Password Policy Languages: Usable Translation from the Informal to the Formal

Michelle Steves[1](✉), Mary Theofanos[1], Celia Paulsen[1],
and Athos Ribeiro[2]

[1] National Institute of Standards and Technology, Gaithersburg, MD, USA
{michelle.steves,mary.theofanos,
celia.paulsen}@nist.gov
[2] Universidale de Brasilia, Brasilia, Brazil
athosribeiro@gmail.com

Abstract. Password policies – documents which regulate how users must create, manage, and change their passwords – can have complex and unforeseen consequences on organizational security. Since these policies regulate user behavior, users must be clear as to what is expected of them. Unfortunately, current policies are written in language that is often ambiguous. To tackle ambiguity, we previously developed a formal language for stating what behavior is and is not allowed regarding password management. Unfortunately, manual translation of the policy to this formal language is time consuming and error prone. This work focuses on providing an interface for policy users to generate accurate models of their interpretations of a password policy. This will aid password policy research, formalization, and ultimately more usable password policies. This paper describes the requirements, design, high-level application features, application validation, user testing, and includes a discussion of how this work is expected to progress.

Keywords: Usable security · Password policy · Question-answer system · Policy workbench · Formal language · XML

1 Introduction

As part of the research and development thrust of the United States' Comprehensive National Cyber-Security Initiative, we undertook an exploration of the relationship between usability and security in password policies. Previous work in this area includes development of a formal language for representing the rules governing user behavior surrounding password creation and maintenance [1]. Having developed a prototype formal representation of password policy rules, we next undertook an effort to map real-world policies onto the formal rule set to validate it and gather users' interpretations of actual password policies. Such a mapping for any policy produces a model of the policy that can be used by analytical tools, including a policy workbench – a set of tools and methods that is used to develop, analyze, and improve policies [2]. Earlier efforts of model generation for policy workbenches have shown only limited success, with resulting models being error-prone, difficult and time consuming to produce [3–6].

T. Tryfonas and I. Askoxylakis (Eds.): HAS 2015, LNCS 9190, pp. 119–130, 2015.
DOI: 10.1007/978-3-319-20376-8_11

These approaches did not show promise in aiding us meet our objective of easily and accurately mapping a policy user's interpretation of a password policy to its formal representation.

We looked beyond these translation methods for an approach that would reduce the learning curve and effort required of the human translator to produce an accurate model. Use of an on-line question-answer system was explored and piloted. While the overall approach of a question-answer system showed promise, our hard-coded prototype made the application impossible to modify without programmer intervention, which would be untenable during the highly-iterative process envisioned to validate the application's user-facing elements, e.g., question and answer wording. Further, we needed the ability to make changes to rules in the underlying set of formal password policy statements as we validate it. A review of question-answer applications showed that no existing applications met our broadest requirement: to construct a set of statements (rules from our formal language) from user-selected, predefined responses to our predefined questions. Therefore, a custom application for our specific requirements was developed. The resulting application provides us with a tool that is highly usable and configurable, including researcher specified content, to present questions and answers to users from which a user's responses are used by the application to construct a set of syntactically correct formal statements that constitutes the user's model of a password policy. Preliminary results show that users with no experience can produce an accurate translation of a real-world password policy in less time and with fewer errors than an experienced translator using the manual translation method.

Why focus on the policy user? Policy users, the end users of policy, are critical components in the ultimate effectiveness of any password policy. A typical user might be governed by multiple policies both at work and at home (e.g., to access corporate email servers, personal financial information, medical records, and e-commerce sites). Ambiguities in these policies, discrepancies between them, and the sheer number of different policies may cause confusion. As a result of this cognitive burden, users may choose weak passwords, write them down, or violate policies in other ways. Consequently, overall security may be weakened. At a minimum, users must be clear as to what is expected of them for a policy to be effective, since these policies attempt to govern user behavior. Ambiguity contained in policies and users' misinterpretation of policy can have complex and unforeseen consequences on organizational security.

