Validating an Agent-Based Model of Human Password Behavior

Bruno Korbar
Department of Computer Science
Dartmouth College
Bruno.Korbar.18@dartmouth.edu

Jim Blythe
Information Sciences Institute
University of Southern California
blythe@isi.edu

Ross Koppel
Department of Sociology
University of Pennsylvania
rkoppel@sas.penn.edu

Vijay Kothari
Department of Computer Science
Dartmouth College
Vijay.Kothari@dartmouth.edu

Sean Smith
Department of Computer Science
Dartmouth College
Sean.Smith@dartmouth.edu

Abstract

Effective reasoning about the impact of security policy decisions requires understanding how human users actually behave, rather than assuming desirable but incorrect behavior. Simulation could help with this reasoning, but it requires building computational models of the relevant human behavior and validating that these models match what humans actually do. In this paper we describe our progress on building agent-based models of human behavior with passwords, and we demonstrate how these models reproduce phenomena shown in the empirical literature.

1 Introduction

The valuation of a given security policy is often predicated upon assumptions that fail in practice (e.g, (Blythe, Koppel, and Smith 2013)). For example, a plethora of password discussions begin with the password paradox: users must pick strong passwords–so strong that the average user cannot remember them–yet they must never be written down. We *know* this doesn't work. We *know* that humans have inherent limitations that invalidate the capabilities we blindly attribute to them. Yet we continually design ineffective policies based on those very assumptions that we *know* to be wrong. And, in doing so, we induce unexpected workarounds that invalidate our valuations. The natural questions are then: can we do better? More importantly, how do we better? That is, how do we rectify our flawed assumptions and evaluate security policies not based on assumptions that we'd like to hold, but assumptions that do hold? And how do we harness the corpus of existing security knowledge to design better security policies?

One approach is agent-based simulation that faithfully models human behavior, including circumvention of security rules. As we discuss in Section 3 below, such a simulation would let us measure the aggregate security afforded by a set of password policies (e.g., the average strength of user-created passwords, the number of passwords the user has written down, the rate of password reuse across services). We could then evaluate the effect of various policy choices before deploying them and avoid the problem of "dialing up" security rules only to make things worse. Even learning the relative shape of the utility vs. parameter curve would be helpful.

One further motivation for using an agent-based approach to investigate password security is the sheer complexity of the environment. Many users must manage a sizable portfolio of passwords; to do so, they employ coping techniques (e.g., (Florêncio, Herley, and Van Oorschot 2014b). Some users reuse a single password across different services. Others use a family of related passwords across services. The varying extent to which a compromised account at one service can escalate to compromise accounts on other services further complicates matters. And we're just scratching the surface. In such complex environments, a mathematical analysis of security can quickly become unwieldy, while a simulation-based approach remains viable.

Kothari et al. (2015) reported on previous work on developing an agent-based password simulation to evaluate the security afforded by a password composition policy, given the nuances of human cognition and behavior as it pertains to the target organization and other organizations. But why should we believe it? In this paper, we try to answer this question by corroborating our model with empirical data from the password security literature. That is, we parameterize our model so that our inputs agree with empirical data. Then we show that under these real-world settings our simulations produce results consistent with other empirical data.

The rest of this paper is structured as follows. In Section 2 we discuss related work. In Section 3 we provide an overview of our simulation goals. In Section 4 we detail the current state of the password simulation. In Section 5 and Section 6 we explain our validation methodology and results. In Section 7 we discuss the results of performing a one-factor-at-a-time sensitivity analysis. In Section 8 we discuss future work and conclude.

2 Related Work

In our earlier work, we argued that agent-based modeling can be useful for assessing the aggregate security of an organization and for accounting for user circumvention of security (Kothari et al. 2014). More recently, we developed a password simulation for measuring the security afforded by a password composition policy (Kothari et al. 2015), as well as catalogued and analyzed the structure of user circumvention scenarios (Smith et al. 2015). We build upon this work by documenting our steps toward validating the model, and discussing our plans to explore more circumvention issues.

Other password simulations exist (e.g., (Shay, Bhargav-Spantzel, and Bertino 2007), (Renaud and Mackenzie 2013), and (Choong 2014)). While there is certainly some overlap between these simulations and ours, our objectives are to create an agent-based simulation that can be used to assess the efficacy of a password policy in a global context in which users have accounts on multiple services, and to model the underlying human processes based on minimal assumptions. We believe this approach widens the scope of what we can model, but it also requires significant validation, which is one focus of this paper.

