRIOR, OFFICE_WORKER, and EXECUTIVE, in a variety of different types of location: HOME, WORK, PUBLIC, and IN_TRANSIT.

All of our users need to log-in daily and exchange information with the corporate intranet via their desktop/laptop systems. In making these accesses, they need to 'prove' and test their credentials by typing in their password, typically doing this authentication several times per day. We only consider those accesses where an authentication challenge is required, ignoring tasks where no challenge is necessary.

From the systems modelling perspective, we consider the life-cycle of a user (of a given type) for whom authentication challenges are events that randomly occur from the environment. The key conceptual components of the model are then *authentication*, *password resets*, and the *threat environment*. Note that, critically, all of these components exist at the operational level, and that their dynamics influence the utility (loss) calculation of the levels of the associated declarative components (here breaches, productivity and investment). Investment is represented here by the number of *help desk staff* employed in the model. The utility (loss) function (of the remaining two components) is evaluated post-facto from the outcomes produced experimentally by running our model.

Finally, a critical conceptual component of the model is the *transfer function*, (explained in detail in Section 5.4). The key idea is to represent an interaction between the users as represented in the model and the environment. More specifically, the transfer function characterizes the relationship between the number of attempts to enter correctly a password and the probability of successfully entering a password. The use of a transfer function — as initiated in a related previous paper on USB memory stick security [3] — is a convenient way to capture a very complex situation, the exploration of which is beyond the scope of this paper. The form of the function taken should remain consistent with the empirical data obtained. It is not, however, determined by it. Such a determination would require extensive and delicate experimental work, beyond our present scope.

Sections 5.1 and 5.2 describe the functioning of the model with respect to those concepts.

5.1 Authentications

The authentication step requires entering a password of some length. We assume usernames are perfectly recalled hence they are not represented in the model. Note that the probability of failing to correctly type the password increases with password length and complexity, as determined by the empirical data presented in Section 3.1 and expressed via the transfer function. The following list represents the relevant states of the user as seen by the authentication system (there are some model-specific states that are necessary for model execution but have no wider relevance):

UnAuthenticated The user is not currently authenticated: productivity is zero.

Authenticated The user has currently authenticated and productivity is allowed.

Failed This temporary state occurs after an entry failure. Retries may be attempted after a short delay. If the user does not retry after a longer delay, their state returns to **UnAuthenticated** (i.e., this means that the failure state is forgetfully forgiving);

Expired The user is explicitly prohibited from accessing the system. This arises because users failed to comply with the 90 day password renewal policy. Once a password expires, the password is automatically reset – in other words, the user does not have to *initiate* the reset process (in effect, it just happens). At this point the user incurs a short delay in acquiring a new password. We treat the entire elapsed time is an accountable period of productivity loss;

Locked-out The user is locked out, as result of multiple entry failures, but a password reset is pending: this is an accountable period of productivity loss.

Our model has a simulated duration of a single year, with time unit = 1 day. For each user type, we generate authentication attempts randomly at an average rate per day. These then proceed according to the flow described above.

Productive work is assumed to be any task that requires a successful authentication to be made — we ignore all those tasks that don't need authentication in the first place. Productivity losses arise when a user wishes to perform work but is in an authentication state that does not allow it. Our proxy statistic for a user's productivity loss is taken to be *the total elapsed time whilst inhibited from working because of password resets*. This statistic neatly accounts for both the number of resets and usefully encodes how much delay there is in performing a password reset. This is analogous to the concept of computer downtime arising in classic performance analysis.

We could naturally encode various password policies imposed by the organization such as: passwords are required to have at least n characters (for some suitable n); after 5 failed attempts, a password reset is always required; every user must have changed their password some time in the last 90 days; equivalently, passwords older than 90 days will expire. All of these policies are represented in the current model.

Because of users′ respective rôles and their locations, they are all subject to different levels of password capture risk. These are represented in the model by a location-indexed point distributions; see Table 3. For instance, we simply note that ROAD_WARRIOR users are more likely to be working in risky places (i.e., IN_TRANSIT and PUBLIC) and are thus more exposed to potential breaches. We assume the standard fiction that all breaches are perfectly and accurately observable only after the fact and never before. This says that breach detection is essentially perfect, while acknowledging that any breaches that do happen to occur do so only because they are executed extremely well and therefore could not have been anticipated. We do not model the details of the password administration process or its infrastructure as such. All we capture in the model are the *service characteristics*, such as the times taken from initial request to resolution. The number of staff present at the helpdesk is set by the level of investment. For the purpose of our simulation we use three fixed levels of investment, representing 1, 2 and 3 staff members.