Previously, the involvement of policy users in the interpretation of policies to produce formal language models has had a low return relative to the investment of resources needed to produce these models. This was due to the learning curve for policy users associated with manual translation and methods using quasi-natural language approaches, coupled with the error rates in the resulting models. The application described in this paper is anticipated to collect policy user interpretations to generate accurate models of those interpretations to aid password policy research, formalization, and ultimately more usable password policies. This paper describes the broader requirements that drove the requirements, high-level application features, validation of the application, and includes a discussion of how this work is expected to progress.

2 Background

Our overarching goal is to resolve the problem of password policy ambiguity by developing methods and tools for studying and clarifying policy statements. Thus this effort is grounded in research in password usability, password policies, and usability of automatically processing policy texts for the extraction and representation of knowledge and rules. Previous work in each of these research areas will be addressed.

Much research has focused on characterizing passwords that people employ, including [7–11] to name a few. These works rely on lists of passwords and provide insight into the actual passwords that people choose. Since users' password choices are governed in part by policy requirements, these studies are related to our research in password policies.

The relationship between password policy and user behavior has also been studied. Mannan and Oorschot [12] surveyed users of online banks and their understanding of bank's security requirements. They found disconnects between user practice and the banking guidelines. Furnell [13] examined the password-creation guidelines, the enforcement of password-composition restrictions and the reset policy of 10 website policies. In a survey regarding password usage of 32 staff members at a research university and a financial-services organization, Inglesant and Sasse [14] found that password policies which ignore human factors may result in unexpectedly poor security. Choong et al. [15] surveyed approximately 5,000 United States federal government staff members and found that users' password experiences were significantly influenced by password policies.

Others have specifically examined password policies and their content. Summers and Bosworth [16] described what should be in a password policy arguing for user guidance and organizational enforcement of passwords. Spafford [17] argued that the best practices shared by many modern password policies are actually artifacts based on out-of-date risk assessments. Based on his number and use of passwords, Farrell [18] argued that policy writers must acknowledge the increasing burden of password management on their users. Bonneau and Preibusch [19] surveyed the empirical password policies of 150 websites and were able to infer the enforced lengths and complexity requirements by creating accounts and systematically changing the password. Florêncio and Herley [20] surveyed length and complexity requirements for 75 websites according to their password policies and determined that sites that depend on users for revenue have weaker, more accommodating policies. Komanduri et al. [21] performed a large scale study that examined password composition policy and users' password choices. They found that increases in entropy of passwords correlate with decreases in usability but believe that policies can be optimized for entropy and usability.

Some research has attempted to make writing security policies easier. Xu et al. [22] presented a visualization of access-control policies. A list of guidelines for architecting a system for writing security and privacy policies is provided by Johnson et al. [23]. A few have focused specifically on writing password policies. The development of standardized password policies is proposed in [24]. A language for expressing a password-policy scenario in a formal language is presented by AlFayyadh et al., [25]. A measure of system harm resulting from the given scenario can be estimated using simulation. Parkin et al. [26] presented an ontological framework for reasoning about the security and usability costs of different policy decisions.

Finally, several tools have been architected for developing, studying and implementing policy. These systems are referred to as policy workbenches. Policy workbenches generally require that the policy makers interact through a quasi-natural language [2]. Many [2–5] have found that manual translation to the quasi natural language is time-consuming, error prone and thus not necessarily usable. Automated extraction attempts were also shown to be error prone. To address this concern, Michael et al. [2] developed an architecture for mapping policies submitted in a natural language to formats suitable for further processing. However, the approach still required grammar rules for the natural language to ensure accurate semantic interpretation. In [3–5] the researchers also developed a policy workbench for privacy management, again using a constrained natural language as the interface to the tool. They found the manual preprocessing of the policy was time consuming and impacted accuracy. Breaux and Antón [6] also examine policy management. They too use a structured natural language to identify actors, actions and objects in privacy policies.

While our research overlaps that of [19, 20] and the policy workbench efforts in [2–6], the focus of our work differs in important ways. We are focused on how policy users interpret password policies, highlighting the diversity and ambiguities in password policies so that policies can be strengthened. Our specific focus on policy users and their interpretations of password policies versus the machine interpretation of policies has led us to examine an alternative usability driven input approach to a policy workbench.