The cognitive architecture ACT-R provides a memory module that has been widely employed in the literature (e.g., (Pavlik and Anderson 2005)). We adopted this memory module to model the process by which users forget passwords.

To revise and parametrize inputs to our model, we incorporated models and data from the following password security papers. Florêncio and Herley (2007) conducted a large-scale study of user-password behavior, discovering a number of interesting statistics. Florêncio, Herley, and Van Oorschot (2014a) provided a survey of recent literature on password security and also provided sound guidance. Florêncio, Herley, and Van Oorschot (2014b) constructed mathematical models for how users manage password portfolios and, using said models, justified circumvention practices as necessary coping mechanisms given the finite effort users are willing to expend on password management. NIST provides an entropy-based password strength measure (Burr and others 2013).

To validate our model we compared outputs from simulation runs with data from Florêncio and Herley (2007) and Bonneau and Schechter (2014). Florêncio and Herley (2007) was mentioned earlier; Bonneau and Schechter (2014) discussed an incremental approach toward having users memorize a 56-bit code and provided baseline values pertaining to how long it takes a user to memorize security codes.

There also exists important password security research that is related to our work in this paper, albeit not directly used to parametrize our model or to validate our results. Here, we provide a small sampling of such research. Riley (2006) conducted a survey and found that users generally employ password management practices that are weaker than those they believe they should employ. Gaw and Felten (2006) conducted a survey to learn about user password management strategies. Das et al. (2014) conducted a survey and analyzed password breaches for different services to better understand password reuse.

3 Simulation Goals

When choosing policies for passwords and other user-facing interfaces, security administrators (or the authors of the guidelines they follow) intend to optimize their aggregate site security given various cost and usability constraints. However, this approach implicitly requires an effective understanding of how aggregate security follows from policy decisions.

In our extensive fieldwork (partially documented in Smith et al. (2015) and Smith and Koppel (2014)), we have catalogued many ways in which human behavior undermines the "textbook" understanding of this mapping.

For one example, one might imagine that aggregate security is a monotonically increasing function of tunable parameters: "dialing up" security obviously makes things better. Unfortunately, we found many scenarios where this fails to happen. Two such patterns:

- Dialing up security can make things worse. For example, a hospital used short timeouts and proximity detectors to log out abandoned sessions, but irritated users ended up placing Styrofoam cups over the detectors, causing them to have permanent false positives. In hindsight, a 20-minute timeout would have yielded better overall security than a 5-minute one.
- Dialing down security can make things better. For example, an infosec officer forced executives to use the same password for both their work and their benefits accounts. Consequently, executives stopped sharing passwords with their assistants.

We also find situations where the implicit assumption that a site's effective security follows only from that site's policies (and a site's policies affect only that site's security) fail to hold. Two examples:

- Numerous enterprises report that business sites were set up with SSL/TLS servers using self-signed certificates. This practice effectively teaches users to ignore browser warnings for these sites, and all other sites as well. Thus, an employer's bad decision puts her employees' bank accounts at risk.
- The sheer prevalence of password reuse creates even more ways for risk to flow. If a user reuses a password across services, then a password breach at one service can increase risk of account compromise on others.

Thus, understanding what happens to aggregate security at a site—let alone over a set of sites—requires taking into account when and how human users will start to break the rules. This requires an effective model of human behavior as well as a way to look at how collections of humans across collections of systems interact. Once we have a validated model, we will be able to examine these questions more deeply. For example:

- What *does* the graph of a site's aggregate security versus password length, or password change frequency, look like?

- What does it look like if we take into account that some users will start to reuse passwords at other sites, so making site S_A's rules stronger means more user accounts at S_A will be susceptible to vulnerabilities at site S_B?

Understanding these curves may help the field achieve better and more usable security.

4 The Model

As noted earlier, this work builds on our earlier password simulation, discussed in Kothari et al. (2015). Here, we explore the latest iteration of the model, only going into deeper discussions for essential details and updates.

4.1 DASH: The Underlying Software Platform

Our password simulation is based on DASH, a software platform that supports cognitive agents with modules for reactive planning, mental models and dual-process reasoning (Blythe 2012). The reactive planning module, building on Morley and Myers (2004), is used to provide the agent with goal-directed behavior that is responsive to changes in the world. A goal is a predicate, describing an objective or a desired state of the world, that is matched to methods of achievement from the agent's library. These methods disaggregate the goal into simpler goals that are recursively matched to other methods, or to actions that the agent can take. After an agent selects and performs an action in service of a goal, she makes observations about changes in her environment and continues working on her goals. In this way, she monitors progress toward goals, choosing her working goal and methods to achieve it as warranted by her observations.

