A Study of Online Security Practices

by

Garrett Gutierrez

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2017 by the
Graduate Supervisory Committee:

Gail-Joon Ahn
Rida Bazzi, Chair
Adam Doupe

ARIZONA STATE UNIVERSITY

August 2017

ABSTRACT

Data from a total of 282 online web applications was collected, and accounts for 230 of those web applications were created in order to gather data about authentication practices, multistep authentication practices, security question practices, fallback authentication practices, and other security practices for online accounts. The account creation and data collection was done between June 2016 and April 2017. The password strengths for online accounts were analyzed and password strength data was compared to existing data. Security questions used by online accounts were evaluated for security and usability, and fallback authentication practices were assessed based on their adherence to best practices. Alternative authentication schemes were examined, and other security considerations such as use of HTTPS and CAPTCHAs were explored. Based on existing data, password policies require stronger passwords in for web applications in 2017 compared to the requirements in 2010. Nevertheless, password policies for many accounts are still not adequate. About a quarter of online web applications examined use security questions, and many of the questions have usability and security concerns. Security mechanisms such as HTTPS and continuous authentication are in general not used in conjunction with security questions for most web applications, which reduces the overall security of the web application. A majority of web applications use email addresses as the login credential and the password recovery credential and do not follow best practices. About a quarter of accounts use multistep authentication and a quarter of accounts employ continuous authentication, yet most accounts fail to combine security measures for defense in depth. The overall conclusion is that some online web applications are using secure practices; however, a majority of online web applications fail to properly implement and utilize secure practices.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Background

Web services and applications are increasingly popular with users of the internet, and this popularity creates a drive for personalized experiences for users. This is reinforced by the fact that social media services like Facebook and LinkedIn provide a platform to represent a person's identity online, which can reflect various aspects of their physical identity. Consequently, a user's account can hold sensitive or valuable information that they may not want to be accessible to others. Hence, there is a need to secure web applications against improper access, which includes establishing account credentials that could be effectively and securely used to authenticate users.

However, there exists a perception that online web applications do not have adequate security. Over the past 17 years, there have been over 200 major data breaches for online accounts. Even well-known web services such as Yahoo!, Gmail, and Twitter have had data breaches in the last 7 years (Pagliery, 2013). The consequences of the breaches range from a few 100,000 accounts compromised by attackers to a few million compromised accounts, which often included financial information such as credit card numbers. Some of these attacks involve retrieving an offline copy of the database with users' hashed passwords and launching a brute-force password cracking attack. While these large-scale attacks are not against individual accounts, it is important to understand the security of authentication practices for individual accounts. In fact, stronger authentication mechanisms not only can protect against targeted attacks, but also can reduce the effects of large-scale breaches by using stronger passwords, continuous authentication, and multistep authentication. A major issue with authentication practices is that there have been documented problems with their implementation.

For example, as of this work, the most common method for web accounts to identify and login users, also called primary authentication, is the username and password scheme (Florencio & Herley, 2010). Passwords are notorious for having issues with usability (Adams & Sasse, 1999). Users tend to choose insecure passwords that are easier to remember (Florencio &

Herley, 2007). The passwords that are the easiest to use are typically weak. Passwords with common sequences of numbers and digits or that use dictionary words can be brute forced using programs such as John the Ripper (n.d). Making passwords stronger typically requires making the password more complex, which can make it more difficult for the user to remember their passwords. Even if users employ strong passwords, it is common for users to reuse the same password for multiple web services (Florencio & Herley, 2007). Therefore, if an attacker discovers the password for one web service, the attacker can then breach other web services where that same password is used. These considerations suggest that, overall, passwords have a poor balance of usability and security.

Another problem arises if a user forgets their credentials for primary authentication. The user can recover access to their account in a process called fallback authentication. One method that web services have used to authenticate a user during fallback authentication is via registered personal questions also known as security questions. Unfortunately, there have been documented problems with the implementation of security questions (Haga & Zviran, 1991). Depending on the type of security question chosen, the answer set can be small and therefore easily guessable. Other questions such as “What is your mother’s maiden name?” can be researched using public data or answered by an attacker who is a close friend or family member to the user (Griffith & Jakobsson, 2005). If the web service only requires a small number of security questions to regain access to the account and the questions are guessable, then an attacker can easily gain access to the user’s account as was the case in 2008 when vice presidential candidate Sarah Palin had her Yahoo! account compromised (Johnson, 2010).

While existing literature indicates that security practices that can be used for authentication and password recovery are not secure, the question that can be raised is the following: What is the current level of security for security practices that are actually used by web applications today? Existing works have not provided large-scale, empirical data that measures the degree of security for these practices that are currently implemented. Therefore, the main question that our work aims to answer is the following: How adequate are the security practices used by web applications for the purpose of protecting user accounts?

Problem

Our goal was to analyze the security practices implemented for various online web application in various categories. Specifically, we examine the types of authentication mechanisms for each account including the strength of their password policies, multistep practices, and continuous authentication mechanisms. In addition, we assess the implementation of fallback authentication for each account, especially the usage of security questions implemented by online web applications in practice. Finally, we use data about the other security mechanisms for each account to assess the combination of the security mechanisms with primary authentication, fallback authentication, and security question practices.

Existing research and literature has addressed some of these areas; however, the existing works were on smaller scales and focused on specific domains. Herley and Florencio (2010) examined the password policies for 75 online web applications in 2010. Their work examined a smaller number of web applications and made some assumptions about the value of web applications that may not reflect their actual value, such as assuming that university accounts do not hold as much value as financial accounts. Researchers such as Schechter et al. (2009) and Rabkin (2008) examined security questions; however, they limited their research to specific domain such as email providers and banks, respectively, and did not examine the use of a combination of security mechanisms, which can improve the overall security of security questions.

Our research addresses these gaps by gathering data for 282 online web applications. We created accounts for 230 allowing us to directly gather data about the policies as they are implemented. For the remaining accounts, we gathered data directly from policies without creating accounts due to the difficulty or impracticality of creating some accounts such as accounts with various banks or accounts at various universities. For all 282 applications, we examined the collected and posted password policies and analyzed their strength. We went a step further by comparing the strength of password policies for the online applications gathered by Herley and Florencio in 2010 to the strength of the password policies for those same accounts in 2017. By comparing the results, we can report the evolution of password requirements for online web applications. Beyond passwords, we examine multistep authentication mechanisms and continuous authentication mechanisms. In order to more effectively evaluate the security of the authentication mechanisms for each account, we not only considered the password strength, but we also took into consideration any requirement on the use of other authentication mechanisms in addition to passwords.

Of the 282 online web applications, 79 used security questions in some manner, and we were able to collect security question data for 47 accounts. We classified the questions from the 47 accounts into categories in order to assess their usability and resistance to attack. Furthermore, we examined and discussed the usage of security questions in practice. To bolster our analysis of the security of security questions, we examined the combination of the usage of security questions with the presence of multifactor authentication and usage of CAPTCHAs in order to determine the resistance of security questions to large-scale attacks.

Our work also addresses the fallback authentication practices used by online web applications. Using recommended best practices from existing works, we assess the security of the fallback authentication mechanisms based on their adherence to the recommended practices. We went further by collecting and examining other security features offered by the accounts. With the data for password practices, multistep practices, continuous authentication, security questions, and fallback authentication, we determined how web application use a combination of mechanisms to increase their security or fail to take advantage of security mechanisms. By collecting data on whether accounts store sensitive information such as social security numbers and credit cards, we assessed whether storing information related to monetary value impacts the usage of security mechanisms.

To our knowledge, no other work has performed an analysis of multiple security mechanisms for a large number of web applications. Hence, our work addresses the missing links for the usage and strength of multiple security practices for web applications. Our work can help identify poor implementations of security practices and allow us to recommend security practices for online accounts in order to help strengthen the security practices used on the web.

Summary of Results

We summarize our results for each security practice. For password policies, we found that from the 2010 data for password policies, there is an increase in password strength in 2017 for almost each web application category. Out of the seven hypotheses considered by Florencio and Herley, our results are different from theirs for three hypotheses. The differences in the results regarding the hypotheses also indicate that there is a strengthening of security requirements.

For security questions, we found that about a quarter of web applications implemented security questions. However, a large portion of security questions in use are guessable or researchable and about a quarter of security questions have at least one characteristic that hampers their usability. Of the security questions analyzed, the best questions have specific format, have a large and clearly defined answer space, ask about a specific event or person in the past, have answers that are not contained in public records, and have answer spaces without popular answers that represent a majority of the population. Most online web applications do not use the security mechanism of HTTPS nor CAPTCHA in conjunction with security questions to improve the security of security questions. We found no relationship between usage of security questions and storage of financial information.

Regarding fallback authentication, we found that many accounts rely on email as login credential and email for initiating and implementing password recovery, which emphasizes the need for security of email provider web applications. Our analysis indicates that few web applications follow best practices and that most web applications do not follow password recovery best practices or they implement insecure practices.

We found that less than a quarter of online web applications use multistep verification, about half of web applications allow single-sign on, and over a quarter of web application implement continuous authentication. However, we found no relationship between single-sign on and storage of financial information and no relationship between continuous authentication and storage of financial information. Overall, our results suggest that most online web applications do not follow best practices and do not effectively utilize security mechanisms nor use a combination of security practices to improve overall security.

Outline of Thesis

The rest of this thesis is organized as follows. Chapter 2 provides an overview of existing work for the topics that are addressed in this thesis such as password, security questions, and fallback authentication. In Chapter 3, the methodology for gathering the data for online web applications is described. Chapter 4 describes the results of the password practice data gathered for the online web applications, the comparison of password policy data from 2017 to the password policy data in 2010, and the trends in password practices for online web applications. Chapter 5 describes the classification of security questions and the examination of security question practices. Chapter 6 discusses and evaluates fallback authentication practices for online web applications. Chapter 7 discusses the results for other security mechanisms related to authentication such as multistep authentication, single-sign on, and continuous authentication. Chapter 8 discusses the results of examining various security considerations for online web applications such as use of HTTPS and use of CAPTCHA. The conclusion and future improvements to this work are provided in Chapter 9.

CHAPTER 2

RELATED WORK

Primary Authentication

There is a plethora of existing research on authentication mechanisms, especially on the usability and security of password schemes. One topic of concern is the burden placed on users when passwords are used for authentication. Adams and Sasse (1999) performed a study assessing how well participants remembered their passwords and discovered that users have more difficulty recalling their password as the password complexity increases. By examining the amount of time and number of attempts that participants needed in order create complex passwords and the password memorability, Vu et al. (2007) demonstrated that enforcing strict password requirements did not guarantee that users will select more secure passwords. For participants who had to remember five passwords, 25% of them were unable to recall at least one. This relates to the results from a survey by SafeNet (2005) indicating that 47% of respondents managed over 5 passwords and that 47% of respondents forgot at least one password per year. In terms of error tolerance, Brostoff and Sasse (2003) examined enforcing a threshold of three attempts for participants to correctly enter their password. Their results indicated that only 10% of users failed to login and that the failure rate could be reduced if the number of allowed attempts was increased to ten. The previous literature support some of the earliest recommendations by Zurko and Simon (1996). In their work, they recommended being aware of the security burden placed on users. In this context, security system designers should be aware of the memorability burdens that complex passwords place on users.

One of the common themes from existing work is that users make poor security decisions with regards to passwords. Herley and Florencio (2007) examined password use and reuse for half a million users over a three-month period. The main results of their work indicated that the average password is reused at six distinct accounts, a user on average has about 25 accounts, and that users choose weak passwords. On a smaller scale, Gaw and Felton (2006) surveyed password usage by undergraduates. Their findings were that undergraduates had at most three passwords on average, reused passwords at least twice, and that they were more likely to reuse passwords as they accumulated more accounts. Further promoting the idea that users make poor security decisions, Beautement et al. (2008) showed that users only have a limited amount attention that they are willing to allocate for security, which is based on the actual and perceived benefits and costs for adherence. Furthermore, Herley's (2009) examination of security advice supports the idea that users perform a cost-benefit analysis when they consider following security advice, even when it involves ignoring advice for strong passwords. On a larger scale, Bonneau and Preibush (2010) performed an empirical analysis of password implementations for various accounts. They found that there were poor security practices with accounts with low security incentives yet accounts that stored sensitive information tended to have more password security. Likewise, Inglesant and Sasse (2010) examined password policies and practices in the workplace. Instead of focusing on the most secure policies, they advocated for security policies that are based on a human-computer interaction framework because users do consider security but many security protocols are incompatible with the user's needs.

In terms of the future of passwords, St. Clair et al. (2006) conducted an empirical study of password practices in order to determine the effectiveness of offline attacks. Their results suggest that many systems are vulnerable to offline attack and need to be made more secure against such attacks. On a similar note, the current usage of passwords and the reasons why password practices are still in place was performed by Herley et al. (2009). They cited diversity of requirements, competing proposals, scarcity of loss data, user reluctance and usability, no standardization, and individual control of end-user platforms as some of the reasons why passwords practices are still in use. For future improvement on password systems, Sasse et al. (2001) explored ways to improve user memorability and increase user adherence to security practices. Furthermore, Bellovin (2008) recommends performing analysis on the security practices in order to decide which security practices should be implemented.

In an attempt to address the security problems associated with textual passwords, several other primary authentication schemes have been proposed. Dhamija and Perrig (2000) proposed an image based authentication scheme that works by authenticating users based on their ability to recognize previously observed images. Their proposed solution achieved a 90%

authentication rate compared to the 70% authentication rate for passwords and PINs, which indicates that their proposed method is reliable and memorable relative to passwords. Using a similar scheme, Jermyn et al. (1999) formulated a graphical password mechanism that utilizes spatial position and temporal order in order to increase the graphical password answer space. Rather than using one static password for primary authentication, Herley and Florencio (2006) as well as Haller (1994) performed research on one-time password authentication systems used as potentially viable authentication schemes.