3 The Application

The functionality of the application centers on collecting an interpretation of a password policy from a user and generating the associated formal rule representation of that policy interpretation. The application design was based on requirements identified through envisioned use case scenarios and established usability principles. To enhance usability of the application, a usability expert reviewed the interface design prior to development. Multiple types of validation testing of the application were performed to ensure effective functionality and usability by the intended user populations. The requirement concerns, system architecture and validation testing are described in this section.

3.1 Application Requirements Considerations

Three main use cases were identified that drove the requirements specification.

1. Collect a policy user's interpretations of the rules in a password policy document and then translate those interpretations into a formal representation of the policy. This yields a model of the policy user's interpretations of the selected password policy. The target user population includes password policy users, specifically those without special password policy expertise. Policy user models are submitted anonymously.

2. Collect a password policy expert's interpretations of the rules in a password policy document and then translate those interpretations into a formal representation of the policy. This yields a model that can be used as the 'ground truth' representation of the selected password policy during analysis. These models must be distinguishable from those produced by policy users. The target user population includes password policy researchers and policy makers.
3. System administration to configure and maintain application software and operations, including configuring and installing new question-answer sets. It was expected that most users having a role of researcher will also have an administrator role for the application.

Finally, the application must support a flexible method for specification of the user-facing content and the formal language onto which user interpretations are mapped.

3.2 Implementation: Architecture and Features

From the requirements considerations related in the previous section, the use of a question-answer system approach was selected to collect user translations of existing policies. The question-answer approach has the advantage that questions and response choices can be crafted such that the mechanics of the translation to the formal rule set are not readily evident to non-expert translators and avoid having translation move beyond the intended scope of the formal rule space. Figure 1 shows an overview of the architecture.

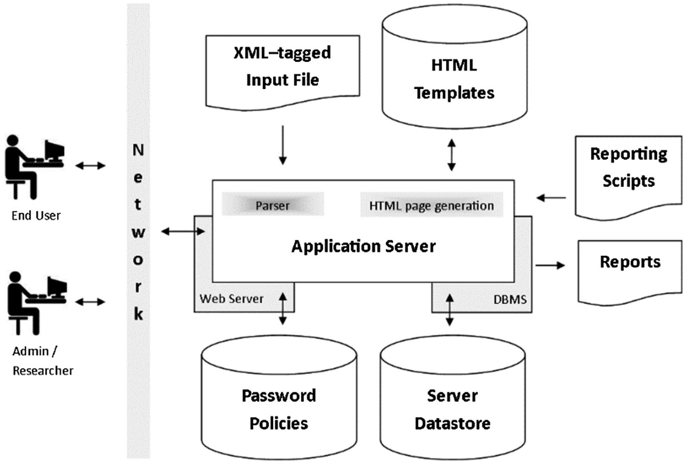


Fig. 1. System overview

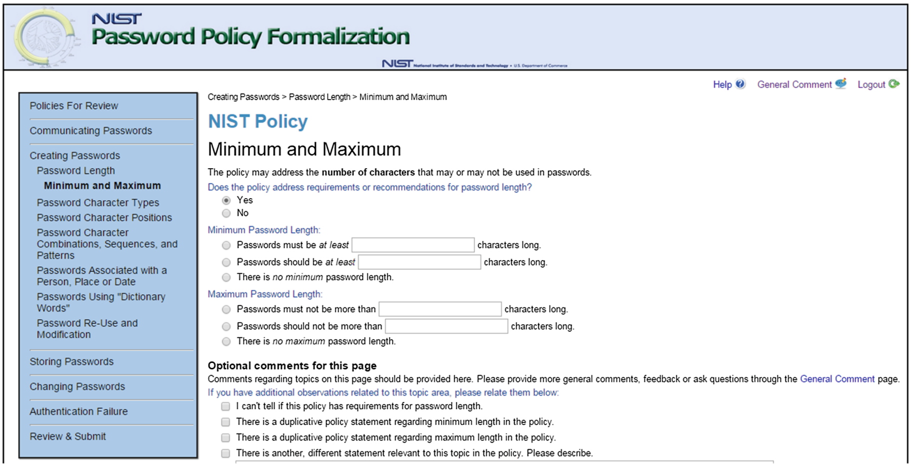
In our implementation, policy users and policy makers access the system via a web-based, client-side interface. Once an account is authenticated, the user chooses which policy to review and a browser window provides a document containing the pertinent password policy, while another browser window presents questions and response choices. The user answers each question presented and then completes the review of a password policy. All responses and their corresponding formal language statements are stored by a server-side application. The review can be suspended and resumed at any time by the user.