We use the reactive planning module to create an agent who works with passwords in service of her higher-level goals (e.g., logging in to an account). This approach is based on observations about human security behavior: users focus on performing everyday tasks, resulting in behaviors that differ from those observed when subjects attend to security as a primary goal (Jakobsson et al. 2008). Our password simulation therefore involves a user agent who tries to achieve goals that require creating, using, and managing several accounts for several services. This provides a context for modeling cognitive load and temporal repetition that affect password generation and recall.

Actions taken by agents are processed by a module called the *world hub*, which also maintains world state. When the world hub receives an action from an agent, it processes the action by generating a result for the action, updating world state, and relaying the action's result to the agent. In general, a DASH agent may interact directly with its software environment, bypassing the world hub, or many DASH agents can connect to the same world hub, which can then model their interaction through shared resources and world state. In this paper we focus on a single agent who interacts with the world hub.

Essential agent behaviors, interactions, and processes can be viewed as a repeating cycle as seen in Figure 1. First, the agent chooses a goal that provides maximum utility and an action in service of that goal. Once this action is chosen, the agent performs the action and sends it to the world hub. Next, the world hub processes the action, updates the world state accordingly, and determines the result of performing the action. This result is then sent to the agent. Last, the agent processes the result and and updates her beliefs, completing one cycle.

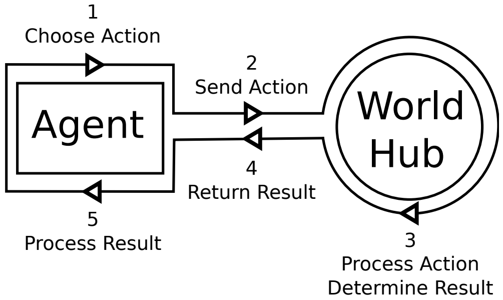

Figure 1: One agent cycle.

The agents and the world hub are coded in Prolog. Agents communicate with each other and with the world hub by passing messages to Java code which handles the communication. The linking between the Prolog and Java code is achieved via JPL [1].

4.2 Our Password Simulation

We have extended DASH by constructing agents who behave as humans using passwords.

The agent simulates a user who interacts with services, primarily by creating accounts, logging in to accounts, logging out of accounts, and resetting account passwords. Each of these tasks is a subgoal that satisfies a repeatable top-level goal. Consider an agent choosing a subgoal during the beginning of a cycle. If, during the previous cycle, the agent was working toward a subgoal that was not achieved, then she will continue to work toward that subgoal. Else, if her latest action achieved a subgoal, she will choose a service uniformly at random from all services and choose an appropriate subgoal as guided by viable interactions with that service (e.g., if an account has not been made for that service, she will create one). She will then recursively disaggregate subgoals, ultimately arriving at an action she can perform, and the cycle will continue as discussed earlier.

Let's consider some agent interactions. Imagine agent Alice chooses to create an account with an email provider. When prompted to enter a password, she enters and submits a password that she believes is memorable and secure. The email provider then determines whether her password meets its criteria and relays to her the result of her password creation attempt. If the creation attempt succeeds, then, at some later point in time, she will attempt to log in to that account. Suppose that when she tries to log in, she believes

[1] JPL: A bidirectional Prolog/Java interface. `http://www.swi-prolog.org/packages/jpl/`

her password is P, but in reality it is something else. She will attempt to log in with P and the service will tell her that P is the wrong password. In turn, her confidence that P is the correct password will reduce. Suppose she tries another password P' and it is indeed the right password. The service will relay to her that it is the right password and her confidence in P' will increase. Successful log in attempts with the right password also reduces the rate at which she forgets her password.

Below, we examine aspects of the model in greater detail. We begin by exploring modules that are critical to faithfully modeling a human. Next, we explore the primary agent subgoals of creating accounts, logging in to accounts, resetting passwords, and logging out of accounts. We then discuss password circumventions and important functionality of the world hub. Last, we explain the stopping condition.

Password Beliefs The agent maintains a set of password beliefs associated with each of her accounts. This set comprises passwords and associated password strengths, numerical values that signify the confidence the agent has that the the given password is the correct one for the service. Successful and unsuccessful log in attempts affect these beliefs. Also, agents slowly forget passwords as we will discuss.

Cognitive Burden As the agent accrues accounts and expends mental effort to remember passwords for those accounts, her cognitive burden increases. The cognitive burden is a measure of the amount of memory the agent devotes to remembering passwords and the associations between passwords and services. A user who cannot cope with remembering many passwords may opt to write down or reuse passwords. The cognitive burden allows us to model such phenomena. Creating accounts, resetting passwords, and forgetting passwords affect the cognitive burden, which, in turn, affects the agent's willingness to circumvent.