The security of passwords can be improved by combining them with another type of authentication mechanism, which is called multifactor authentication. Multifactor authentication is a scheme where two different types of authentication information used in order to authenticate a user. The types considered by two-factor authentication are proof of knowledge of a secret, proof of possession of a physical token, and proof of possession of physical characteristics (biometrics), There are several web applications that already offer two-factor authentication, which spawned the creation of the two-factor.org project that documents a partial list of web accounts that offer two-factor authentication (Davis, 2017). One common method of multifactor authentication is the combination of passwords with one-time passwords sent to mobile phones. Tiwari et al. (2011) proposed a multifactor mechanism using a transaction ID code and SMS. A major issue about using phones as an implementation of multifactor authentication is the fact that phone numbers change and phones are often lost or are not operable everywhere. According to Consumer Reports, about 2.1 million American had their phones stolen in 2014 compared to 3.1 million Americans having their phones stolen in 2013 (“Smartphone thefts drop as kill switch usage grows But Android users are still waiting for the technology”, 2015).

The most relevant related work to this thesis was the work by Herley and Florencio (2010) concerning the password policies for online web applications. They examined the password policies for 75 online accounts for categories including the top traffic web accounts, banks, and top universities. Their results indicate that presence of brute force attacks, higher user traffic, and extractable value of the accounts do not correlate with stronger password policies;

however, competition for user traffic and making a profit from advertisements correlate with weaker password policies.

Security Questions

Overall, many of the of the works that examine security questions focus on a specific area such as banking or user-written questions. One of the earliest examinations of security questions was performed by Haga and Zviran (1991). They analyzed the memorability and guessability of cognitive (i.e., based on user opinions, interests, and biography) and associative (i.e., based on word or concept association) security questions and found that family members could correctly answer 60% of fact-based security questions, spouses could answer 41% of fact-based security questions, and friends could answer 23.5% of fact-based security questions. In order to assess the security questions used in practice, Rabkin (2008) surveyed the security questions used by various bank accounts. Their major findings include that banks do not provide guidance for security questions and many of the questions are not considered strong due to being guessable or researchable. Similarly, the memorability and security of security questions employed by major email providers were analyzed by Schechter et al. (2009). According to their results, 75% of participants could recall their answer and 76% of participants who could recall their answer did so within five attempts, however, the security questions proposed by email providers and user written questions were weak due to guessability.

Numerous researchers including Podd et al. (1996) and Zviran and Haga (1991) performed research on the guessability of security questions by significant others. Podd et al. (1996) also examined the recall rate for various security questions and showed that 77% of participants could recall their security question answers. Another security aspect for security questions addressed by Griffith and Jakobsson (2008) was whether the security questions could be answered from information available publicly. One of their main findings was that security questions such as “What is your mother’s maiden name?” can be derived by an attacker with reasonable probability. One of the main issues with security questions is answering the question in the same format and spelling as established during security question registration. Both Ellison et al. (2000) and Frykholm and Juels (2001) explored tolerating errors for incorrectly answered security questions. Their methods involve tolerating errors by requiring a threshold of questions to be correctly answered before a user was authenticated.

Toomin et al. (2008) also examined user written questions but focused on social access control for social groups enforced by answering a question based on shared information. Based on their user survey, they showed that the participants chose questions that could be authenticated by members of their social groups yet were only correctly guessed by at attacker 11% of the time given ten guesses. In an effort to analyze questions chosen by users, Just and Aspinall (2009), measured usability and guessability of user-written questions. Their work showed that user-chosen questions had individually low entropy, but a majority of users chose a combined set of three questions that were less susceptible to guessability and observability. Hence, they suggested combining security questions with other techniques and requiring a subset of questions to be answered in order to improve their security.

An alternative scheme for security questions proposed by Jakobsson et al. (2008) involves questions based on user's preferences on a specific topic. In order for this scheme to have tolerable error rates, a large set of security questions must be set up and a fairly large subset of 16 questions must be answered at authentication time. O'Gorman et al. (2004) developed preference based security questions using a protocol called Query Directed Passwords (QDP) and found that the scheme provided reasonable security. Rather than having security questions that are likely researchable using public information, QDP asks numerous questions about trivial facts and opinions. In a slightly different vein, instead of registering security questions, Asgharpour et al. (2007) experimented with a scheme where users answer questions about their browser history.

The most thorough classification of security questions was performed by Just (2004) where he defined the security and usability categories for security questions. His security question metrics have been the basis for the categories used by other researches such as Rabkin (2008). Just (2004) divided security question attributes into three classes: usability, security, and privacy. In terms of usability, he defined the criteria as applicability, memorability, and reliability.

He separated security criteria into two parts: observability and guessability. Just (2004) defined the privacy criteria as collection limitation and use limitation. In addition to his security categories, Just provided recommendations for improving security questions. Beyond research papers, the Canadian government posted recommended practices for security questions (“Office of the Privacy Commissioner of Canada”, 2006).

Fallback Authentication

In terms of fallback authentication, there is research on the different types of methods of fallback authentication. Reeder and Schechter (2011) defined and evaluated the numerous fallback authentication methods that can be used in practice and then recommended best practices for each mechanism. Focusing on fallback authentication, Just (2004) also discusses the types of authentication that can be used for fallback authentication with a focus on security questions. Similarly, Kluecker (2013) provides an overview of fallback authentication mechanisms and then analyzes each approach in terms of usability and security. As an alternative method for fallback authentication, the social authentication method of using trustees was proposed by Schechter et al. (2009, April). This authentication method involves a user registering several other users as “trusted users for password recovery” or trustees. A threshold number of three trustees must successfully respond with an emailed code after being contacted by the user in order for the user to recover their password. The scheme had promising results because 89% of participants were able to contact their trustees; however, 72% of participants had to be reminded of their trustees in order to complete the process.

CHAPTER 3

METHODOLOGY

We created accounts and gathered data for web applications from different categories between June 2016 to April 2017. The data was further validated between January and April 2017. In order to properly evaluate the security practices for each web application, we created and registered 230 web accounts out of 282 web applications observed. In some cases, we were not able to create accounts for applications such as banks, insurance companies, and universities. However, we used publicly available information to record data on the password policies, multistep authentication options, and password recovery methods for accounts where we could not register for an account. In order to collect data for the web applications with high user traffic, we referred to the top traffic websites for different categories using Alexa (www.alexa.com). Alexa is a web application that provides statistics on commercial web traffic data. The results provided by Alexa are considered in the research community to be reasonable for representing the top traffic websites for various categories. We collected account data for the following categories defined by Alexa:

- top earning websites,
- top art websites,
- top business websites,
- top computer websites,
- top gaming websites,
- top health websites,
- top home websites,
- top kids and teens sites,
- top news websites,
- top recreation websites,
- top reference websites,
- top regional websites, and

- top shopping websites.

We note that Alexa did not provide a description or definition for these categories, While some categories are self-evident such as computer websites and gaming websites, but some categories are vague such as home websites. In addition, any website listed by Alexa could fall into multiple categories so each Alexa category does not have a unique set of websites. For each web application in our data set, we collected the following data:

- information required for user registration,
- password policies,
- types of multistep authentication mechanisms offered,
- types of continuous authentication available,
- types of single-sign on available
- the process for password recovery,
- the security features explicitly offered by the account,
- the login and password recovery credentials,
- the types of financial information that can be stored on the account.

User Registration Data Collection

When creating an account, we documented the account name and its URL. Each input field on the registration page was recorded. The most common fields documented include name, address, phone, email, username, social security number, CAPTCHA, terms and conditions, and privacy policy. If an account allowed certain credentials to be verified, we would verify the credential and record whether the verification was mandatory for registration.

Password Policy Data Collection

We recorded the password policy for as many of the web applications as possible. If we had the ability to register for an account, we created the account and recorded the posted password policy in the registration page. Next, we attempted to submit the weakest password based on the posted password policy in order to determine whether the password policy was properly enforced. For example, if the password policy only specifies a minimum password length and the minimum length is 8, then the password 11111111 was submitted in order to see if it would be accepted. In order to confirm that the password policy was consistent throughout the web application, we recorded the password policy for the change password page and the password reset page. The web applications with inconsistent password policies were documented. If we were unable to register for an account, we used officially and publicly posted password policies. One drawback of relying only on the posted password policies was that it prevented us from determining if the password policies were enforced.

Alternative Authentication Mechanisms Data Collection

We collected data regarding whether the web applications implemented multistep authentication and the types of multistep authentication offered for each web application. Unlike multifactor authentication, multistep authentication allows two authentication credentials from the same category; therefore, multifactor authentication is a type of multistep authentication. For accounts that we could create, we searched the account settings and official account help documentation to determine whether multistep authentication can be enabled. We recorded information about whether any type of multistep authentication was available and the types of multistep authentication that were available such as using a mobile phone one-time password or using a hardware token. In order to test the multistep authentication options, we configured multistep authentication where we had the resources to configure the option (e.g., phone, email, and backup codes but not hardware tokens). For the accounts that we could not create, we documented the publicly posted information on their multistep and multifactor authentication policy.

Additionally, we collected data about each web application's usage of single-sign on. We recorded whether an account utilized another system's login mechanism such as Facebook or Google for their own authentication system. We documented all the possible single-sign on providers for each web application.

Security Question Data Collection

We recorded whether web applications implemented security questions and what security questions were used by each web application. We recorded whether web applications prompted users to setup security questions. The main fields recorded for security questions were the minimum number of required security questions, the maximum number of security questions, the types of security questions required, whether the answers were visible on the account page, whether the answers are visible when answering security questions, and the type of security mechanisms that used security questions. If a web application made the set of possible security questions available publicly, we documented the pool of security questions. For each security question, the type of security question and the type of answer format were recorded. In addition, we examined whether security questions could be edited and if the same security questions were used for different security mechanisms. If we could not create an account and had no access to the content of the questions asked, we used publicly posted information in order to determine whether a web application implemented security questions in some fashion. For example, a web application's help page may mention a step that involves security questions for fallback authentication or may recommend updating their security question periodically.

Fallback Authentication Data Collection

We were able to create accounts for 230 web applications. For each of those accounts, we documented the password recovery process. First, we recorded the types of password recovery options that were available. Next, we initiated password recovery for online recovery options and noted the credentials that were necessary to initiate password recovery. The types of password recovery include Help Desk Recovery, In Person Recovery, Text Temporary Password

Recovery, Personal Info Password Recovery, Call Mobile OTP Recovery, Phone OTP Recovery, Email Link Recovery, Email New Password Recovery, and Email Current Password Recovery. We documented the steps necessary to complete password recovery and data relevant for evaluating the web application's adherence to best practices such as logging out active sessions after successful recover and whether a web application locked a user out of their account after password recovery was initiated. If email link recovery was used, we recorded the validity period for the link if available. In addition, whether CAPTCHAs were utilized in password recovery and whether one time passwords were used in conjunction to other methods like emailing a password recovery link were documented.

Security Feature Data Collection

In order to assess the security of the web applications in terms of their security features, we documented the implemented security mechanisms that each web application employed. One security feature that we investigated was the presence of HTTPS. We recorded the availability of HTTPS prior to login, during registration, during login, and after login for each web application. Prior to registration, we visited the home page and a few linked pages on the web application. Using the Google Chrome browser's URL bar, we recorded whether the web application used HTTPS, or an EV Green Bar HTTPS prior to being logged into the application. For the account registration page and login page, the type of HTTPS used was documented, if any. After login, we recorded the type of HTTPS present in various pages including the account, profile, and settings pages. However, one limitation was that we did not examine HTTPS network traffic to determine all requests used HTTPS until the final redirect occurred. Thus, a web application may seem to use HTTPS but not all redirects have HTTPS enabled, yet we assume that this case is unlikely.

After the account creation, we documented the presence of other security features that could be enabled on the account. The following are the security feature categories documented: One-Time Password (OTP) to Access Account, Login Again to Access Account, Challenge Response for Anomalous Behavior, Ability to Add Public Key, Trustee System for Fallback Authentication, Permission Management, Authorized Users, Trusted Devices, Backup Codes, Manage App Access, Manage Third Party Application Access, White List of IP Addresses, Session Management, Multiple Tabs Allowed for Same Session, CAPTCHA During Login, Email or Phone OTP for Unrecognized Browser, Security Question for Account Changes, Max Consecutive Reset Attempts, and Max Consecutive Login Attempts.

Another feature for which we gathered data was the ability to delete the account from the account's user interface. After an account was created, we searched the user interface for a method to delete the account from the account interface. Since the delete option was not always easily found on the interface, we referred to help documentation for the process of deleting an account. It is important to note that we did not record whether an account could be deleted by contacting customer support if it was not explicitly described as a valid option by the account interface or the help documentation. We did not attempt to complete the delete account process as we wanted to keep access to account in case data verification was needed.

Financial Information Data Collection

One of the metrics for assessing the value of an account was by determining whether an account stored forms of financial information such as credit cards, bank information, or links to PayPal accounts. For each account, we documented whether credit cards, debit cards, bank accounts, PayPal account links, and Bitcoins could be stored on the account. In order to assess whether attackers could redirect physical packages, we documented whether shipping, mailing, and billing addresses could be stored on the account. Next, we attempted to change the addresses and recorded whether we could do so successfully in order to determine whether an attacker could do so if they compromised the account.