On the server side, an eXtensible Markup Language (XML) and JavaScript Object Notation (JSON) -tagged input file provides most of the user-facing content used by the system. The tags specify questions, response options, comment solicitation, placement and type of user input, e.g., radio button, textbox options, etc. Further, a mapping of responses to formal statements, along with other user-facing content in the interface, such as text for alert notifications and help page are also specified in this file. The content for each user-facing page of the application is pulled from the XML file by a parser and populated onto the pertinent HyperText Markup Language (HTML) template page.

The application server includes the XML file parser, HTML templates, and a file repository of password policies. Further, the application server uses a web server package and a document-oriented, database management system (DBMS) to manage the datastore containing user account data and policy reviews. The DBMS system chosen supports dynamic schemas, which provides the needed flexibility to support the XML-defined, data-driven nature of the application. The application also accesses a directory of files, each containing a password policy. Finally, a set of scripts is employed to extract data from the datastore to generate spreadsheets and reports used in data analysis by researchers outside of the application.

Figure 2 shows a sample page from the question set regarding questions about minimum and maximum length rules when creating passwords.

The following code segment shows an example the XML-tagged data used to produce the question-answer set for the application. The code segment shown was used to create the first two questions shown in the screen capture contained in Fig. 2.


NIST
Password Policy Formalization
NIST National Institute of Standards and Technology • U.S. Department of Commerce
Help
General Comment
Logout
Policies For Review
Communicating Passwords
Creating Passwords
Password Length
Minimum and Maximum
Password Character Types
Password Character Positions
Password Character Combinations, Sequences, and Patterns
Passwords Associated with a Person, Place or Date
Passwords Using "Dictionary Words"
Password Re-Use and Modification
Storing Passwords
Changing Passwords
Authentication Failure
Review & Submit

Creating Passwords > Password Length > Minimum and Maximum

NIST Policy

Minimum and Maximum

The policy may address the **number of characters** that may or may not be used in passwords.

Does the policy address requirements or recommendations for password length?

● Yes
○ No

Minimum Password Length:

○ Passwords must be *at least* ______ characters long.
○ Passwords should be *at least* ______ characters long.
○ There is *no minimum* password length.

Maximum Password Length:

○ Passwords must not be more than ______ characters long.
○ Passwords should not be more than ______ characters long.
○ There is *no maximum* password length.

Optional comments for this page

Comments regarding topics on this page should be provided here. Please provide more general comments, feedback or ask questions through the General Comment page.

If you have additional observations related to this topic area, please relate them below:

□ I can't tell if this policy has requirements for password length.
□ There is a duplicative policy statement regarding minimum length in the policy.
□ There is a duplicative policy statement regarding maximum length in the policy.
□ There is another, different statement relevant to this topic in the policy. Please describe.

Fig. 2. Example of the user interface

```
<question id="q.2.0">
   <text>Does the policy address requirements or recom-
mendations for password length?</text>
   <response type="select one">
      <option id="q.2.0.A">
         <text>Yes</text>
      </option>
      <option id="q.2.0.B">
         <text>No</text>
      </option>
   </response>
</question>
<question id="q.2.1" display_when="q.2.0.A">
   <text>Minimum Password Length:</text>
   <response type="select one">
      <option id="q.2.1.A">
         <text type="json">Passwords must be at least
            {"cloze": {"validation": {"min": "1"},
             "type": "numerical", "id": "q.2.1.A.a"}}
             characters long.</text>
         <BNF_mapping id="b.35" type="json">Users must
             create passwords with length greater than or
             equal to {"insert": {"qref": "q.2.1.A.a"}}
             characters</BNF_mapping>
     </option>
     <option id="q.2.1.B">
        <text type="json">Passwords should be at least
           {"cloze": {"validation": {"min": "1"}, "type":
            "numerical", "id": "q.2.1.B.a"}} characters
            long.</text>
        <BNF_mapping id="b.36" type="json">Users should
            create passwords with length greater than or
            equal to {"insert": {"qref": "q.2.1.B.a"}}
            characters</BNF_mapping>
     </option>
     <option id="q.2.1.C">
        <text>There is no minimum password length.</text>
     </option>
   </response>
</question>
```

Finally, a second interface is provided for system administrators to set various system parameters such as the number of policies policy users are assigned to review, privileged account management, and similar functions. This interface is served to the client when a user with the appropriate access right authenticates to the system. Users in the role of researcher can use this mode to select a particular policy to review with the same question-answer set that is served to policy users.