Forgetting Passwords Agents forget passwords according to the ACT-R model as discussed in Pavlik and Anderson (2005) where forgetting rates grows logarithmically. Each piece of information is stored as a trace of memory and if it is not used it is slowly forgotten. However, if the trace is "refreshed" (e.g., because the password is used for the given service) then the forgetting rate reduces.

Creating Accounts If the agent decides to interact with a service for which she does not have an account, she will try to create an account (refer to Figure 2 for an overview of this process). This involves creating a username and password that the service accepts. For simplicity, the agent uses the same username for all her accounts. The agent then chooses a password from a predefined password list of approximately 30 passwords that are ordered by complexity. This password list mimics the sorts of passwords that a user– one who chooses a base password and modifies that base password to satisfy different password composition requirements–might use. It is not a predefined password list in the sense of being the top 100 passwords from a password breach.

So long as as the agent's cognitive burden does not exceed a threshold called the password reuse threshold, the agent chooses the simplest new password from the list that satisfies the requirements the agent knows about. If the agent's cognitive burden exceeds the threshold, she will reuse either the simplest or most complex existing password that she knows depending on the value of an input parameter. We believe this is a good approximation to how users actually behave, but it could be improved by employing an objective function that incorporates cognitive effort and fear of loss as mentioned in Florêncio, Herley, and Van Oorschot (2014b) instead of just effort. This is discussed further in Section 8.

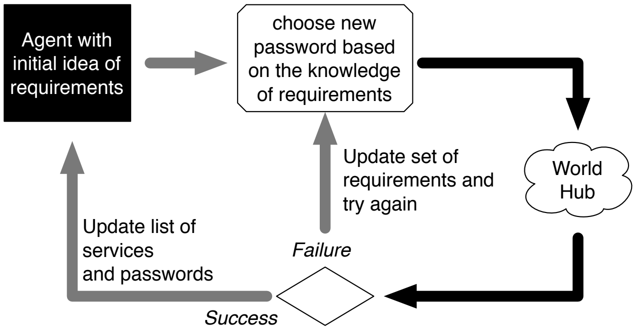
Agent with initial idea of requirements
choose new password based on the knowledge of requirements
Update set of requirements and try again
World Hub
Update list of services and passwords
Failure
Success

Figure 2: An agent creating an account.

If the password satisfies the requirements of the service and the account is successfully created, the agent will update its list of passwords, the cognitive burden associated with maintaining her password portfolio, and set the password beliefs for the service. If the password does not satisfy the requirements of the service, the agent updates the password requirements according to the new information received from the service and repeats the same procedure, trying to create a password that satisfies this new set of requirements. The agent may opt to reuse or write down the password to cope with the cognitive burden associated with maintaining a large password portfolio. We will later discuss these forms of circumvention in some detail.

Logging In Once the agent has an account for the given service, she can log in to it (refer to Figure 3 for an overview of this process). If she wrote down her password, she will simply use it. Else, she chooses the password with highest associated strength according to her password beliefs. The log in attempt may either succeed or fail; the outcome affects password belief strengths and forgetting rates. If the log in attempt fails, she will repeatedly attempt to log in with the password with highest strength until all password beliefs have strengths below a threshold called the password recall threshold.

Resetting Password If the agent fails to log in because she did not write down her password and all password belief strengths are below the password recall threshold, then she will reset her password. The process for resetting a password mimics that of creating a password for an account.

Logging Out If the agent is logged in to a service, she simply clicks a button to log out of the service.

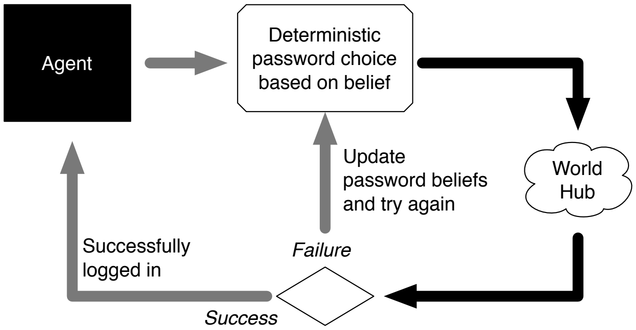
Agent
Deterministic password choice based on belief
World Hub
Update password beliefs and try again
Failure
Successfully logged in
Success

Figure 3: An agent logging in to an account.