Identity Credential Modification Collection

In order to assess the usable security of the account, we gathered data concerning the ability to change the identity credentials. By having the ability to change the identity credentials for login, if an attacker manages to gain access to the account, the user could change the login credential in order to deny the attacker access, but the attacker could also change the login credential in order to deny the user access to the account. In addition, the ability to change the identity credential for password recovery grants the user the ability to regain access to their account if the original recovery credential is inaccessible. During account registration, we recorded the login identity credential for the account. In addition, we documented the password recovery identity credential needed to initiation password recovery such as username. Next, the ability to change the login identity credential and password recovery credential were documented. If the identity credential involved an external communication medium such as email or phone, we recorded whether the credential could be verified and whether it was required to be verified before it could be officially modified.

CHAPTER 4

PASSWORD PRACTICES

We measured the strength of the password based on the password requirements for 274 out of the 282 total accounts examined. Password policy data could not be gathered for seven of the accounts due to reasons such as the account had no publicly posted password policy or the system only allowed system created passwords. Out of 230 accounts, only one did not use passwords because it allowed many different university login mechanisms to gain access to an account. The strength of a password is dependent on the complexity of the password. Complexity is often measured by resistance to brute force attacks, which implies that passwords should be long and have a large character set to increase the computation power and time needed for a brute force attack to be successful. Therefore, a 12-character password with lowercase, uppercase, and digits has more complexity and thus more strength than a password with four digits.

For the password strength, we calculate the password entropy assuming that all passwords of minimum length are equally likely and only passwords of minimum length are chosen. The entropy is given by $l*\log_2(c)$, where l is the minimum password length and c is character set cardinality. As Herley and Florencio (2010) note, this measurement only approximates the resistance of the password to a brute force attack. They reinforce this concept by explaining that a 34-bit password like “aIRKzO” could be more resistant to brute force attacks than 66-bit password “Let_Me_In!” since the later uses common English words. However, absent other information about how users choose passwords, this measurement gives an approximation of password strength for password chosen by a careful user or by a password manager. Also, this measurement allows us to compare the data from 2017 to their data from 2010.

As described in Chapter 2, we collected account information for web applications in categories defined by Alexa, but Alexa did not provide a meaningful description for each category that they used nor do they justify why or how they categorize a website into a specific category. In order to provide more meaningful categories for web applications, we group the web applications into the following categories:

- Airlines,
- Banks and Brokerages,
- College,
- Computer,
- Educational,
- Email Providers,
- Entertainment,
- Gaming,
- Government,
- Health,
- Hotels and Motels,
- Insurance,
- News and Journalism,
- Online Shopping,
- Real Estate,
- Research,
- Restaurants,
- Social Media,
- Software Service Provider, and
- Travel.

Using the overall data set and the groupings, we analyze the median password policy strength and average password strength for each grouping.

Password Policies

Table 1

Password Strength Statistics per Category for 274 Accounts

Category	Median Password Strength	Average Password Strength	Number of Accounts	Percent
Government	49.03426414	55.17180148	15	5.5%
College	47.63357048	43.76181419	29	10.6%
Email Provider	41.35940001	35.9593968	11	4.0%
Banks and Brokerage	31.01955001	35.65218826	12	4.4%
Airline	28.20263831	35.26099865	11	4.0%
Gaming	33.21928095	34.32685587	19	6.9%
Hotels and Motels	37.60351775	33.29323434	7	2.6%
Software Service Provider	28.20263831	32.51399701	14	5.1%
Computer	26.57542476	32.15947037	9	3.3%
Health	30.64548429	30.64548429	2	0.7%
Restaurant	26.57542476	30.29014859	20	7.3%
Online Shopping	26.57542476	28.15667288	43	15.7%
News and Journalism	19.93156857	24.39874673	19	6.9%
Travel	26.57542476	23.43429817	9	3.3%
Insurance	23.25349666	23.25349666	2	0.7%
Education	19.93156857	21.48565524	5	1.8%
Social Media	19.93156857	20.44263443	13	4.7%
Research	19.93156857	19.93156857	1	0.4%
Entertainment	19.93156857	19.68252362	28	10.2%
Real Estate	4.700439718	12.4590829	5	1.8%

Table 2

Password Bit Strength Statistics per Category for 229 Accounts

Category	Median Password	Average Password	Count	Percent
Government	48.86819565	51.82850981	9	3.9%
College	48.52871352	48.52871352	1	0.4%
Banks and Brokerages	37.10449762	36.70506595	4	1.7%
Email Providers	43.48145888	36.13507265	10	4.3%
Airlines	28.20263831	35.26099865	11	4.8%
Gaming	33.21928095	33.70009837	19	8.3%
Hotels and Motels	37.60351775	33.29323434	7	3.0%
Software Service Provider	28.20263831	32.51399701	14	6.1%
Computer	26.57542476	32.15947037	9	3.9%
Health	30.64548429	30.64548429	2	0.9%
Restaurants	26.57542476	29.72413359	20	8.7%
Online Shopping	26.57542476	28.2733202	43	18.7%
Travel	26.57542476	23.43429817	9	3.9%
News and Journalism	19.93156857	22.40157396	19	8.3%
Educational	19.93156857	21.48565524	5	2.2%
Social Media	19.93156857	20.44263443	13	5.7%
Entertainment	19.93156857	19.68252362	28	12.2%
Real Estate	4.700439718	12.4590829	5	2.2%
Research	9.965784285	9.965784285	2	0.9%

Tables 1 and 2 show the password policy statistics for the 229 and 274 online accounts, respectively. Both tables indicate that the Government and College categories had the strongest password policies while real estate had the weakest password policies. In both tables, the Email Provider category had a relatively strong average password strength.

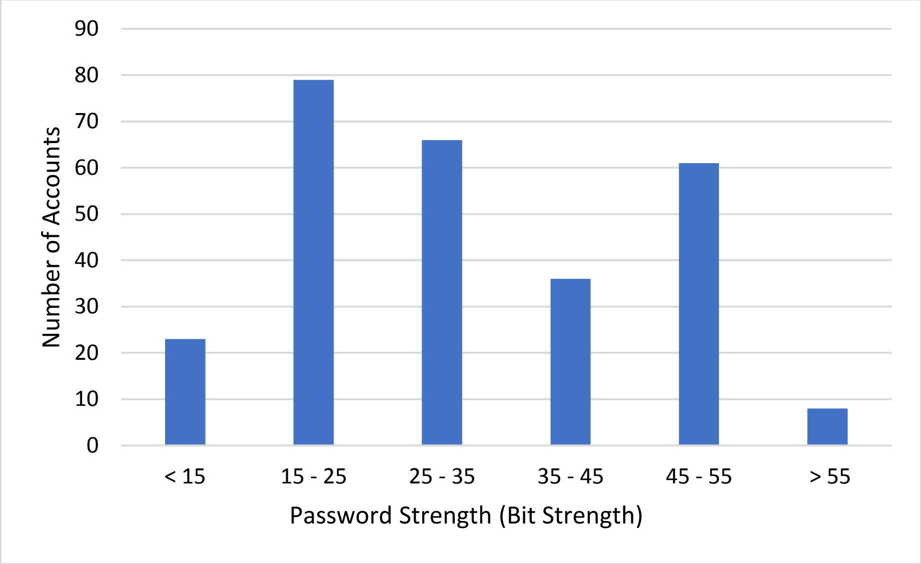

Figure 1. Password strength histogram for 274 web applications.

Figure 1 shows a histogram of the bit strength of the password policies for 274 online accounts. About 60% of the online accounts have a password policy bit strength that is fewer than 35 bits. About 13% of the accounts have a bit strength between 35 and 45 bits. Approximately 25% of online accounts have a password policy strength of 45 bits or greater.

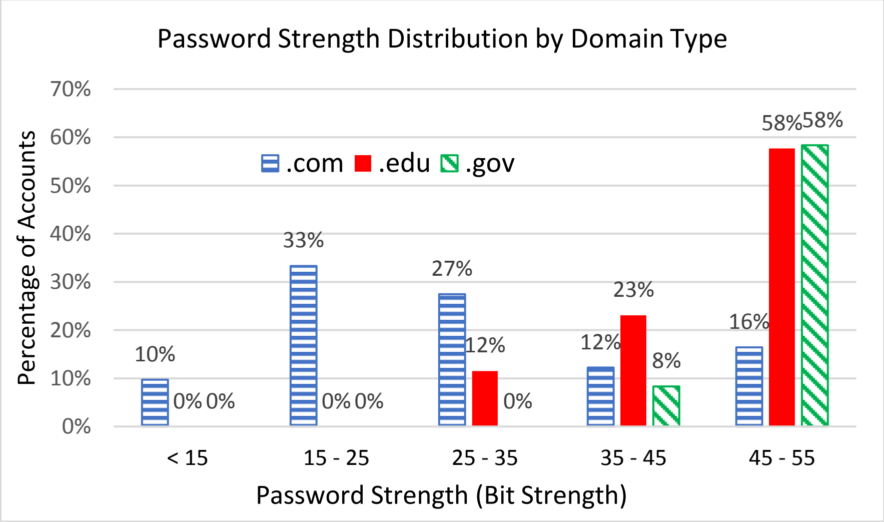

Figure 2. Password strength histogram for 274 web applications by domain type.

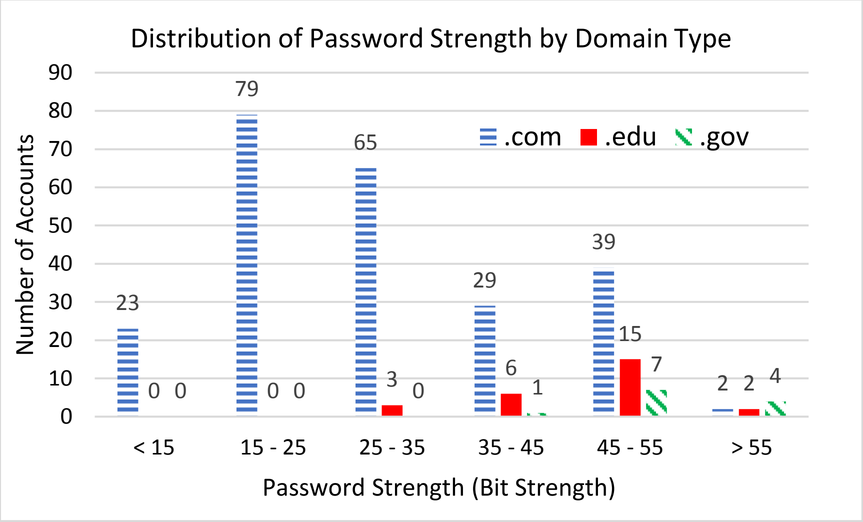

Figure 3. Password strength histogram for 274 web applications by domain type.

In Figures 2 and 3, the distribution of password strengths for each domain type is displayed. For .com accounts, 43% had a password policy strength of fewer than 25 bits, 29% had passwords strengths between 25 bits but fewer than 45 bits, and 28% had password strengths of 45 bits or greater. For .edu accounts, 0% had a password policy strength of fewer than 25 bits, 35% had passwords strengths between 25 and 45 bits, and 65% had password strengths of 45 bits or greater. For .gov accounts, 0% had a password policy strength of fewer than 25 bits, 8% had passwords strengths between 25 bits but fewer than 45 bits, and 92% had password strengths of 45 bits or greater.

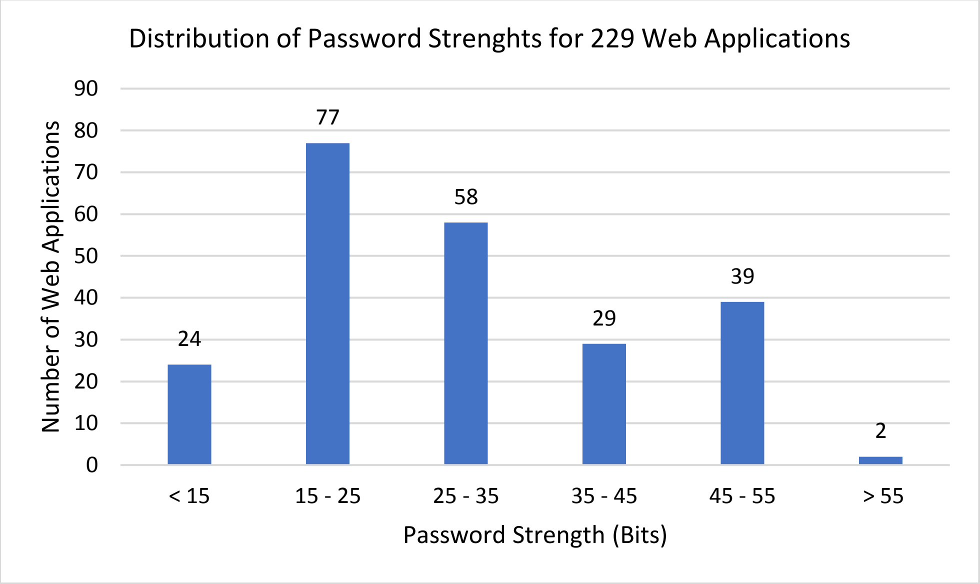

Figure 4. Password Strength Histogram for 229 web applications.

Figure 4 shows a histogram of the bit strength of the password policies for 229 online accounts. About 70% of the online accounts have a bit strength that is less than 35 bits. About 13% of the accounts have a bit strength between 35 and 45 bits. Approximately 17% of online accounts have a password policy strength of 45 bits or greater.