3.3 Application Validation

A validation plan was developed to establish that the application and its' question-answer sets are usable and working as intended before the system was deployed to ensure accurate model generation. While identifying validation objectives, efforts were made to isolate concerns to make testing more effective. Identified objectives fall into two broad categories: those for the application itself and those for the data contained in the XML input file, e.g., the wording of the questions and responses. Validation of the system need only be performed once; however, each new input file or one with substantive changes should be validated before deployment. At the writing of this paper, we have completed the first phase of validation – assessing the mechanics of the application with respect to functional correctness and usability objectives.

Validation Objectives. The following validation objectives were excerpted from those identified for the application and its behaviors for any question-answer set:

- The application correctly utilizes the XML input file, e.g., recognizes all defined tags, utilizes tagged data correctly, and renders user-facing content correctly.
- The application correctly captures user responses, including user modifications to previously answered questions.
- For each policy review, the application correctly constructs and returns formal statements based on user responses, and correctly identifies formal statements where the reviewer did not supply all of the responses necessary to completely construct a formal statement.
- The application is usable for intended users.

For question-response sets specifically, the following validation objectives were identified:

- All user-facing content is usable, e.g., question-answer layouts, wording of questions, answers, alerts, button labels, and so on.
- The mapping of responses to formal statements is correctly specified, complete, with no redundancies or inconsistencies.

Validation Strategy. Testing efforts were generally divided along the lines defined by validation objectives, first validating the application and then the input file. Phase I of the strategy tackles the objectives for the application and its behaviors for any question-answer set. Phase II addresses test objectives specifically for question-answer sets.

Phase I: Validating the application. This first phase incorporated many types of tests ranging from black box functionality tests to usability testing of the interface components. Iterative work with the developer addressed issues that were found while testing progressed. Multiple input files were needed for this phase of testing. To assist in developing these input documents, one team member developed a parser which processes minimally-tagged input data and produces an XML file suitable for input to the application. This parser greatly facilitated the process of producing fully-tagged input files while reducing the syntax and typographical errors that would likely be produced if these files were manually generated.

Once the application was assessed to be behaving as expected with respect to the design specification of the XML tag set, usability tests were incorporated. A set of usability tests was developed and used to assess the user's ability to understand the workflow to review a policy, e.g., how to start a policy review, submit a review, and other application mechanics such as navigation within the application, leaving feedback, getting help, and so on. A draft version of the question-answer set and real password polices were used by members of the target end user population to help assess the usability of the application.

Phase II: Validating the input file. During Phase II, testing efforts will focus on assessing the wording of the questions, response options, button labels, alert messages and other user-facing content. To start, the input file will undergo an expert review to ensure that plain language [27] is used throughout and iterative refinements will be made as testing result data are analyzed. Testing will also assess the usability of different question-answer structures and users' feedback on those structures. Finally, the mapping of responses to formal language statements must be assessed for completeness and correctness.

4 Discussion

This work arose from a need to deal with ambiguities in current password policies. These ambiguities not only prevent researchers from comparing and contrasting policy statements, they also could cause users to misinterpret what is expected of them. To tackle ambiguity, we developed a formal language for stating what behavior is and is not allowed when creating, managing, and changing passwords. After developing our formal policy representation, we had several practiced translators perform manual translations of actual password policies. We found that manual translation of existing policies to the formal language to be time consuming and that one translator's set of formal language rules for a given policy might differ from another translator's results. Examination of the differing rules by these translators yielded explanations for the variation which fell into the following categories: (1) translator misread the source policy, (2) translator made a simple syntax error constructing the formal statement, (3) translator had imperfect understanding of the formal rule scope, and (4) translator had difficulty interpreting the source policy in an unambiguous way.