The World Hub The world hub is responsible for creating services, maintaining account information for all accounts on all services, producing results for agent actions, and making security assessments.

When the simulation begins, the world hub generates a number of services grouped by importance to the user; this ordering coincides with the ordering by complexity of password composition policies (weak, average, good, strong) as suggested by Florêncio, Herley, and Van Oorschot (2014a). When an agent wishes to create an account it consults the username and password composition policies to ensure the proposed username and password satisfies them. Each service maintains a database comprising a list of usernames and passwords, one for each user's account. Also, the world hub keeps track of whether or not agents are currently logged in. When an agent sends an action to the world hub, it processes the action, taking the aforementioned state into account, produces a result that it sends back to the agent, and updates world state accordingly.

The world hub also makes security assessments for each account and a global assessment derived from those individual assessments. These assessments incorporate susceptibility to three forms of attack. The first is a brute force attack wherein the attacker simply guesses the account password. The second is a reuse attack wherein the attacker manages to compromise another account that uses the same password as the one used for the target account. The attacker then reuses that password to access the target account. In the third attack, the attacker finds the password written down and uses it to log in to the victim's account. The global security measure M that we use is based on the aggregate security threat posed to all accounts via all attack vectors.

Writing Down and Reusing Passwords Every password has a bit complexity according to the NIST password guidelines for password strength of human chosen passwords (Burr and others 2013). To cope with the cognitive burden of managing a large password portfolio, agents may circumvent by writing down passwords and reusing passwords from other services. Two parameters serve as thresholds for user engagement in circumvention. One threshold determines the cognitive burden over which the agent will reuse passwords; the other determines the cognitive burden over which the agents will write down passwords.

If, while creating a new password, an agent's cognitive burden exceeds the password reuse threshold, the agent will pick a password from the existing password portfolio based on a binary input parameter that specifies whether the agent will use the least or the most complex password. Performing this act of circumvention greatly reduces the cognitive burden associated with memorizing a completely new password that is not associated with any of the existing accounts. Of course, the tradeoff here is security; specifically, this form of circumvention increases susceptibility to a reuse attack.

Similarly, if the cognitive burden exceeds the password write threshold during account creation or a password reset, then, right after creating the password the agent will write down the password. Doing so allows her to bypass the password recall process for that account. The tradeoff is that an attacker can steal the document on which the password was written and perform a targeted attack.

The Stopping Condition Agent behaviors that we are particularly interested in observing involve creating accounts, reusing passwords, writing down passwords, and resetting passwords. When the forgetting rates for all services are below a certain threshold, we've reached a state where we expect no interesting behavior to emerge because the agent will no longer forget passwords. Hence, we stop the simulation.

5 Experimental Setup

The experimental goal was to verify our model. To do this we first parametrized our inputs to reflect real-world empirical data from the password security literature. We then ran our simulation and produced outputs, which we compared to other empirical data from the password security literature. We iteratively updated the remaining input parameters (i.e., the ones that we did not validate with real world data) until we had a simulation that produced results consistent with the empirical data.

In this paragraph we discuss the empirical data we used to parameterize the model. Our setup involved a single user interacting with the world hub. We set the password reuse threshold to 56 bits of memory and the password write-down threshold to 68 bits. As discussed earlier, the classification of services follows that of Florêncio, Herley, and Van Oorschot (2014a). There are four primary classes of services grouped by importance to the user; this ordered ranking doubles as a ranking by complexity of password composition requirements. The strongest services have the most variation in password composition requirements. We believe this holds in practice since services critical to the operation of large companies, banking services, and other highly sensitive services frequently change their password composition requirements. We assumed a uniform distribution of the service types. To best satisfy these constraints and the observation by Florêncio and Herley (2007) that "each user has about 25 accounts that require passwords," we set the number of services to 24, and we assigned 6 services to each of the four classes of services: weak, average, good, and strong.

The input parameters that are not validated by empirical data are discussed here. When an agent opts to reuse a password, she chooses the most complex password available in her existing password portfolio. The initial forgetting rate was set to 0.0025. The password recall threshold, which specifies the threshold over which users may consider a password when attempting to log in to a service was set to 0.5. That is, an agent may only recall a password for a service if the password strength exceeds this threshold of 0.5 for the service in question. A predefined list of passwords with varying complexity was used to mimic how we believe the user behaves.

We conducted 30 runs. The simulation stopped when the forgetting rate for every service was under 0.0005.