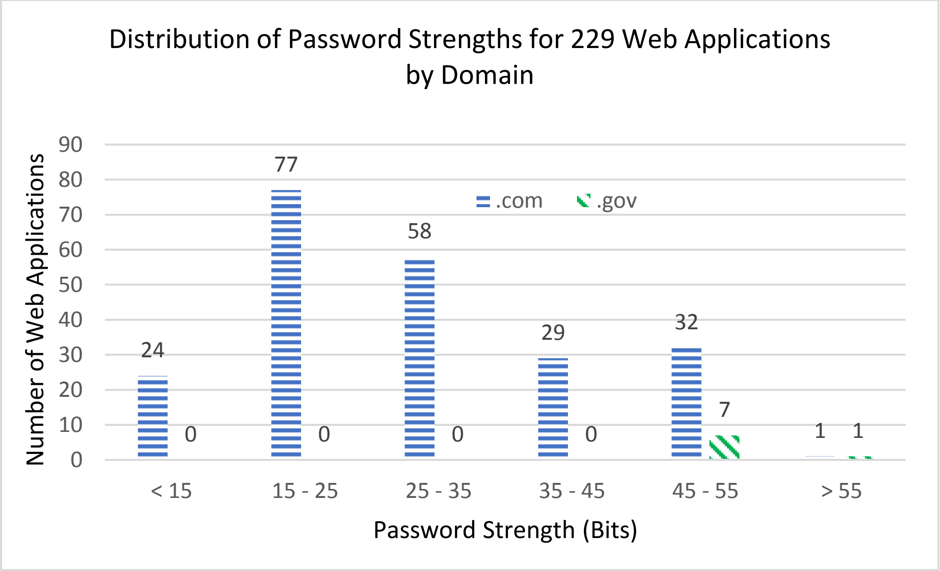

Figure 5. Password strength histogram for 229 web applications by domain type.

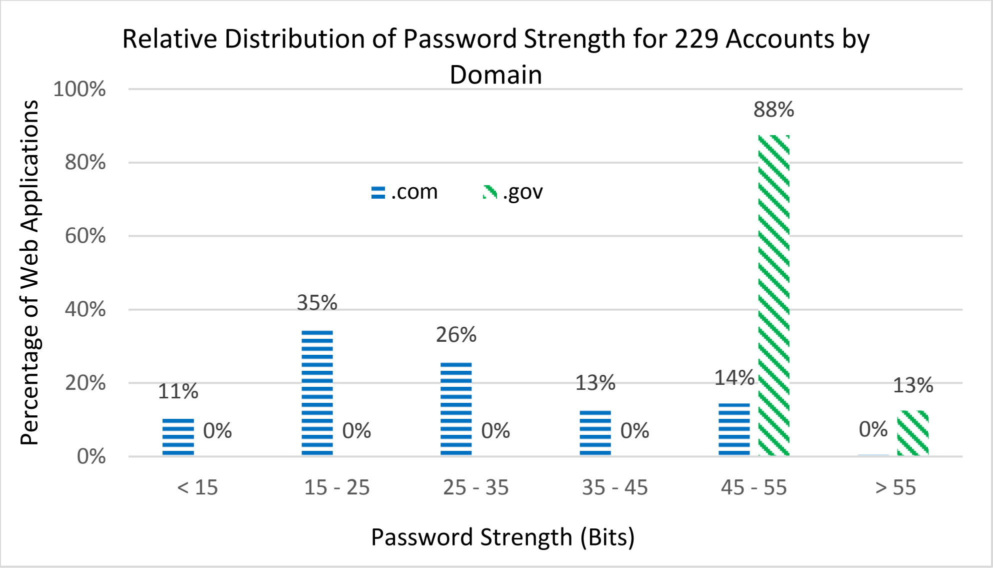

Figure 6. Password strength relative histogram for 229 web applications by domain type.

In Figures 5 and 6, the distribution of password strengths by domain type is shown. For .com accounts, 46% had a password policy strength of fewer than 25 bits, 39% had passwords strengths between 25 bits but fewer than 45 bits, and 14% had password strengths of 45 bits or greater. For .edu accounts, we could not create more than one edu account so it was not meaningful to represent it in figures n and n. For .gov accounts, 0% had a password strength fewer than 25 bits, 0% had passwords strengths between 25 bits but fewer than 45 bits, and 100% had password strengths of 45 bits or greater.

During our data collection for the 230 created accounts, we came across some web applications that had inconsistent password policies. This means that the web applications posted conflicting requirements related to minimum password length, required characters, and allowed characters on the registration page, change password page, or reset password page. The following are the web applications with inconsistent password policies:

- Amazon,
- Blizzard,
- Expedia,

- Hotels.com,
- IMDB,
- iPhoto,
- King,
- Lenovo,
- Netflix,
- Open Table,
- Orbitz,
- Roblox,
- Sonic Merchandise,
- Thoth Lab,
- Travelocity,
- United Airlines,

Comparison to Existing Data

One of the main inspirations for our research was the examination of password practices by Herley and Florencio (2010). They recorded the password policies and calculated the password strength for 75 different online web applications for categories such as top traffic sites, high traffic site, medium traffic sites, banks and brokerages, large universities, top university computer science departments, and government. From this data set, they tested hypotheses that correlated password policy to various factors. The factors that they examined were presence of brute force attacks, the amount of traffic or number of registered users for the site, whether the username is public, the value of the resources protected, the extractable value of the resources protected, who faces the consequences of the breach, acceptance of advertisements, evidence of advertising, and user choice of the service. Based on the relevance of each hypothesis to our work, we tested a subset of their hypotheses.

In order to compare the changes in password policies for those accounts from 2010 to 2017, we gathered password policy data for 70 of those 75 web applications. We could not gather data for five of the web applications because either those accounts no longer exist or no longer post their password policy publicly. We needed information related to user traffic ranking, number of registered users, asset value, whether an account accepts ads, top phished brands, and user choice for the accounts that were analyzed by Herley and Florencio. We were able to use web applications like QuantCast (https://www.quantcast.com) to collect information about traffic ranking, but we had to rely on other sources for information related to ad acceptance, top phished brands, number of registered users, and assets values. For data related to ad acceptance, we manually examined each site using an ad blocking software called AdBlock and attempted to determine whether an application accepted ads or not.

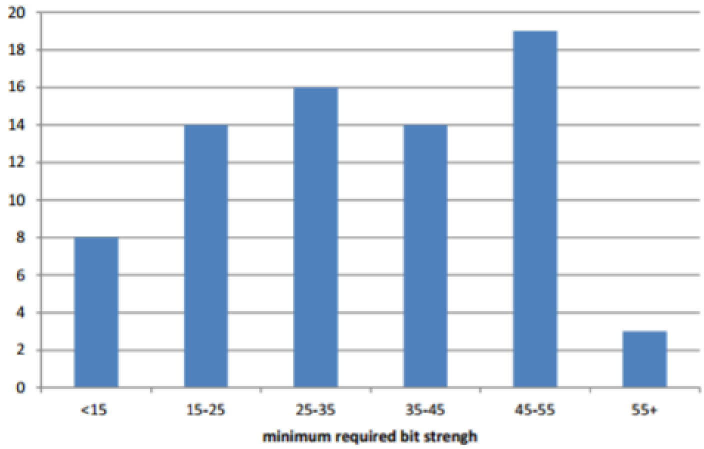

Figure 7. Herley and Florencio password strength histogram for 75 online accounts in 2010.

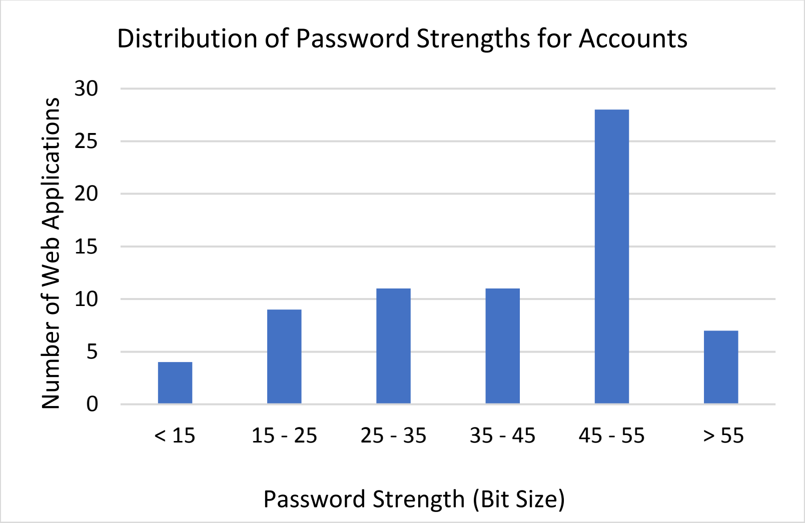

Figure 8. Password strength histogram for 70 online accounts in 2017.

Figures 7 and 8 show the distributions of password strengths from 2010 and 2017 respectively. For Herley and Florencio's data, 8% of accounts had a bit strength fewer than 15 bits, 14% of accounts had a bit strength between 15 and 25 bits, 16% of accounts had a bit strength between 25 and 35 bits, 14% of accounts had a bit strength between 35 and 45 bits, 19% of accounts had a bit strength between 45 and 55 bits, and 3% had password strength of greater than 55 bits. These results contrast with the data gathered from 2017, where 29% of accounts had a bit strength fewer than 15 bits, 0% of accounts had a bit strength between 15 and 25 bits, 1% of accounts had a bit strength between 25 and 35 bits, 13% of accounts had a bit strength between 35 and 45 bits, 7% of accounts had a bit strength between 45 and 55 bits, and 44% had password strength of greater than 55 bits.

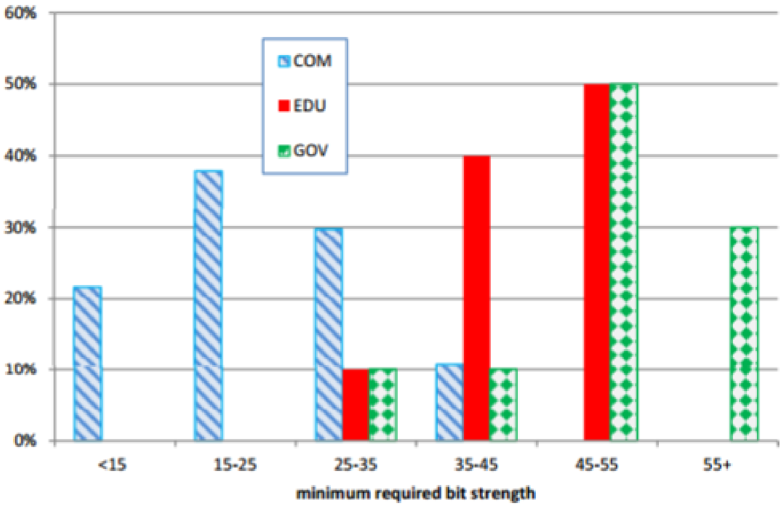

Figure 9. Herley and Florencio password strength histogram by domain type.

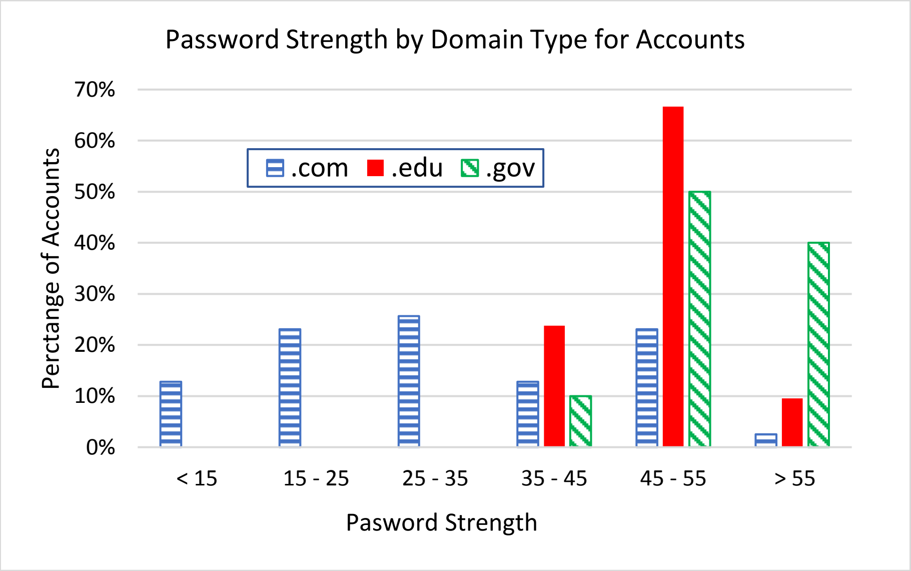

Figure 10. Password strength histogram by domain type in 2017.

Figures 9 and 10 show the distribution of password strength by domain type from 2010 and 2017, respectively. Herley and Florencio's data indicated that for .com accounts over 60% of accounts had a bit strength fewer than 25 bits, 40% of accounts had a bit strength between 25 bits and 45 bits, and 0% of accounts had a bit strength of over 45 bits. For .edu accounts, 0% of accounts had a bit strength fewer than 25 bits, 50% of accounts had a bit strength between 25 bits and 45 bits, and 50% of accounts had a bit strength of over 45 bits. For .gov accounts, 0% of accounts had a bit strength fewer than 25 bits, 20% of accounts had a bit strength between 25 bits and 45 bits, and 80% of accounts had a bit strength of over 45 bits. Our data had different trends. For .com accounts over 26% of accounts had a bit strength fewer than 25 bits, 48% of accounts had a bit strength between 25 bits and 45 bits, and 26% of accounts had a bit strength of over 45 bits. For .edu accounts, 0% of accounts had a bit strength fewer than 25 bits, 24% of accounts had a bit strength between 25 bits and 45 bits, and 76% of accounts had a bit strength of over 45 bits. For .gov accounts, 0% of accounts had a bit strength fewer than 25 bits, 10% of accounts had a bit strength between 25 bits and 45 bits, and 90% of accounts had a bit strength of over 45 bits.