Karat et al. [5] when evaluating privacy rule authoring methods also found that policy makers had trouble writing policy rules without some form of guidance. Performing a usability test comparing interfaces using natural language with guidance, structured lists, and unassisted natural language they found that users performed better with guidance. Users performed faster, preferred the interface, and the quality of the resulting rule was higher with the assisted natural language interface. This encouraged us to develop a question and answer interface that assisted users in translating existing password policies into their formal representation. This user guided approach to translation specifically reduces translator variability due to simple syntax errors constructing the formal statement and translators' understanding of the rule scope.

Phase I testing has demonstrated that the application behaved as expected with respect to the design specification of the XML tag set. Application testing was followed by usability tests for the application. A set of usability tests was developed and used to assess the user's ability to understand the workflow to review and translate a policy, e.g., how to start a policy review, submit a review, and other application mechanics such as navigation within the application, leaving feedback, getting help and so on. A draft version of the question-response set along with real password polices were used by members of the target end user population to help assess the usability of the application. Users easily navigated through the system, followed the workflow, initiated policy translations and submitted translations. User feedback was positive and users felt confident that they could use the system to review and translate password policies.

Now that the application is validated and usable, our user testing will transition to focus on how users interpret the wording of questions and answers. Once this phase is completed, users will use the application to build a collection of translated password policies. This collection will help validate the formal language as well as reveal how policy users interpret current, natural language password statements and which statements lead to misinterpretation within a policy. This analysis results in the identification of ambiguities in current password polices leading to a better understanding of what makes a policy usable from a user perspective. Understanding the usability impact of individual statements and combinations of statements allows organizations to author policies that better support their security concerns.

Future research includes three objectives. The first is translating the formal representation of each statement in a policy to its plain language equivalent and assessing how users interpret a policy presented in a plain language format. Providing policy developers and experts a way to express an existing password policy in a plain language representation is our second goal. Our final goal is to provide policy developers and experts with analysis of security and usability factors for individual and combinations of rules within a policy. We believe that the insights provided by this user-centered approach will enable policy developers and maintainers to improve their password policies, which will, in turn, improve organizational security and the user experience.

Acknowledgements. The authors would like to acknowledge the contributions by James Foti, at the National Institution of Standards and Technology, for his work on the draft question-answer set used during validation of the application. His use of plain language in the text of the questions and answers helped differentiate concerns about the application from concerns about the wording of questions and answers during an early phase of the application's validation. Additionally, the authors would like to acknowledge the contributions of Susanne Furman, also at the National Institute of Standards and Technology, for her contributions to enhance the usability of the application, both during review of the initial design and usability testing.

References

1. Killourhy, K., Choong, Y., Theofanos, M.: Taxonomic rules for password policies: translating the informal to the formal language. Internal report 7970, National Institute of Standards and Technology, Gaithersburg, Maryland (2013)