We sought to reproduce the password reset rates as derived from (Florêncio and Herley 2007), the number of accounts served by a password as derived from (Florêncio and Herley 2007), and the number of logins before a password is memorized (i.e., the number of logins before the stopping condition is met in our simulation), as found in Bonneau and Schechter (2014).

6 Validation

Our validation results are as follows:

- Florêncio and Herley (2007) found that 15% of login attempts on Yahoo involved a password reset. In our simulation the agent logged in on average 30.08 times over the 30 simulation runs before the stopping condition was met, which provides a baseline expectation of $0.15 * 30.088 = 4.51$ password reset attempts. Our simulation runs resulted in an average of 5.14 (SD = 0.95) resets per run. We do not find this to be a very reliable measure. This is because in our simulation all agents ultimately reach a stable state where they do not forget passwords, which results in the stopping condition being met. In practice, this is not the case; a person can use an account frequently but still forget her password after a long period of nonuse. Still, this provides a good first attempt toward validating password reset behavior.
- Florêncio and Herley (2007) also found that the average user maintains 6.5 passwords with each password being used approximately 3.9 times across 25 accounts. After scaling this result for 24 accounts, we expect to find that each unique password is used on average for 3.85 accounts across 24 accounts. In our simulation, we found that a password serves on average 4.26 accounts (SD=1.13).
- Bonneau and Schechter (2014) studied how long it takes users to memorize security codes. They found that "most participants learned their security codes early in the study, after a median of 36 logins (37 for letters and 33 of words)." In our simulation we found that it took agents 30.08 logins on average to reach the stopping condition, which serves as our indicator that the agent memorized the password.

Overall, we believe these first steps toward validation were successful. The results of simulation runs appear to corroborate the findings of the password security literature with which we compared it.

7 Sensitivity Analysis

To better understand the model, we conducted a one-factor-at-a-time sensitivity analysis of key parameters that were not validated by the literature. The purpose of conducting this sensitivity analysis was to observe how and to what degree the model's behavior changed as we varied input parameters. Comparing results of the analysis to our expectations enables us to improve our understanding of the correctness and accuracy of the model. To perform the analysis, we used the parameter settings provided in Section 5 as a baseline. Then, for each input parameter under consideration, we ran simulations over a range of values, keeping the other parameters consistent with this baseline. For every setting of parameter values we ran the simulation six times. Last, we plotted the results (note: error bars correspond to standard deviations).

7.1 Results

Initial Forgetting Rate Figure 4 illustrates the negative relationship between the initial forgetting rate and our security measure: the more frequently an agent forgets her password, the more likely she will be to circumvent. As expected, we also observed that as the forgetting rate increases, more passwords are written down and reset.

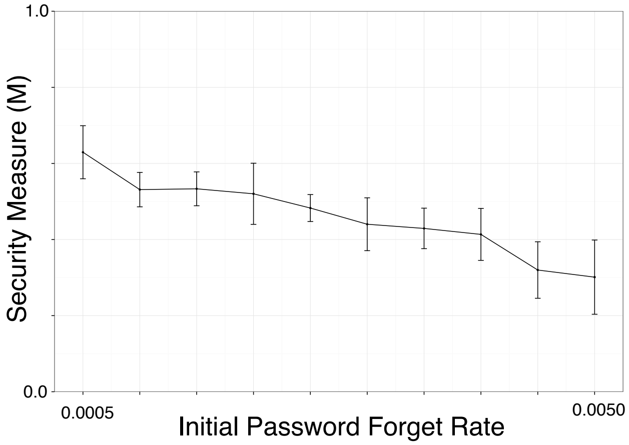

Figure 4: Sensitivity of Initial Password Forgetting Rate

Password Recall Threshold Figure 5 demonstrates that security drops as as we increase the recall threshold. Again, this is expected: the easier it is for an agent to remember her password, the less likely it will be that she circumvents.

Password Write Threshold Figure 6 resembles the graph from Kothari et al. (2015). The observed dip is due to the security tradeoff between writing down and reusing passwords.

Password Reuse Threshold Figure 7 is as expected: increasing the password reuse threshold improves security.