Table 3

Median Password Strengths Comparison between 2010 and 2017

Account Group	**Median Password Strength 2010**	**Median Password Strength 2017**	**Herley and Florencio Number of Accounts**	**Our Number of Accounts**
Top Traffic	19.9	26.6	15	14
High Traffic	19.9	35.2	8	8
Medium Traffic	8.3	28.1	8	6
Banks and Brokerages	31.0	31.0	9	8
Large Univ.	44.5	47.6	10	10
Top CS Dept.	46.4	50.1	10	10
Government	47.6	51.3	10	10
.com	19.9	26.6	41	37
.edu	43.7	47.6	22	22
.gov	47.6	49.1	12	12

As Table 3 shows, we compared the median password strengths from 2010 to the password strengths in 2017 for the groupings defined by Herley and Florencio (2010). The column “Our Number of Accounts” shows that we were missing a total of four accounts from Herley and Florencio’s data. We included only one account for MIT but the 2010 data has two accounts for MIT and we were not able to get information on how the two accounts differ. There was one account missing for the Top Traffic and Banks and Brokerages groupings and only two accounts missing from the Medium Traffic grouping.

Table 4

Registered Users for 5 Top User Sites and 5 Largest Universities in 2010

Web Application	Users	Rank	Min Strength 2010
Facebook	400 million	2	19.9
Yahoo	260 million	3	19.9
Live	260 million	8	19.9
Gmail	91 million	1	19.9
Twitter	76 million	31	19.9
Ohio State	51,800	1811	41.4
ASU	51,200	3288	47.6
University of Florida	50,900	1382	47.6
University of Minnesota	50,400	919	35.7
University of Texas	49,000	946	47.6

Table 5

Registered User for 5 Top User Sites and 5 Largest Universities in 2017

Web Application	Users	Rank	Min Strength 2017
Gmail	3+ billion	1	53.6
Facebook	2+ billion	4	26.6
Yahoo	1+ billion	6	46.0
Live	400+ million	5	26.6
Amazon	304+ million	8	19.9
ASU	98,146	7086	49.5
Ohio State	66,046	7247	43.4
Texas A&M	66,426	2372	47.6
University of Central Florida	64,318	NA	32.6
University of Florida	52,286	2800	52.6

Tables 4 and 5 compare the changes in number of users from 2010 to 2017. As seen in Table 4, Gmail has outpaced Facebook in terms of registered users and Amazon has replaced Twitter as the fifth largest number of registered users. Herley and Florencio (2010) used

QuantCast (https://www.quantcast.com) to get the number of registered users and the traffic rank for their data in 2010. QuantCast.com had no ranking for the University of Central Florida in 2017 so we provided the value NA for not available in order to keep our ranking data consistent with that of Herley and Florencio.

Table 6

Comparison of Account Assets for Financial Institutions between 2010 and 2016

Account	Assets 2010	Assets 2016	Min Strength 2010	Min Strength 2017
Bank of America	$2,200,000,000,000	$2,198,884,000,000	41.4	41.4
Chase	$2,000,000,000,000	$2,521,029,000,000	36.2	31.0
Citibank	$1,800,000,000,000	$1,818,117,000,000	31.0	31.0
Fidelity	$1,400,000,000,000	$2,100,000,000,000	19.9	47.6
Wells Fargo	$1,200,000,000,000	$1,942,124,000,000	31.0	19.9
Vanguard	$1,000,000,000,000	$1,781,000,000,000	26.6	26.6
PayPal*	$290,000,000,000	$354,010,000,000	26.6	26.6

In Herley and Florencio’s study of password practices (2010), they tested the hypothesis that accounts with higher assets have more strict password policies. Therefore, they collected data on account assets for financial accounts from FFIEC (2016, September 30). We also collected the FFIEC data for account assets in 2016 because this portion of our data was collected in February 2017. Like Herley and Florencio, we list annual transaction volume for assets for PayPal since they do not manage assets (“PayPal's annual payment volume from 2012 to 2016 (in billion U.S. dollars”, 2017).

Table 7

Comparison of Password Strengths for Top Phished Brands in 2010

Account	Number of Phishing Attacks	Min Strength 2010
PayPal	32205	27.0
Chase	25901	36.2
eBay	18738	31.0
Bank of America	4540	41.0
IRS	3712	47.0
Citibank	2265	31.0
Facebook	2217	20.0
Gmail	761	27.0
Yahoo	761	20.0
Wells Fargo	541	31.0

Table 8

Comparison of Password Strengths for Top Phished Brands in 2017

Web Application	Min Strength 2017
Facebook	26.6
Wells Fargo	19.9
Bank of America	41.4
LinkedIn	19.9
Booking	26.6
IRS	48.9
Amazon	19.9
Steam	52.7
Apple	47.6
Alibaba	19.9

Since Herley and Florencio tested a hypothesis related to the password policies of top phished accounts, they gathered data on the most phished brands from Avira (“The Most Phished Brands of 2009”, 2009). However, Avira has not updated their data in recent years; therefore, we used reports from Stastica (“Online brands most affected by phishing attacks as of 1st quarter 2016, by share of attacks”, 2017) and PhishMe (“Most Phished Brands ‘Missed’ by AntiVirus Based on Big Data Security Intelligence – Q3 2013”, 2013) to determine the top phished brands but we could not get data on number of phishing attacks. We provide data on the password strengths for top phished accounts in 2017 without the exact number of phishing attacks in Table 6.

Hypothesis Testing

The following are the hypotheses that we tested based on Herley and Florencio's hypotheses.

- A majority of web applications that have higher traffic or number of users enforce stricter password policies.
- A majority of web applications that have that use public username have stronger password policies.
- A majority of web applications that have high value assets enforce stronger password policies.
- A majority of web applications that have higher extractable value enforce stronger password policies.
- A majority of web applications that pay to attract traffic correlates less stringent password policies.
- A majority of web applications that compete for user traffic correlates less stringent password policies.

We did not test the hypothesis by Herley and Florencio (2010) that password policy strengths increased based on evidence of brute force attacks. Their argument does not involve data gathered from any online web applications. In addition, Herley and Florencio's (2010) reasoning that password policies cannot change easily, that only when policies are too weak is there any indication of an attack, that best practices inhibit gathering evidence, and that web applications cannot differentiate between brute force and other attacks are arguments that still hold today. We could not gather data that could be used for a more formal argument regarding the presence of brute force attacks, hence we did not retest this hypothesis, which was rejected by Herley and Florencio (2010).

One consideration that can affect the password strength given a web application's password policy is the amount of user traffic that a web application receives or the number of registered users for a web application. If a web application has a large number of users, then it the number of usernames for the service is likely high. With a large amount of the username pool utilized, a large-scale guessing attack could be deployed in order to try to guess the usernames for web applications. Based on these assumptions, one would assume that with a large number of users that the password policy of a web application would be stricter. We test this hypothesis using the same method employed by Herley and Florencio (2010) as seen in Tables 4 and 5. Tables 4 and 5 shows the password policy strength for the top five web applications for registered users and the five largest U.S. universities. Based on this table, the password policy of accounts with more user traffic tend to have stronger password policies. The only exceptions are Facebook with a password strength of 26.6 bits and the University of Central Florida with a strength of 52.6 bits. These two exceptions do not provide a sufficient evidence to reject this hypothesis, hence we fail to reject the hypothesis that more user traffic or a high number of registered users correlates with stronger password policies. This is different than the findings of Herley and Florencio (2010) who rejected this hypothesis.

In regard to large-scale guessing attacks, one would assume that if the username was publicly available via email address or username lookup, then the attacker would be able to acquire the list of usernames with less difficulty than if the usernames were not public. Since large-scale guessing attacks dispense their brute force guessing attempts evenly to web applications in order to mitigate lock outs and intrusion detection schemes, an attacker who possesses a public list of usernames would have an increased likelihood of a successful attack. If the login credential is an email address, this may be problematic. It is a common tactic to use one's name as the local part of the email address, such as GarrettGutierrez@email-service.com. This especially true for companies who assign emails automatically to employees based on their name. Given this argument, we expect that the fact that usernames are publicly available for a web application means that the password policy strength is stronger than a web application without publicly available usernames. Appendix B shows the table of password strengths for the top social media, top email providers, and universities gathered for the 70 web applications corresponding to the web applications of Herley and Florencio. The password strengths for email providers tends to be in the 40-bit range, social media is in the 19 to 26-bit range, and universities tend to be in the 35 to 65-bit range. These are the type of web applications that Herley and Florencio (2010) considered to have public usernames. Given this data, we cannot support the claim that web applications enforce stricter password policies based on having public usernames. Hence, we reject the hypothesis that web applications with public usernames have stronger passwords as Herley and Florencio did in 2010.

Monetary assets are typically regarded as high value. Thus, web applications that manage finances and that have a large number of assets are expected to be well protected. We tested the hypothesis that web applications with larger assets have stronger password policies. As seen in Table 6, except for Wells Fargo, web applications with high value assets tend to have stronger bit strengths. Since there is only one counterexample to this hypothesis, we fail to reject this hypothesis, which Herley and Florencio rejected in 2010.

Herley and Florencio also examined the whether the value of extractable assets affected the strictness of password policy by analyzing the top phished brands using the assumption that attackers who phish web applications do so because the web applications have an extractable value. We expect web applications with high extractable assets to be have stronger password policies in order to protect those assets. Tables 5 shows the password strength for the top phished brands in 2010 while Table 6 shows the top phished brands in 2016. In both tables, there is no clear indication that web applications with high extractable asset value have stronger bit strength. Therefore, we reject that hypothesis like Herley and Florencio did in 2010.

We did not examine the hypothesis that web applications that bear the consequences for the cost of the account breaches have stronger password policies for the same reasons that we gave for the brute force hypothesis. There is no data that we collected that could address this hypothesis more formally, and Herley and Florencio (2010) claim that the banks and brokerages provide examples that led them to reject that hypothesis.

Revenue is one of the main drivers for any business, and this is no different for online web applications. For applications that receive revenue based on advertising, we expect that these web applications would prioritize making advertising as prevalent as possible in order to maximize profit. In some cases, users logging into the web application in order to access its services represents a revenue opportunity to the web applications. Therefore, web applications that allow users to quickly gain access to their account will likely be able to earn more profit. Thus, we expect the web applications that allow advertising to have weaker password policies so that users can quickly access their accounts. The data from Appendix B shows that accounts that accept advertising have a wide variety of password policies. The range of bit strength ranges from 13 to 45 bits with an average of 29 bits of strength. Out of 28 accounts that accept advertising, 20 accounts had a bit strength of fewer than 35 bits. Since a majority of web applications had relatively weak passwords, we fail to reject the hypothesis that web applications that allow advertising have weaker password policies, which was the same result from Herley and Florencio (2010).

Like Herley and Florencio, we collect data on Google AdWords in order to determine if a web application's willingness to purchase advertising relates to weak password policies. Google Adwords Protocol (https://adwords.google.com/home/) is an advertisement method provided by Google to allow businesses and web applications to purchase certain keywords that when searched by a user, the search result will show the advertisement for the web application as the top result of the search query. According to Alexa.com, Google is the top ranked globally and nationally in terms of traffic, thus web applications that purchase AdWords can have their advertising exposed to a potentially large number of people ("Google.com Traffic Statistics", 2017). As with Herley and Florencio (2010), we searched each web application name with and without spaces using the Google search engine. In addition, we used a new incognito browser after each time that we searched for two different web applications. According to "Why you may not see your ad" (2017), repeated searches and previous search history may cause AdWords to not properly show. By using a new incognito browser for sets of searches, we decreased the chance that an ad word was not shown when there exists an AdWord for a particular web application. An example is the search term for Citi Bank. Using a non-incognito browser did not cause the ad word to show but using an incognito browser allowed the ad word to show. As Herley and Florencio (2010) noted, lack of presence of an ad word does not mean that the account did not purchase an ad word. As Appendix B shows, only 12 accounts visibly purchased AdWords. 7 of the accounts had a password strength fewer than 35 bits while 5 accounts had a password strength between 40 bits and 50 bits. The average password strength was 37 bits. Based on this information, we cannot conclude that purchasing AdWords correlates with weak password policies. Thus, we reject the hypothesis that accounts that are willing to advertise have weaker password policies, which goes against the findings of Herley and Florencio (2010) who failed to reject that hypothesis.

The profit of a business is affected by whether the audience must use the services of the business or whether the business must compete for user traffic. Businesses or accounts that have a captive audience do not have to worry about attracting customers so they may not focus on usability of the service because the user has no other choice but to use that service. On the other hand, accounts that must compete for user traffic only make money if the user chooses that service compared to other services so usability would be a prime concern. Since complex passwords are not as usable for users who want to quickly gain access to their account, we expect that accounts that must compete for users have weaker password policies. Appendix B shows whether users have a choice for services for a web application. Herley and Florencio determined user choice based on the type of web application where they declared no choice for universities applications and government as well as some banks. We follow their methodology for consistency. There are 27 web applications where the user has choice and 18 of those web applications have password strengths fewer than 35 bits while the other 8 web applications have password strengths between 40 and 65 bits. For the other 43 web applications with no user choice, 38 web applications have a password policy strength of over 35 bits with only 5 web applications having a password policy of fewer than 35 bits. Thus, on average, web applications where users have choice have weaker password policies compared to web applications where the users have no other choices. Hence, we fail to reject the hypothesis that web applications where users have choice have weaker password policies, which was the same result as Herley and Florencio (2010).