5.2 Resets

For verisimilitude and general credibility, we split the password reset/renewal service into two levels of service, reflecting a common practice in service industries:

Light reset Essentially a mitigating quick-response, automated, password renewal system, involving zero administration overhead. Although typically a cheaper and easy alternative, it may have some security issues (such as potential for malicious resetting, replays or interception) and hence some unwelcome security drawbacks — that is, it is imperfect. It still takes some time to perform (yielding a productivity loss, albeit a lower one than the heavy reset) — but doesn't require expensive resources such as an administrator's attention;

Heavy reset This is a more involved process and requires a password administrator to check credentials and appropriately reset the password. Password requests go into a queue and could potentially take a long time (i.e., more than a day) to reach resolution.

Note that this allows us to neatly capture the issue of escalation: some password resets require escalated responses (heavy reset) while others can be safely resolved automatically via light reset. As we offer the decision-maker control over the proportion of light and heavy resets, we can study the friction points that exist in the trade-offs between productivity loss, breaches and investment required in the help desk to fulfil resets. Whichever reset method occurs, a password reset (for a discussion of costs, see [19]) should be considered to include the non-trivial time and effort of generating a new password and, possibly, the time for the password to propagate around multiple servers. For the purposes of our model, however, these are complexities to be added as later refinements.

5.3 Threat Environment

An organization faces numerous threats, some of which are due to the capture of a user's credentials. Such an adversary may casually shoulder-surf a user entering their password or may gain access to the full password through some other means, perhaps placing a user under duress.

A number of abuse cases exist for the leakage of a user's password, from corporate espionage to a jealous partner seeking revenge. A system model however does not need to capture such details of a user's life, we instead abstract away from this to an understanding that, when passwords happen to be captured, they are either fully or partially captured.

We parametrize the global threat environment by allowing the customer to choose a threat level, allowing them to model hypothetical scenarios for changes in their threat environment. The threat level defines how many attacks hit the system. We extend this idea of a threat environment and consider it not just at a global level but allow the customer to define threat levels at a local level. As the model is located we allow each of the locations a user may authenticate to have probabilities assigned to the type of threats that may exist. Table 3 gives an example definition of a threat environment.

Table 3. Example probabilities for types of password captures in locations within the model

Location	Home	Work	In transit	Public
Full	$\frac{1}{3}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{2}$
Partial	$\frac{2}{3}$	$\frac{3}{5}$	$\frac{3}{5}$	$\frac{1}{2}$

While we have attempted to provide empirical backing for as much of the model as possible, given the scope of our project, the data contained in Table 3 are examplar values. A detailed risk analysis of the locations in question would be required in order to provide accurate, real-world values.

We model a partial capture as observation through shoulder-surfing, for example and, for simplicity, we assume that three characters are captured; the number of partial captures required to fully capture the password therefore depends on the password length.

5.4 Transfer Function

We abstract a user's authentication process as a transfer function that takes the following parameters: password age, policy-defined required length, policy-defined required complexity, policy-defined maximum number of attempts, number of attempts, policy-defined maximum time allowed to authenticate, and time to enter. The function provides us with a probability that a user was successfully logged into the particular system, and is taken to be the product of a number of dependencies, with parameters defined below:

$$P(X) = \prod_{i=1}^{5} x_i \tag{3}$$

These dependencies are based on modelling intuition and, critically for the validity of the model, the empirical data we collected in support of the model. The empirical data collected allows modellers to attune their intuitions on the morphology of the transfer function. If a purely data-driven transfer function were desired then one must collect a much larger data-set to improve confidence levels, thus our current dataset provides only an indication of the possible shape of the transfer function. However, grounding the model behavour in empirical results allows the output of the model to be used in support of decision making in comparable real-world situations. This is a necessary step in moving toward a more systematic science of security management.

We now discuss the assumptions that have been made to model each of the dependencies.