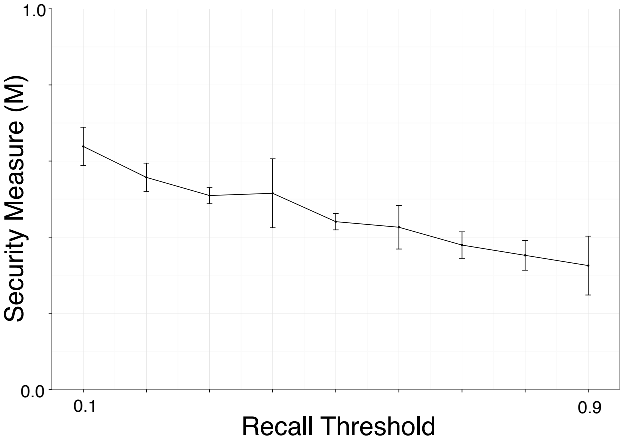

Figure 5: Sensitivity of Password Recall Threshold

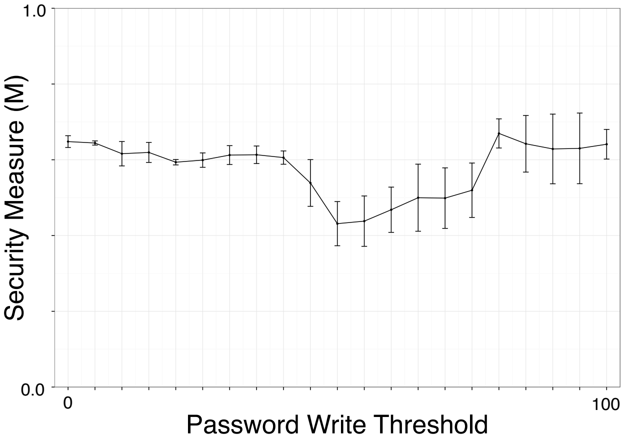

Figure 6: Sensitivity of Password Write Threshold

7.2 Takeaway

The sensitivity analysis largely agrees with the sensitivity analysis conducted for the initial model proposed by Kothari et al. (2015). We observed larger standard deviations, which we attribute to a more sophisticated forgetting model. Overall, we found we could logically justify the curves produced by varying input parameters, which is a good sign.

8 Conclusion and Future Work

We've discussed the applicability of agent-based simulations to security policy design, expanded our password simulation to more accurately evaluate the security afforded by password policies, and validated our model, to some extent, with empirical data from the password security literature. We believe the model holds promise, but there's still work to be done.

We aspire to further improve the model to reflect real-world behavior. For example, we are looking to incorporate password reuse findings from Das et al. (2014) into our simulation.

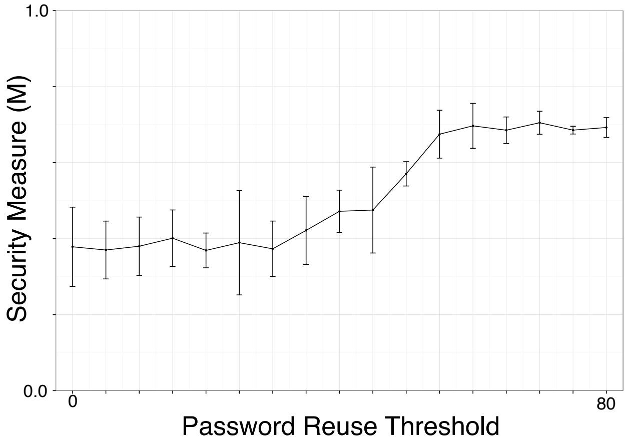

Figure 7: Sensitivity of Password Reuse Threshold

Anecdotes point to many avenues of potentially significant circumvention behavior, such as users sharing passwords when under time pressure, trying all their other passwords when a service rejects their password, using similar password families across different services, making trivial deltas (e.g. "fluffy1" to "fluffy2") when required to change passwords, or even setting all passwords to the same password string when forced to reset a password. In order to build these into the model in a way that can be validated, we need ground truth on when and how often these behaviors happen; some of this data can be derived from the empirical literature, but we are planning some experiments to get the rest.

Now that we can model one human, we plan to extend the model to support multiple interacting users (including sharing and use of common password strategies), and to model new security interactions across multiple sites and services.

To achieve the goal of helping security practitioners improve their policy decisions, we must understand the state of the art in actual policy design, the needs of policy designers, and what actually happens. One way we hope to bridge this gap is by conducting interviews.

In our simulation we are currently using a fixed set of passwords in increasing order of complexity that we believe simulates the types of passwords a user might choose. We seek to employ a more accurate model for the process of password generation and to plug it into our simulation.

We would also like to separate the world hub from the threat model. That is, coding different types of attacker agents may provide a new dimension to the simulation that improves both its accuracy and versatility. Moreover, it may be interesting to model how user behavior changes in response to attacks. For example, if an agent learns that a service she used was successfully attacked, would she change all her passwords that are the same or similar to the one used at the compromised service? Alternatively, if she hears about attacks on a regular basis, would she become indifferent toward password security as she perceives it to be out of her control?