Table 9

Hypothesis Comparison between 2010 and 2017

Hypothesis	2010 Result	2017 Result
Web applications with high user traffic or number of users have strong password policies	Fail to reject	Reject
Web applications with public usernames have strong password policies	Reject	Reject
Web applications with high assets have strong password policies	Fail to reject	Reject
Web applications with high extractable assets have strong password policies	Reject	Reject
Web applications that accept advertisement have weak password policies	Fail to reject	Fail to reject
Web applications that advertise have weak password policies	Fail to reject	Reject
Web applications where users have choice have weak password policies	Fail to reject	Fail to reject

Statistics

Table 10

Password Strengths for 274 Web Applications and Multistep Authentication Presence

Category	Median Strength	Average Strength	Number of Web Applications
Multistep	41.3594	38.31625	70
No Multistep	26.57542	29.26745	205

Table 11

Password Strengths for Multistep Authentication Web Applications by Category

Category	Median Strength	Average Strength	Number of Web Applications
Banks and Brokerages	36.18948	37.20613	10
College	47.63357	46.61283	19
Email Provider	33.96741	32.13695	8
Gaming	34.31153	36.25275	8
Government	48.95123	52.35047	6
Online Shopping	19.93157	23.62756	3
Social Media	23.2535	23.2535	4
Software Service Provider	28.20264	33.33187	9
Travel	41.3594	33.10948	3

Table 12

Password Strengths for Non-Multistep Authentication Web Applications by Category

Category	Median Strength	Average Strength	Number of Web Applications
Airlines	28.20264	34.65116	10
Banks and Brokerages	31.01955	34.46617	3
College	47.63357	46.11863	10
Computer	26.57542	30.22521	8
Educational	19.93157	21.22664	6
Email Providers	49.60451	46.52284	4
Entertainment	19.93157	19.6733	27
Gaming	33.21928	32.87274	11
Government	49.3594	57.05269	9
Health	30.64548	30.64548	2
Hotels and Motels	34.31153	31.94887	6
Insurance	26.57542	31.84709	3
News and Journalism	19.93157	23.64611	18
Online Shopping	26.57542	28.85275	40
Real Estate	4.70044	12.45908	5
Restaurants	26.57542	29.72413	20
Social Media	19.93157	19.19336	9
Software Service Provider	27.38903	29.19011	6
Travel	26.57542	24.28738	8

In order to better examine the password policy trends, we examine the combination of password strength and multistep options. Tables 10, 11, and 12 show the average and median password strengths for the web applications that offer multistep authentication and those that do not. 70 accounts of 274 web applications (25%) allow multistep and 205 web applications do not (75%). When separating data into categories, we found that some categories only had one account when considering whether accounts offered multifactor authentication. To make the groupings more meaningful, we merged the few accounts from airline and hotels into travel and research into educational.

Based on Table 10, web applications with multistep authentication on average had stronger password policies than web applications without multistep authentication options. From Tables 11 and 12, for almost every category except categories such as Government and Online Shopping, web applications that allow multistep authentication have stronger password policies. This result may be due to the fact that accounts that are willing to invest in security features such as multifactor authentication also want their password mechanisms to be reasonable secure as well.

Table 13

Password Strengths for 229 Web Applications Based on Single-Sign On Usage

Category	Median	Average	Number of Web Application
Single-Sign On	19.93157	24.85569	107
No Single-Sign On	28.20264	32.00492	122

Table 14

Password Strengths for Single-Sign On Web Application by Category

Category	Median Strength	Average Strength	Number of Web Application
Computer	26.57542	31.4418	6
Educational	19.93157	21.22664	6
Email Provider	23.2535	23.2535	2
Entertainment	19.93157	20.54863	21
Gaming	26.57542	30.92915	11
News and Journalism	19.93157	22.8647	16
Online Shopping	19.93157	28.50952	9
Real Estate	12.316	14.74337	4
Restaurants	26.57542	29.76319	6
Social Media	19.93157	19.93157	6
Software Service Provider	27.38903	30.38078	10

Table 15

Password Strengths for No Single-Sign On Web Application by Category

Category	Median Strength	Average Strength	Number of Web Application
Airlines	28.20264	35.261	11
Banks and Brokerages	37.1045	36.70507	4
Computer	26.57542	33.59481	3
Email Provider	45.84913	39.35547	8
Entertainment	19.93157	22.94091	10
Gaming	38.54229	38.92519	8
Government	48.8682	51.82851	9
Hotels and Motels	37.60352	33.67218	6
News and Journalism	19.93157	19.93157	3
Online Shopping	26.57542	27.76369	35
Restaurants	26.57542	29.7074	14
Social Media	19.93157	20.88069	7
Software Service Provider	41.91152	37.84705	4

Another aspect that could affect password policy is the usage of single-sign on. Tables 13, 14, and 15 show the password strengths for various web application based on whether they allow single-sign on. We merged hotels and motels into travel and health into entertainment and research into educational for categories for single-sign on and we merged college to educational and real estate into online shopping for non-single-sign on web application.

Based on the data, it appears that web applications that offer single-sign on use on average weaker passwords than web applications that do not offer single-sign on. It is important to note that not all web applications have the same security needs, thus some web applications may have weaker security practices but they are acceptable to that web application. The categories that have the most single-sign on applications are Entertainment, Gaming, Online Shopping, and Software Service Provider. The categories Gaming and Software Service providers overall have strong password bit strengths while Online Shopping has an overall medium average bit strength and Entertainment has a weak average bit strength. When compared to those same categories for web applications without single-sign on, most of the average password bit strengths increased. In addition, most categories for web applications without single-sign on have stronger average passwords than categories for web applications with single-sign. One reason for this observation could be the web applications that allow single-sign on assume users will use the single sign on in place of the web application's default authentication mechanism. Logging in with the credentials for one web application instead of remembering credentials for many web applications is more usable for the user, hence the web application may assume that users will opt for this option. In addition, the nature of web applications that offer single sign on tends to be related to recreation such as journalism, restaurants, and entertainment. Thus, these web applications may not be as security conscious or require a high level of security. Therefore, relying on the security of another web application's authentication mechanism is acceptable for their security and business model.

Table 16

Password Strengths for 229 Web Applications and Continuous Authentication

Category	**Median Strength**	**Average Strength**	**Number of Web Applications**
Continuous Auth	26.57542	29.05927	77
No Continuous Auth	26.57542	28.37452	152

Table 17

Password Strengths for Continuous Authentication Web Applications by Category

Category	Median Strength	Average Strength	Number of Web Applications
Airlines	43.8723	43.8723	2
Banks and Brokerages	47.63357	42.29623	3
Computer	33.78257	33.78257	2
Educational	34.23014	34.23014	2
Email Provider	45.60352	39.49398	5
Entertainment	19.93157	19.19336	9
Gaming	33.96741	33.59825	10
News and Journalism	19.93157	21.59253	4
Online Shopping	27.38903	29.04906	18
Restaurants	37.1045	36.70507	4
Social Media	19.93157	21.26034	10
Software Service Provider	28.20264	31.92259	3
Travel	26.57542	23.2535	5

Table 18

Password Strengths for Non-Continuous Authentication Web Applications by Category

Category	Median Strength	Average Strength	Number of Web Applications
Airlines	28.20264	33.34738	9
Computer	26.57542	31.69573	7
Educational	19.93157	21.48566	5
Email Provider	41.3594	32.77616	5
Entertainment	19.93157	20.93626	21
Gaming	33.21928	33.81326	9
Government	48.8682	51.82851	9
Hotels and Motels	37.60352	33.67218	6
News and Journalism	19.93157	22.61732	15
Online Shopping	23.5022	26.77836	27
Real Estate	12.316	14.74337	4
Restaurants	26.57542	27.9789	16
Social Media	19.93157	17.71695	3
Software Service Provider	28.20264	32.67529	11
Travel	26.57542	25.13215	5

If web applications require continuous authentication, the user can be burdened by repeatedly re-authenticating into their account. This can be cumbersome for users, especially if their password is complex. The user burden of continuous authentication may influence web applications to relax their password requirements so that users are not overly fatigued. Hence, we investigated how password strength relates to the presence with continuous authentication. Tables 16, 17, and 18 show the password strengths of web applications based on whether they use continuous authentication. For web applications that did not offer continuous authentication, we merged banks and brokerage into online shopping since it only had one web application in its category.

Surprisingly, there does not seem be much of a relationship between password strength and presence of continuous authentication. The average passwords strength for web applications with and without continuous authentication are similar. With the exception of a few categories such as Airlines and Computers, Tables 17 and 18 provide evidence that the average password policy for various categories based on presence of continuous authentication is relatively similar. While web applications that offer continuous authentication have stronger password policies in some of the categories, there is no clear pattern in regard to continuous authentication, thus we cannot claim there is a relation between the use of continuous authentication and the strictness of the password policy.

Table 19

Password Strengths for 274 Web Applications and Presence of HTTPS Usage

Category	**Median Strength**	**Average Strength**	**Number of Web Applications**
HTTPS	27.38903	32.6808	252
Non-HTTPS	19.93157	19.40878	23

Table 20

Password Strengths for HTTPS Web Applications by Category

Category	Median Strength	Average Strength	Number of Web Applications
Airlines	28.20264	34.65116	10
Banks and Brokerages	31.01955	36.57383	13
College	47.63357	46.12838	28
Computer	26.57542	32.15947	9
Educational	19.93157	21.22664	6
Email Providers	41.3594	35.9594	11
Entertainment	19.93157	20.93626	21
Gaming	35.72518	35.2042	17
Government	49.03426	55.1718	15
Hotels and Motels	34.31153	31.94887	6
Insurance	26.57542	31.84709	3
News and Journalism	19.93157	24.57938	12
Online Shopping	26.57542	29.31071	41
Real Estate	25.47556	25.47556	2
Restaurants	26.57542	30.23953	19
Social Media	19.93157	20.44263	13
Software Service Provider	28.20264	31.67517	15
Travel	26.57542	26.69341	11

Table 21

Password Strengths for Non-HTTPS Web Applications by Category

Category	Median Strength	Average Strength	Number of Web Applications
College	55.23562	55.23562	1
Entertainment	19.93157	19.10109	8
New and Journalism	19.93157	22.97853	8
Online Shopping	19.93157	14.39502	3
Real Estate	3.321928	3.781432	3

A user who has a strong password is still vulnerable if the attacker can steal their session information after a user successfully authentications. Thus, web applications should use some form of session protection such as the use of HTTPS in order to mitigate this attack. We examine the relations of password policies with presence of HTTPS in order to address this. Tables 19, 20, and 21 show the password strengths for web applications based on whether they use HTTPS. For HTTPS web applications, we merged health into entertainment, merge research into educational; and for non-HTTPS web applications merged restaurant into online shopping and health into entertainment.

The data from Tables 19, 20, and 21 strongly indicate that web applications that use HTTPS enforce stricter password policies. In every category except College, web applications in the HTTPS category have a stronger average password policy. This makes sense as organizations that desire to protect users' accounts would want to provide session based security as well as the privacy, which HTTPS provides. As stated previously, a strong password is not that useful if an attacker can intercept the session information. Hence, it is not unexpected that web application that do not use HTTPS do not require strong passwords.

Table 22

Password Strengths for 229 Web Applications and Storage of Financial Information

Category	**Median Strength**	**Average Strength**	**Number of Web Applications**
Financial Information	26.57542	29.51079	109
Non-Financial Information	24.91446	27.89568	120

Table 23

Password Strengths for Applications that Store Financial Information by Category

Category	**Median Strength**	**Average Strength**	**Number of Web Applications**
Airlines	32.90308	35.96683	10
Banks and Brokerages	26.57542	31.38019	3
Computer	26.57542	30.179	4
Email Providers	33.96741	30.98411	4
Entertainment	19.93157	18.98245	7
Gaming	26.57542	34.47579	7
Hotels and Motels	34.31153	32.57485	6
News and Journalism	18.2706	18.2706	2
Online Shopping	30.45845	30.55779	34
Restaurants	26.57542	30.96289	12
Social Media	19.93157	18.82426	6
Software Service Provider	36.18948	35.52873	5
Travel	26.57542	23.4343	9

Table 24

Password Strengths for Applications that Do Not Store Financial Info by Category

Category	***Median Strength***	***Average Strength***	***Number of Web Applications***
Computer	26.57542	33.74385	5
Educational	19.93157	25.12694	7
Email Providers	45.84913	39.56905	6
Entertainment	19.93157	19.91588	21
Gaming	34.47223	34.19097	12
Government	49.1138	52.35288	8
Health	30.64548	30.64548	2
News and Journalism	19.93157	22.88757	17
Online Shopping	19.93157	27.08826	13
Real Estate	4.70044	12.45908	5
Restaurants	24.91446	27.866	8
Social Media	19.93157	21.82981	7
Software Service Provider	28.20264	30.83915	9

Web applications that store financial information have monetary value that should be protected against attackers. One method of preventing attackers from accessing the financial data stored on the account is making passwords harder to crack. We examine the relation of the password policy strength with a web application’s ability to store financial information. We merged college with online shopping for web applications with financial information and we merged airlines, banks, and hotels and motels into online shopping and college and research into educational for web applications without financial information.

The data from Tables 22, 23, and 24 do not provide a clear indication of a relationship between password policy strength and ability to store financial information. While the average password strength for web applications that allow users store financial information is slightly higher than the average for web applications without financial information, categories like Computer, Email Providers, and Social Media provide counterexamples to the claim that there is a correlation with password strength and ability to store financial information. It is expected that web applications that store valuable information should offer increased protection over web application that do not hold such valuable data. These results suggest that the value of an account via the ability store financial information is not a major force that drives strength in password policy.