Password length Based on the empirical data results we consider that passwords under 6 characters are very easy to recall, therefore passwords of this length do not reduce the users chances to successfully authenticate. For longer passwords the probability of successfully authenticating is inversely proportional to the length of a password; that is, as $length \to \infty$, $P(success) \to 0$:

$$x_1 = \begin{cases} 1 & \text{if } length \leq 5 \\ 1 - \frac{length-6}{5000} & \text{if } length > 5 \end{cases} \tag{4}$$

Number of tries The number of tries affects the probability only when the maximum number of tries permitted by policy is met, at which point the user cannot authenticate.

$$x_2 = \begin{cases} 1 \text{ if } tries \leq maxTries \\ 0 \text{ if } tries > maxTries \end{cases} \tag{5}$$

Password complexity All password complexities have a $P(success) < 1$, simple, moderate and complex passwords are increasingly harder to recall and/or correctly type. We characterize simple passwords as being all lower case characters [a-z], moderate as comprising lower case, upper case, and numbers [a-z, A-Z, 0-9], and complex as being formed from any available characters.

$$x_3 = \begin{cases} 996/1000 \text{ if } complexity = complex \\ 998/1000 \text{ if } complexity = moderate \\ 999/1000 \text{ if } complexity = simple \end{cases} \tag{6}$$

Time taken to authenticate Using the empirical data we set a maximum time window for password entry. Entry attempts that exceed this policy-defined window are timed out.

$$x_4 = \begin{cases} 1 \text{ if } timing \leq maxTime \\ 0 \text{ if } timing > maxTime \end{cases} \tag{7}$$

Password age We assume a password requires a training period before the probability of a successful recall reaches the values used in (6) (the point at which a user has learnt their password), at which point point as $age \rightarrow \infty$, $P(success) \rightarrow 1$. Thus, the probability of a successful authentication is directly proportional to the password's age:

$$x_5 = \begin{cases} 0.999 & \text{if } age \leq 2 \\ 1 - \frac{1}{1000(age-1)} & \text{if } age > 2 \end{cases} \tag{8}$$

By fixing a number of parameters an instance of the transfer function that can be generated that allows us to relate the probability of a successful authentication against password length and age. It is worth noting that for a model that allows a maximum of 5 tries and 25 time steps to enter a password of moderate complexity there is little difference in this probability between 0 days (≈ 0.995) and 90 days (≈ 0.998). However in a model with 100 users who authenticate with a mean of every 20 minutes, this equates to an order of magnitude difference in the number of daily password failures.

6 Analysis and Conclusions

In Section 2, we considered the example of a security manager needing to determine the impact on utility (loss) — as represented by the cost of breaches, productivity loss, and investment in support services — of switching to a password policy mandating a minimum of 12 characters from one requiring only 6. To investigate this decision, we executed the model described in Table 4 (6 combinations in total). These combinations represent three levels of investment in infrastructure (expressed as the number of helpdesk staff) and the two policies under consideration.

Table 4. Experimental parameter combinations

Parameters	Values
Password Length	6; 12
Number of Help Desk Staff	1; 2; 3

We ran each of these experimental combinations 48 times, for a simulated duration of 1 year and with an independently randomized random number generator seed to enable Monte Carlo-style simulation. The results obtained from these runs yield the following data: number of breaches seen; productivity loss because of password resets for OFFICE_WORKER, ROAD_WARRIOR and EXECUTIVE.

With the results generated we can make use of the utility (loss) function as described in Equation 2. As the utility (loss) function is not required in the simulation runs itself, the manager can investigate the effects

of different trade-offs and target preferences using the same underlying data. For the purposes of this paper we have taken the following targets and weightings in Table 5, where the *perfomance loss* is measured in terms of the total salary paid out while staff are unable to work effectively due to password reset.

Table 5. Preferences for Targets and Loss Function Weightings

	Targets	Loss Function Weighting	Linex (α)
Breaches	1.25	3	0.4
Performance Loss	0.65	6	3

Both values are scaled so that a value of 1 means the same as 1 basic yearly salary. The above targets mean that annual costs due to performance loss should be at most 65% of a basic annual salary and, likewise, annual costs due to breaches should be at most 125% of a basic annual salary.

The loss function weightings we took are a value of 1 to account for help desk costs (e.g., salary), 3 for breaches and 6 for performance loss. This choice makes performance loss twice as important as breaches, expressing a preference for minimizing performance loss over breaches. For example, such a preference might represent a corporate bias towards penalizing anything that immediately causes a decrease in staff productivity. This preference can be adjusted by changing the loss function weighting and thus it forms one of the parameters through which a manager can adjust the model to represent better his organization.

The Linex values are both positive (to penalize any overshooting of loss targets) and their different relative strengths further emphasises the security manager's preference to penalise performance loss more heavily than breaches.

6.1 Results and Discussion

Our organizational utility equation was formed of three factors; breaches, productivity loss and investment in support services. Our simulation runs took investment in support services, in the form of the number of helpdesk staff, as a parameter. Thus our results, with respect to organizational utility, are formed of the following:

- Cost of Breaches (CB): This value represents the total annual breach costs to the organization;
- Cost of Performance Loss (CP): This value represents the total annual costs of performance loss (due to password resets) for the organization.