Also, certain aspects of the model could be improved to make it more faithful to reality. For example, in the current state, users reuse passwords with a tendency towards complex or simple passwords as specified by an input parameter. Work by Florêncio, Herley, and Van Oorschot (2014b) suggests that the decision of what password to reuse may be more complex, involving an evaluation of loss-minimization and user effort. We plan to incorporate such ideas in future iterations of our simulation.

In conclusion, we've demonstrated through validation that agent-based simulation is a viable approach to gauging the efficacy of security policies in a real-world context. Our end goal is to further develop this simulation and ultimately create a usable and useful tool for security designers to assess the security afforded by their policy decisions. And our future work stems from this pursuit.

9 Acknowledgements

This material is based upon work supported by the Maryland Procurement Office under Contract No. H98230-14-C-0141. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Maryland Procurement Office. Koppel's work was supported in part by NSF CNS-1505799 & the Intel-NSF Partnership for Cyber-Physical Systems Security & Privacy. We also thank the anonymous reviewers for suggestions for future work.

References

Blythe, J.; Koppel, R.; and Smith, S. W. 2013. Circumvention of security: Good users do bad things. *Security & Privacy, IEEE* 11(5):80–83.

Blythe, J. 2012. A dual-process cognitive model for testing resilient control systems. In *Resilient Control Systems (ISRCS), 2012 5th International Symposium on*, 8–12. IEEE.

Bonneau, J., and Schechter, S. 2014. Towards reliable storage of 56-bit secrets in human memory. In *Proc. USENIX Security*.

Burr, W., et al. 2013. *Electronic Authentication Guideline*. NIST Special Publication 800-63-2.

Choong, Y.-Y. 2014. A cognitive-behavioral framework of user password management lifecycle. In *Human Aspects of Information Security, Privacy, and Trust*. Springer. 127–137.

Das, A.; Bonneau, J.; Caesar, M.; Borisov, N.; and Wang, X. 2014. The tangled web of password reuse. In *Symposium on Network and Distributed System Security (NDSS)*.

Florêncio, D., and Herley, C. 2007. A large-scale study of web password habits. In *Proceedings of the 16th International Conference on World Wide Web*, 657–666. ACM.

Florêncio, D.; Herley, C.; and Van Oorschot, P. C. 2014a. An administrator's guide to internet password research. In *USENIX LISA*.

Florêncio, D.; Herley, C.; and Van Oorschot, P. C. 2014b. Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts. In *Proc. USENIX Security*.

Gaw, S., and Felten, E. W. 2006. Password management strategies for online accounts. In *Proceedings of the Second Symposium on Usable Privacy and Security*, 44–55. ACM.

Jakobsson, M.; Tsow, A.; Shah, A.; Blevis, E.; and Lim, Y. 2008. What Instills Trust? A Qualitative Study of Phishing. *Lecture Notes in Computer Science* 4886:356.

Kothari, V.; Blythe, J.; Smith, S.; and Koppel, R. 2014. Agent-based modeling of user circumvention of security. In *Proceedings of the 1st International Workshop on Agents and CyberSecurity*, 5. ACM.

Kothari, V.; Blythe, J.; Smith, S. W.; and Koppel, R. 2015. Measuring the security impacts of password policies using cognitive behavioral agent-based modeling. In *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, 13. ACM.

Morley, D., and Myers, K. 2004. The spark agent framework. In *Autonomous Agents and Multi-agent Systems*.

Pavlik, P. I., and Anderson, J. R. 2005. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science* 29(4):559–586.

Renaud, K., and Mackenzie, L. 2013. Simpass: Quantifying the impact of password behaviours and policy directives on an organisation's systems. *Journal of Artificial Societies and Social Simulation* 16(3):3.

Riley, S. 2006. Password security: What users know and what they actually do. *Usability News* 8(1):2833–2836.

Shay, R.; Bhargav-Spantzel, A.; and Bertino, E. 2007. Password policy simulation and analysis. In *Proceedings of the 2007 ACM Workshop on Digital identity management*, 1–10. ACM.

Smith, S. W., and Koppel, R. 2014. Healthcare information technology's relativity problems: a typology of how patients' physical reality, clinicians' mental models, and healthcare information technology differ. *Journal of the American Medical Informatics Association* 21(1):117–131.

Smith, S. W.; Koppel, R.; Blythe, J.; and Kothari, V. 2015. Mismorphism: a Semiotic Model of Computer Security Circumvention. In *Ninth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2015)*, 172–182.