Discussion

The data for 274 web applications indicates that there is a concentration of web application with password strengths fewer than 35 bits. This indicates that many web applications still allow relatively weak passwords for their web applications. When the web applications were separated by domain type, .com web application were much weaker than both .edu and .gov web application, which is expected. However, from our comparison to existing results by Herley and Florencio (2010), there is a stark increase in password strength. From 2010 to 2017, the amount of web applications with fewer than 15-bit strength decreased from 8% to 5%, and there is an increase from 19% to 40% for web applications with bit strengths between 45 and 55 bits. Even when the data is separated by domain type, all web applications in each domain category from 2017 have fewer web applications with low bit strengths and more web application with higher bit strengths. Table 3 reinforces this result by showing that the median password strength for every web application group either stayed the same or increased since 2010. Therefore, the data indicates that password policies are slowly becoming stricter. This may be due to the fact that the public cases of account breaches are applying pressure on online web applications to properly protect their users by implementing more security features.

This data trends support the results of retesting some of Herley and Florencio's (2010) hypotheses. We failed to reject the hypothesis that more user traffic correlates with stronger password policies. Unlike the results from Herley and Florencio who rejected that hypothesis, we found that web applications with high traffic had relatively strong passwords. As explained previously, the publicity of compromised web application by major organizations may be a major factor as to why web applications that rely on user traffic for revenue are willing to sacrifice login usability for security. However, we reject the hypothesis that web application with public usernames have stronger passwords. The password strengths for web applications with public usernames and emails as usernames did not seem to have a pattern relating to password strength. Rather, web application groups had clusters of password strengths where email providers had strict password requirements while social media providers had weak password requirements. This seems to indicate that web application type has bearing on the strength of the password policy and that whether username is public is not taken into much consideration.

Unlike Herley and Florencio, we fail to reject the hypothesis that web applications with high value assets have stronger password policies. It seems that web application providers who own a large number of assets want to protect those assets by improving the security of their authentication mechanism. Unlike 2010, web application providers in 2017 seem to consider their account assets when creating their password policies.

We rejected the hypothesis that web applications with high extractable value have stronger password policies. The web applications with the most phishing attacks had a diverse set of password strengths and no positive correlation with stronger password strength. One reason for this could be that the consequences for user who is the victim of a phishing attack are not significant enough to majorly influence password policy. It appears that these web applications weigh other factors with importance when considering password policy.

Unlike Herley and Florencio, we failed to reject hypotheses accepting advertisements and user choice correlate with weaker password policies. Despite the fact that web application password trends are heading towards stronger policies, web applications that rely on, user choice, and accepting advertisements have relatively weak passwords compared to other web applications. Thus, Herley and Florencio's findings from 2010 seem to hold true today: the profit motive still exerts a pressure on organizations to weaken their password policies so that users can more easily access their services. On a seemingly contradictory note, we rejected the hypothesis that purchasing advertisements correlates with weaker passwords. The web applications that use AdWords had diverse password policies strengths between 19 bits and 49 bits. Over half of the web applications that used AdWords had bit strengths at least greater than 30 bits and many of the web applications fell into the banks and university categories. Since universities and banks have been shown to have stronger password policies, the increase in universities and banks using AdWords explains why we reject this hypothesis while Herley and Florencio failed to reject it.

Limitations

Checking passwords for inconsistent requirements was not performed extensively. Especially during account registration, entering an invalid password could potentially cause a registration error where our account would be registered but there would be no way to login with a valid password. This prevented a full examination on whether posted requirements matched enforced requirements. In addition, previous passwords were not extensively tested. We did not test if previous passwords were allowed for change password, but we tested if current password could be used in change password. Typically, the account provided a notification whether previous passwords are accepted, which we did document. In addition, we could not find reliable password policies for all web application, especially for insurance companies.

Many types of the data recorded by QuantCast in 2010 such as number of registered users and whether an account accepts ads are no longer available today. We used various resources to estimate the number of users for various online web application. Since the sources were various, the data may not be completely accurate. The manual inspection of ads does not guarantee that a web application accepts ads in some cases. Some web application used Google Ad Manager or used for Third-Party content but could be using Ad Manager frameworks for their own ads. Without QuantCast directly collecting the data, we cannot guarantee that accurate of ad acceptance.

CHAPTER 5

SECURITY QUESTION PRACTICES

In our examination of 282 online web applications, we found that 78 online web applications (28%) use security questions in some fashion. Out of the 230 web applications for which we created an account, we gathered security question data for 47 web applications. Of those 47 web applications, only 3 web applications (230 web applications: 1.3%, security question web applications: 6%) implemented security questions in primary authentication in addition to username and password. 11 web applications (230 web applications: 4.8%, security question web applications: 23%) used security questions to provide continuous authentication for web applications, usually for accessing user settings. In terms of fallback authentication, 31 web applications (230 web applications: 13%, security question web applications: 66%) used security questions in some form during password recovery.15 web applications (overall: 13%, security question web applications: 32%) allowed users to recover their password with only their security questions. The average number of minimum security questions for each web application was 1.89 while the average number of maximum security questions that can be registered was 1.97. On average, each web application provides a choice from 10.3 possible questions.

Security Question Classification

One of our goals was to classify security questions that are used in practice into meaningful categories in order to assess their usability and resistance to attack. We based our security question classification on existing classifications proposed by Just (2004), Rabkin (2008), and Reeder and Schechter (2011). In terms of security, Just (2004) proposed the following category: guessing difficulty and observation difficulty. Guessing difficulty is measured as the number of attempts that an attacker would need to guess the answer. Questions that are not guessable should also have answer spaces with distribution that is close to uniform. Observation difficulty is the measurement of how difficult it would be for an attacker to retrieve or observe the answer to the question based on publicly available sources. Similarly, Rabkin (2008) proposed guessable, attackable, and automatically attackable as categories where a question is automatically attackable if the answer can be found on any public social media while a question is attackable if the attacker can learn the answer with a substantial probability. Likewise, Reeder and Schechter (2011) use the categories of guessable, researchable, and known attacker. In terms of usability, Just (2004) defined the categories of applicable, memorable, and repeatable. Similarly, Rabkin (2008) defined the usable categories of inapplicability, not memorable, and ambiguous. Reeder and Schechter (2011) proposed the usable categories of not configurable and forgettable.

Table 25

The Security Question Classification Categories

Category	Definition
Ambiguous	Answer has a variety of possible formats or there is not one specific answer
Ephemeral	Answer likely changed since registering the answer
Guessable Upper Bound	Upper bound on the cardinality of the most probable answer space
Inapplicable	Over 40% of the population cannot provide an answer
Non-configurable	Over 40% of the population cannot recall the answer to the question during setup or cannot provide an answer
Researchable	Answer can be obtained or observed via public records, public social media, or other publicly available resources

Based on these criteria, we defined our own classifications as seen in Table 25. To elaborate, the guessability of a question is defined as the upper bound on the most probable answer space for that question (i.e., if the most common answers covers 50% or more of the answer space). Thus, questions with smaller probable answer sets are more guessable than questions with large and uniformly distributed answers sets.

During our formulation of the classification, we initially included three additional categories: *known attacker friend*, *known attacker family*, and *forgettable*. The category *known attacker friend* encompasses the notion that the answer to the question is known or easily observable by a relatively close friend. One issue with this category is properly defining how close a friend is to a user. Based on how much information a user shares with a friend, there could be wide variation in the knowledge of a known attacker who is a friend. Childhood friends are more likely to know the answers to questions about childhood while more recent friends could answer questions about recent preferences, activities, or trips. Similarly, the *known attacker family* category encompasses the ability of a parent, sibling, child, or spouse to know or easily obtain the answer to the question. This category also has the same problem where the attackers could have completely different levels of knowledge so it is hard to consider all attackers in this category as equally able to answer questions. A spouse should be able to know intimate details about the user's life, but a person can have multiple spouses over different periods of their life. Parents and siblings likely know information about a user's childhood while children are likely to know about more recent preferences, activities, and opinions. Hence, it is difficult to assign a specific value to security questions for the categories of *known attacker friend* and *known attacker family* without making sweeping generalizations or assumptions that do not correctly capture the abilities of all attackers within a specific category.

The category of *forgettable* is related to the categories of *inapplicable*, *ambiguous*, and *ephemeral*. Inapplicable questions cannot be forgettable as they are not able to be configured. It is likely that user would forget the answer to a question if the question is ephemeral. Also, it is likely that a user would not forget all answers to an ambiguous question but may not remember the registered answer or how it was originally formatted. Due to its complexity, the category of *forgettable* requires its own study, which is evident from the fact that Bonneau et al. (2015) studied the *forgettable* category in their research on security questions.

The 47 web applications where we collected security question data yielded 217 unique questions. We consider a question to be unique if it has a unique answer space that is distinct from the other questions. For example, the questions “What is your grandmother's name?” and “What is your paternal grandmother's first name?” as two distinct questions because a user could answer with the full name for either the paternal or maternal grandmother in the first question but the user must reply with the first name of their paternal grandmother in the second question. We classified each of the 217 questions into the categories of *ambiguous*, *ephemeral*, *researchable*, *inapplicable*, and *non-configurable*.

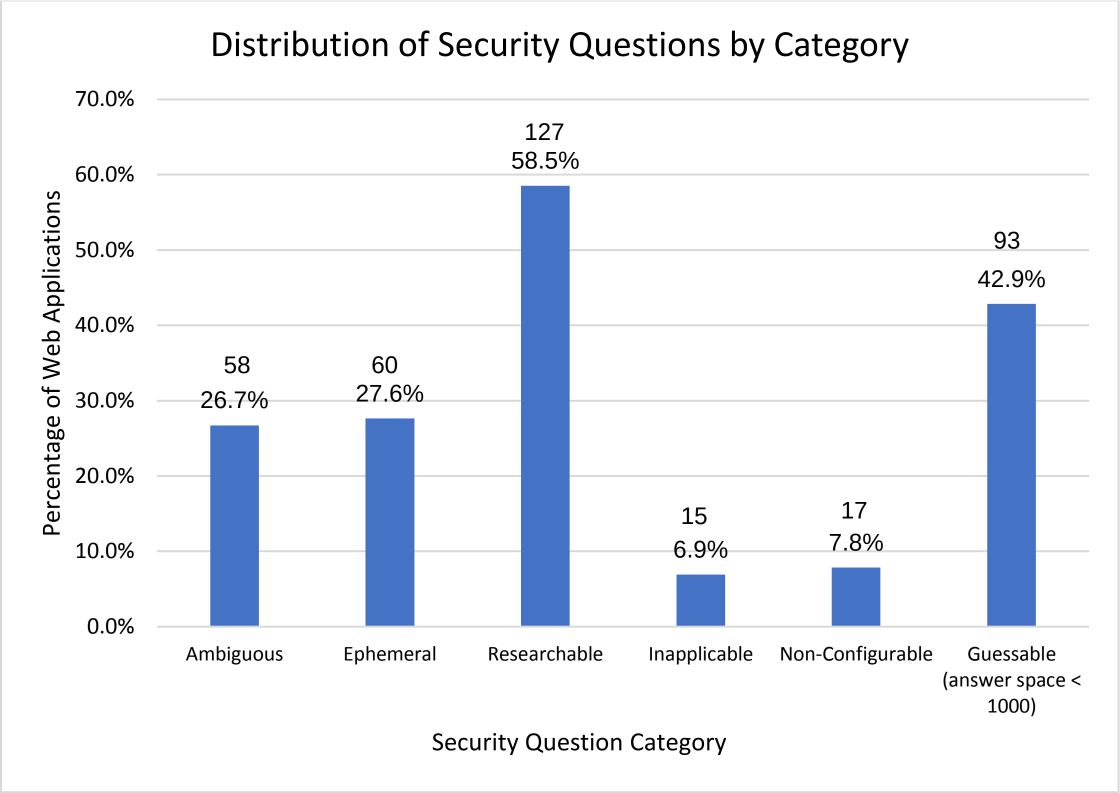

Figure 11. Distribution of security questions by category.

Table 26

Number of Security Questions per Classification

Classification	Number of Questions	Percent
Ambiguous	58	26.7%
Ephemeral	60	27.6%
Researchable	127	58.5%
Inapplicable	15	6.9%
Non-Configurable	17	7.8%
Guessable (answer space < 1000)	93	42.9%

Figure 11 and Table 26 show the number of questions that were classified into each security question category. Since many of the categories are not mutually exclusive, a question could apply to more than one category. For usability, 26.7% of security questions were ambiguous, 27.6% were ephemeral, 6.9% of questions cannot be used since they are inapplicable, and 7.8% are non-configurable. In terms of security, more than half (58.5%) of security questions can be researched via public resources and 42.9% of questions had answer spaces smaller than 1000 answers.