The total performance loss was computed as a weighted sum of the performance losses as seen by each type of user: OFFICE STAFF, ROAD WARRIOR and EXECUTIVE, weighted to their respective salaries.Table 6 shows the minimum and maximum costs associated with each policy across all the simulations:

Table 6. Breach and productivity costs

Utility Loss	6-Character Policy		12-Character Policy	
Range	Min	Max	Min	Max
CB	0.8	3.9	0.2	2.1
CP	0.4	1.6	0.45	1.85

Here the trade-off between productivity and security can clearly be seen. Shifting to a 12-character policy substantially reduces CB, while both the maximum and minimum CP has risen. The 12-character policy incurs higher productivity losses as more resets are required, which can be mitigated by investing in more helpdesk staff. Recall that we ran simulations with 1, 2 and 3 helpdesk staff members. Across both policies the maximum productivity loss occurred with 1 helpdesk employee. Adding a second helpdesk staff member dramatically reduced the productivity losses incurred indicating that a single helpdesk staff was insufficient to cope with the demand for resets. However, moving from 2 to 3 helpdesk staff did not generate gains on the same scale. In fact the minimum CP was relatively similar for 2 and 3 helpdesk staff (0.91 and 0.88 respectively, compared to 1.85 for 1 helpdesk staff). This implies there is little advantage

in employing more than 2 helpdesk staff. That being said, the loss function for the 6-character policy consistently showed much higher loss values (i.e., max. values of 12.5 and 15.5, for 2 and 3 helpdesk staff) than the 12-character policy (i.e., max values of 3.5 and 4.2, for 12 characters and 2 and 3 helpdesk staff). This indicates that the trade-off between security and usability is in security′s favour as breach costs are reduced at a faster rate than productivity costs are incurred.

From the above, we can conclude that:

1. Using 1 help desk staff leads to greater CP in general, whereas either 2 or 3 help desk staff substantially reduces CP losses. This indicates using either 2 or 3 help desk staff - using 1 help desk staff is inadequate;
2. 6-character passwords generally have greater CB than 12-character passwords. While 12-character passwords increase CP this value is lower than the savings through reduced breaches. Thus the overall utility score indicates that 12-character passwords should be used;
3. Comparing the outcomes using 12-character passwords and 2 or 3 help staff, it is clear from the loss function that using 2 help desk staff will incur the least losses overall, both in terms of minimum (2.0) and maximum (3.5) loss.

In the context of our original scenario, the security manager should therefore report back to the business manager that adopting a 12-character password is a feasible proposition. However, this will generate additional load on the help desk. A critical factor in maintaining productivity is adequate staffing in this area, as the study shows, via the large change from one to two help desk staff. In our study, the number of support calls generated was too low to add granularity to the staffing of the helpdesk. Under both policies a single staff member was too low but two helpdesk staff represented a sufficiency. In a larger organization (or under a different reset policy) the optimal zone of staffing will likely differ between the policies. The 12-character policy did generate a higher number of reset calls (and thus productivity loss) but in this case the increase was not sufficient to require three staff members. Additionally, the model does not fully account for the impact on user effort of a 12-character password; this is discussed in more detail in the following section. Thus while we have made a good start in analyzing the impact of this decision there are several further refinements that could alter our current conclusion.

6.2 Conclusions and Future work

We have outlined a methodology that allows us to systematically explore security decision making through the use of a system model supported by empirical data. The conclusions we have drawn are based on the parametrization of the model and thus represent an outcome specifically related to the fictional organization we created for the purpose. The key point to understand here is not that 12-character passwords are better than 6-character, but that it is possible through a rigourous and repeatable process to investigate the impact of security decisions in a utility-driven context. As the application of the utility function occurs after the dataset is generated it is possible for managers to dynamically explore different policies and preferences before making a decision and without repeatedly running the model. This allows security managers to efficiently make systematic decisions and, arguably more importantly, communicate them to other stakeholders in a familiar manner supported by data rather than intuition. The use of empirical data in this methodology greatly improves the strength of the results generated and adds confidence to the conclusions drawn.

There are several key areas of our approach that we have targeted for revision and expansion in follow up work. User effort is currently under represented in the model and has the potential to significantly alter the conclusions drawn. Considering the notion of the Compliance Budget [4] we can see that the impact on the user of being forced to use 12-character passwords could push them to adopting insecure behaviours such as writing passwords down or sharing them with colleagues. The increased risk of breaches resulting from such behaviours has not been factored in and thus should be included in any future work.