2. Michael, J.B., Ong, V.L., Rowe, N.C.: Natural-language processing support for developing policy-governed software systems. In: 39th IEEE International Conference and Exhibition on Technology of Object-Oriented Languages and Systems, pp. 263–274. IEEE Press, New York (2001)
3. Brodie, C., Karat, C.M., Karat, J., Feng, J.: Usable security and privacy: a case study of developing privacy management tools. In: ACM 2005 Symposium on Usable Privacy and Security, pp. 35–43. ACM Press, New York (2005)
4. Brodie, C.A., Karat, C.M., Karat, J.: An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In: ACM 2006 Symposium on Usable Privacy and Security, pp. 8–19. ACM Press, New York (2006)
5. Karat, C.M., Karat, J., Brodie, C., Feng, J.: Evaluating interfaces for privacy policy rule authoring. In: ACM 2006 SIGCHI Conference on Human Factors in Computing Systems, pp. 83–92. ACM Press, New York (2006)
6. Breaux, T.D., Antón, A.I.: Deriving semantic models from privacy policies. In: Sixth IEEE International Workshop on Policies for Distributed Systems and Networks, pp. 67–76. IEEE Press, New York (2005)
7. Morris, R., Thompson, K.: Password security: a case history. Commun. ACM **22**(11), 94–597 (1979). ACM Press, New York
8. Klein, D.V.: Foiling the cracker: a survey of, and improvements to, password security. In: 2nd USENIX Security Workshop, pp. 5–14. USENIX, Berkeley (1990)
9. Wu, T. D.: A real-world analysis of kerberos password security. In: 1999 Network and Distributed Systems and Security Symposium. Internet Society (1999)
10. Florencio, D., Herley, C.: A large-scale study of web password habits. In: 16th ACM International Conference on World Wide Web, pp. 657–666. ACM Press, New York, (2007)
11. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: an empirical analysis. In: 30th IEEE INFOCOM, pp. 1–9. IEEE Press, New York (2010)
12. Mannan, M., van Oorschot, P.C.: Security and usability: the gap in real-world online banking. In: 2007 ACM Workshop on New Security Paradigms, pp. 1–14. ACM Press, New York (2008)
13. Furnell, S.: An assessment of website password practices. Comput. Secur. **26**(7), 445–451 (2007). Elsevier, Amsterdam
14. Inglesant, P. G., Sasse, M. A.: The true cost of unusable password policies: password use in the wild. In: SIGCHI 2010 Conference on Human Factors in Computing Systems, pp. 383–392. ACM Press, New York (2010)
15. Choong, Y.Y., Theofanos, M., Liu, H.K.: United States Federal Employees Password Management Behaviors a Department of Commerce Case Study. Internal report 7991, National Institute of Standards and Technology, Gaithersburg, Maryland (2014)
16. Summers, W. C., Bosworth, E:. Password policy: the good, the bad, and the ugly. In: WISICT 2004, Winter International Symposium on Information and Communication Technologies, pp. 1–6. Trinity College, Dublin (2004)
17. Spafford, E: Security Myths and Passwords. In: CERIAS Blog, 19 April 2006. http://www.cerias.purdue.edu/site/blog/post/password-change-myths/. Accessed Feb 2015
18. Farrell, S.: Password policy purgatory. IEEE Internet Comput. **12**(5), 84–87 (2008)
19. Bonneau, J., Preibusch, S.: The password thicket: technical and market failures in human authentication on the web. In: 9th Workshop on the Economics of Information Security (2010). http://weis2010.econinfosec.org/papers/session3/weis2010_bonneau.pdf. Accessed Feb 2015
20. Florêncio, D., Herley, C.: Where do security policies come from? In: 6th ACM Symposium on Usable Privacy and Security, article 10. ACM Press, New York. (2010)

21. Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Egelman, S.: Of passwords and people: measuring the effect of password-composition policies. In: 2011 SIGCHI Conference on Human Factors in Computing Systems, pp. 2595–2604. ACM Press, New York (2011)
22. Xu, W., Shehab, M., Ahn, G.J.: Visualization based policy analysis: case study in Selinux. In: 13th ACM Symposium on Access Control Models and Technologies, pp. 165–174. ACM Press, New York (2008)
23. Johnson, M., Karat, J., Karat, C.M., Grueneberg, K.: Optimizing a policy authoring framework for security and privacy policies. In: 6th ACM Symposium on Usable Privacy and Security, article 8. ACM Press, New York (2010)
24. AlFayyadh, B., Thorsheim, P., Jøsang, A., Klevjer, H.: Improving usability of password management with standardized password policies. In: 7eme Conférence sur la Sécurité des Architectures Réseaux et Systemes d'Information, 7th Conference on Network and Information Systems Security, SAR SSI 2012. https://sarssi2012.greyc.fr/wp-content/uploads/SAR-SSI-2012_p38-45_AlFayyadh.pdf. Accessed Feb 2015
25. Shay, R., Bhargav-Spantzel, A., Bertino, E.: Password policy simulation and analysis. In: 2007 ACM Workshop on Digital Identity Management, pp. 1–10. ACM Press, New York (2007)
26. Parkin, S.E., van Moorsel, A., Coles, R.: An Information security ontology incorporating human-behavioural implications. In: 2nd International Conference on Security of Information and Networks, pp. 46–55. ACM Press, New York (2009)
27. What is plain language? http://www.plainlanguage.gov/whatisPL/. Accessed on Feb 2015