Additionally, there are two features that subsequent versions of the model will incorporate. These are multiple passwords per user and passwords with varying usage frequency. We can envisage that users may wish to access a range of services, some more frequently than others, each requiring a different password. Less frequently used passwords are more likely to be a cause of failed logins. Incorporating these changes will add significant richness to the model

Further changes may be included to allow a greater policy influence over the model behaviour. For example rather than having reset type controlled by a distribution we could allow this choice to be set by policy such that remote resets (at the PUBLIC or IN_TRANSIT locations) must always use the heavy reset. Again, this adds flexibility and richness to the model.

References

1. A. Adams and M. A. Sasse. Users are not the enemy: Why users compromise security mechanisms and how to take remedial measures. *Communications of the ACM*, 42(12):40–46, December 1999.
2. S. Arnell, A. Beautment, P. Inglesant, B. Monahan, D. Pym, M. A. Sasse. Systematic Decision Making in Security Management Modelling Password Usage and Support. *HP Laboratories Technical Report* HPL-2011-36
3. A. Beautement, R. Coles, J. Griffin, C. Ioannidis, B. Monahan, D. Pym, A. Sasse, and M. Wonham. Modelling the Human and Technological Costs and Benefits of USB Memory Stick Security. In M. Eric Johnson, editor, *Managing Information Risk and the Economics of Security*, pages 141–163. Springer, 2008.
4. A. Beautement, M.A. Sasse, and M. Wonham. The compliance budget: managing security behaviour in organisations. In *Proc. of the 2008 Workshop on New Security Paradigms*, pages 47–58. ACM Digital Library, 2009.
5. Y. Beres, D. Pym, and S. Shiu. Decision support for systems security investment. *Proc. BDIM 2010*, IEEE, 2010.
6. G.M. Birtwistle. *Discrete Event Modelling on SIMULA*. Springer-Verlag, 1987.
7. Robert Coles. Keynote Address, Eighth Workshop on the Economics of Information Security (WEIS 2009), University College London, England. 24–25, June 2009.
8. M. Collinson, B. Monahan, and D. Pym. Semantics for structured systems modelling and simulation. In *Proc. Simutools 2010*. ICST: ACM Digital Library and EU Digital Library, 2010. ISBN: 78-963-9799-87-5.
9. M. Collinson, B. Monahan, and D. Pym. *A Discipline of Mathematical Systems Modelling*. College Publications, 2012.
10. D. Florencio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on the World Wide Web*. Association for Computing Machinery, May 2007.
11. S. Gilmore and J. Hillston. The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. LNCS 794:352–368, 1994.
12. Gnosis. `http://www.hpl.hp.com/research/systems_security/gnosis.html`.
13. J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
14. P. Inglesant and M. A. Sasse. The True Cost of Unusable Password Policies: Password Use in the Wild. *Proc. CHI 2010: 28th ACM Conference on Human Factors in Computing Systems*, April 10–15, Atlanta, GA, 2010.
15. C. Ioannidis, D. Pym, and J. Williams. Investments and trade-offs in the economics of information security. *Proc. FC & DS '09*, R. Dingledine and P. Golle (eds.), LNCS 5628:148–166, 2009.
16. C. Ioannidis, D. Pym, and J. Williams. Information security trade-offs and optimal patching policies. *European Journal of Operational Research*, 216(2):434–444, 2012.
17. R.L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Wiley, 1976.
18. Robert Lemos. Harnessing the cloud for hacking. *Technology Review*, December 2009.
19. Ellen Messmer. Data Breaches Get Costlier. *PC World, January 25, 2010*.
20. R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25(3):267–310, 1983.
21. P.W. O'Hearn and D.J. Pym. The logic of bunched implications. *Bull. of Symb. Logic*, 5(2):215–244, 1999.
22. Donn Parker. *Fighting Computer Crime: A New Framework for Protecting Information*. Wiley, 1992.
23. Francisco J. Ruge-Murcia. Inflation targeting under asymmetric preferences. *Journal of Money, Credit, and Banking*, 35(5), 2003.
24. M.A. Sasse, S.Brostoff, and D. Weirich. Transforming the "weakest link": a human-computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3):122–131, July 2001.
25. Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-theortic, Logical Foundations*. Cambridge University Press, 2009.
26. H. Varian. A bayesian approach to real estate management. In S.E. Feinberg and A. Zellner, editors, *Studies in Bayesian Economics in Honour of L.J. Savage*, pages 195–208. North Holland, 1974.
27. A. Zellner. Bayesian prediction and estimation using asymmetric loss functions. *Journal of the American Statistical Association*, 81:446–451, 1986.