Security Question Practices

Table 27

Usage of Security Questions and Multistep Authentication in Web Applications

Category	Number of Web Applications	Percent
Presence of Security Question and Multistep	37	13.1%
Not Both	245	86.9%

Table 28

Applications Not using Security Questions and Multistep Authentication by Category

Category	Number of Web Applications	Percent
Airlines	1	2.7%
Banks and Brokerages	8	21.6%
Computer	0	0.0%
College	14	37.8%
Educational	0	0.0%
Email Providers	3	8.1%
Entertainment	0	0.0%
Gaming	3	8.1%
Government	4	10.8%
Health	0	0.0%
Hotels and Motels	1	2.7%
Insurance	0	0.0%
News and Journalism	0	0.0%
Online Shopping	1	2.7%
Real Estate	0	0.0%
Research	0	0.0%
Restaurants	0	0.0%
Social Media	2	5.4%
Software Service Provider	0	0.0%
Travel	0	0.0%

Table 29

Applications Not Using Both Security Questions and Multistep Authentication by Category

Category	Total	Percent
Airlines	10	4.1%
Banks and Brokerages	8	3.3%
Computer	16	6.5%
College	8	3.3%
Educational	5	2.0%
Email Providers	8	3.3%
Entertainment	28	11.4%
Gaming	16	6.5%
Government	11	4.5%
Health	2	0.8%
Hotels and Motels	6	2.4%
Insurance	5	2.0%
News and Journalism	19	7.8%
Online Shopping	41	16.7%
Real Estate	5	2.0%
Research	2	0.8%
Restaurants	20	8.2%
Social Media	12	4.9%
Software Service Provider	14	5.7%
Travel	9	3.7%

Tables 27, 28, and 29 show the number of web applications that offer both security questions and multistep authentication options. Only 13% of web applications offered both security questions and multistep authentication options. The categories where both security questions and multistep authentication mechanisms are used the most are Colleges and Banks and Brokerages. This suggests that colleges and financial institutions are likely to use defense in depth for protecting their web applications.

We examined the use of security questions and the presence of CAPTCHAs. While no web application used CAPTCHAs in conjunction with security questions, 5 web applications (230 web applications: 2%, among security questions: 11%), used security questions and employed an CAPTCHA for some other mechanism. The main purpose of using CAPTCHAs security questions is to prevent large-scale guessing attacks. If an attacker uses automated scripts to try to attack multiple accounts on a large-scale, using a CAPTCHA will prevent the automated scripts from being effective since it can be difficult for scripts to bypass more sophisticated challenge response mechanisms. The fact that security questions and CAPTCHAS are not both used to protect an account suggests that automated attacks are not a major consideration when considering account security.

We assessed the use of security questions with presence of HTTPS. For 230 web applications, we found that 42 out of 47 web applications (230 web applications: 18%, among security questions: 89%) use security questions and have HTTPS on at least one page while 26 web applications (230 web applications: 1%, among security questions: 55%) use security questions and use HTTPS on all pages. As explained previously, strong authentication practices are rendered less useful if an attacker can hijack the session. Web applications without HTTPS or without HTTPS on pages with sensitive data can be vulnerable to a man-in-the-middle-attack where the attacker can impersonate the web server by acting as a proxy, which is a security vulnerability (“95% of HTTPS Servers Vulnerable to Trivial MITM Attacks”, 2015).

Another aspect that we examined was the presence of security questions and storage of financial information. Out of 230 web applications where we created an account, 109 (47%) stored at least one type of financial type of data, 47 (20%) used security questions in some role, and 27 web applications (overall: 12%, among security questions: 57%) used security questions and allowed storage of financial information. As seen with correlating password strength to the ability to store financial information, there appears not be a strong relation between using security questions and storing financial information.

Appendix C shows the types of security questions and types of security question answers for the 47 accounts where we could observer the security questions. We use the security question and answer types provided by Just (2004) as defined in Table 30.

Table 30

Types of Security Questions and Security Question Answers

Name	**Definition**
Fixed Question	A list of preset questions is provided to a user
Open Question	User has complete choice and control over the question in free-form text
Controlled Question Add Text	Additional text is allowed to be added, forming a modification of the original question
Controlled Question Hint	User-provided hint, where the hint would be presented to the individual for authentication
Fixed Answer	User selection of an answer from a preset list of answers
Open Answer	User manually enters their response
Controlled Answer Set	Fixed set of answers where the answer space is large enough so that most potential answers are allowed
Controlled Answer Format	User is allowed to enter answer where the answer format is fixed

Appendix D shows the table for other security question practices. Some of these practices include the visibility of the answer before, during, and after typing the answer when editing security questions; the visibility of the answer during and after typing when answering security questions; whether the user must both provide the question and answer; whether the same question to be registered twice with different answers, and availability of security question recovery offline via customer support or online using the web application interface. Practices such as not making the answer visible while or after typing it prevent shoulder surfing attacks, but hamper the usability since a user may not know that they misspelled a word. Allowing the same security question to be registered twice makes answering the question difficult for users since they can provide different answers during registration but may be harder for attackers to correctly guess which answer the user provided. Appendix D also shows which accounts did not prompt security questions, which is why there are null values in a few rows of the columns *Enter Answer Visible while Typing* and *Enter Answer Visible after Typing*.

Discussion

Our examination of security questions shows that many security questions have at least one potential security or usability issue. In terms of usability, a quarter of questions are ambiguous and a quarter of questions are ephemeral. This implies that a significant portion of security questions may be difficult for a user to recall correctly, which may increase false negative authentication rates for web applications. A small number of questions, less than 10%, are either inapplicable or non-configurable. Since web applications on average offer about ten questions from which to choose, this suggests that about one question does not apply to a user, which is not detrimental in terms of usability. However, our classification exposes a potential security issue with security questions. Over 58% of questions can be found via publicly available information. Additionally, about 42% of questions have a small answer space and are relatively easily guessable. Hence, since web applications typically require about two security questions to be registered, there is about a reasonable chance the chosen question chosen is either researchable or guessable by an attacker. This poses a major security risk. Additionally, no web application followed the best practices of using a large pool of registered questions and requiring a subset of questions to be answered.

Based on our analysis of security questions, we have developed recommendations for security questions so that they are not classified as unusable or susceptible to attack. We suggest creating security questions based on the following recommendations. Select questions with a specific answer space and answer format. For example, asking about city name or the first time you visited a different state has a specific answer (city name) given a specific instance (second time visiting a different state). Additionally, the answer to the question should not be available on any common public record. Hence, avoid asking questions about family members. A good example is a question about the title of your most hated sitcom since there is not likely a record that has this information. While it can be difficult to recall first time events, questions about a person's first or second experience can be hard for an attacker to guess. Ideally, questions should have a large answer space where common answers are not easily determined. For instance, "what is your favorite color?" has a small answer space and has popular answers that apply to a majority of the population. Favorites tend to be non-researchable as long as they do not appear on social media, but favorites can potentially change often and can have small answer pool such as the question “What is your favorite food?”. Asking about a favorite for a specific time in the past avoids the issue of answers frequently changing.

Because most web applications use a small answer pool of questions, require about two registered questions, and use questions with known usability and security issues, the current security practices for online web applications do not provide enough security to make security questions viable. In addition, multifactor, HTTPS, and CAPTCHAS are not used in conjunction with security questions, making security question practices vulnerable to hijacking and large-scale guessing attacks.

Limitations

Some web applications do not always prompt the user to answer their security questions. Hence, it is difficult to distinguish between a web application that requires security questions to be set up but does not use them and a web application that uses security questions in specific scenarios outside of primary authentication, fallback authentication, and continuous authentication. This issue may have affected our results for the usage of security questions in practice.

CHAPTER 6

FALLBACK AUTHENTICATION

We collected fallback authentication data for 230 web applications. Out of these 230 web applications, only three did not use any form of fallback authentication mechanism because one used university login mechanisms, another used system generated passwords and did not allow password recovery so a user would have to make a new account if they forget their password, and the other used the password for encrypting the account data so password recovery was not allowed. In terms of offering multiple types of fallback authentication, 31 web applications (14%) provided at least two methods of fallback authentication. For web applications that use email authentication, 221 web applications (97%) use email in password recovery and 191 web applications (83%) only use email recovery. Surprisingly, 29 web applications (13%) send a valid plaintext password via email where 26 applications send a new password while 3 web applications from *Hotels and Motels* and *Real Estate* send the current password. 52 web applications (23%) use defense in depth during password recovery which can involve answering a security question or responding to a challenge response.

Fallback Authentication Usage

Table 31

The Practices for Fallback Authentication Given 227 Web Applications

Fallback Authentication Practice	Number of Web Applications	Percent
At Least Two Fallback Authentication Methods	31	13%
Email Password Recovery	221	97%
Only Allow Email Password Recovery	191	83%
Send Plaintext Password	29	13%
Password Recovery Defense in Depth	52	23%
Temporary Password SMS Password Recovery	2	1%
Call Phone Password Recovery	4	2%
SMS OTP Recovery	20	9%
Personal Info Password Recovery	4	2%

The following are the statistics for the web applications that provide each type of fallback authentication method. 26 web applications (13%) allowed using a phone to recover the user's password. A total of 20 web applications (9%) used security question password recovery with five web applications (2%) used personal information questions not registered by the user and 15 web applications (7%) allowed password recovery using only security questions. In terms of email, 221 web applications (97%) allowed email recovery, 195 of those web applications (86%) used a password reset link, 27 (12%) sent a new plaintext password via email, 6 (3%) used an email one-time password, and 3 (1%) web applications sent the current password in plaintext via email. Some applications allowed using either an email link or an email one-time password so those two recovery options are not mutually exclusive. Out of 227 web applications, only 54 (23%) listed when the password reset link expired, which was an average of 1795 minutes. For 227 web applications, only 4 (2%) provided a user the option to use trustees to recovery access to their web application.

For the 282 web applications, we have partial data on the password recovery for the other 55 web applications. Since we had to rely on publicly posted information, we cannot fully verify that the data for each of these 52 web applications is comprehensive. Bearing this limitation in mind, we provide the statistics on fallback authentication for all 282 web applications. In terms of offering multiple types of fallback authentication, 40 web applications (14%) provided at least two methods of fallback authentication. For web applications that use email authentication, 208 web applications (73%) only allowed email recovery. 63 web applications (22%) used defense in depth during password recovery.

The following are the statistics for the web applications that provide each type of fallback authentication method. Three web applications (1%) allowed a user to recovery their web application using a help desk while five web applications (2%) allowed a user recover their web application in person such as an official university center. In addition, two web applications (1%) can send a temporary password via phone SMS, six web applications (2%) can call a phone with a one-time password, and 24 web applications (9%) can send a one-time password via SMS to the mobile phone. For security question password recovery, eight web applications (3%) used personal information questions not registered by the user and 32 web applications (11%) allowed password recovery with only security questions. In terms of email, 233 web applications (83%)

allowed email recovery, but we do not have complete information about what type of email recovery that could be used.

Some web applications allowed recovery via a threshold of other users confirming the user’s identity. These trusted users are called “trustees”. Considering all 282 web applications, 5 web applications (2%) enable trustees to be used for fallback authentication.

Security of Fallback Authentication

We compare the fallback recovery process of each of the 227 accounts that we created and assess their adherence to fallback authentication best practices. The Open Web Application Security Project (“Forgot Password Cheat Sheet”, 2016) provides a guide for one method of implementing password recovery securely. We use their recommended practices in order to assess the password recovery mechanisms used by each web application. In order to assess n web application’s adherence to these best practices, we assign a score to each web application based on the number of best practices steps that the web application implements.

Table 32

Scoring for Adherence to Fallback Authentication Best Practices

Scoring Fallback Authentication	
Feature	**Score**
Security Question Used	+1
One-Time Token Sent via External Channel	+1
CAPTCHA Used	+1
Log out User after Password Change	+1
Lock Out User until Recovery Complete	-1
Plaintext Password Sent	-1

We assign a point to a web application’s security score for each of the following best practices that are met: using at least one security question during password recovery, using an external channel to send a one-time token, usage of CAPTCHA to prevent large-scale attacks, and logging out the user once the password is successfully changed. If the web application prematurely locks out any user when password recovery is initiated, we subtract a point from their security score because an attacker can perform denial of service attack on that web application, which can occur indefinitely in the worst case. Additionally, we subtract one point if any password is sent in plaintext to the user, which may imply that the password is stored in plaintext on the server. If an attacker compromises the user's email account by attacking the email web application, then the attacker can view the plaintext password for the other account without initiating a password change or password recovery. Therefore, the maximum score for a web application is 4 points while -2 is the minimum score that a web application can receive.

Table 33

Number of Web Applications for a Password Recovery Security Score

Score	**Number of Web Applications**	**Percentage**
4 Points	0	0.0%
3 Points	5	1.8%
2 Points	15	5.3%
1 Point	63	22.3%
0 Points	119	42.2%
-1 Points	14	5.0%
-2 Points	15	5.3%

Table 34

Password Recovery Security Score for Web Applications in Each Category

Category	-2	-1	0	1	2	3	4
Airline	2	0	6	3	0	0	0
Banks / Brokerages	0	1	1	1	1	0	0
Computer	1	0	4	4	0	0	0
College	0	0	0	0	0	0	0
Educational	0	0	2	3	0	0	0
Email Providers	0	0	2	2	5	1	0
Entertainment	1	2	18	6	1	0	0
Gaming	0	0	8	7	2	2	0
Government	0	3	4	1	1	0	0
Health	0	0	2	0	0	0	0
Hotels and Motels	4	1	2	0	0	0	0
Insurance	0	0	0	0	0	0	0
News / Journalism	0	1	15	3	0	0	0
Online Shopping	4	4	20	12	2	1	0
Real Estate	2	0	3	0	0	0	0
Research	0	0	1	1	0	0	0
Restaurant	1	2	15	2	0	0	0
Social Media	0	0	3	8	2	0	0
Software Provider	0	0	5	7	1	1	0
Travel	0	0	7	3	0	0	0
Total	15	14	119	63	15	5	0
Percent	05.3%	05.0%	42.2%	22.3%	05.3%	01.8%	00.0%

Discussion

A majority of web applications rely on email for password recovery. 221 web applications out of 227 used email password recovery and 191 web applications only allowed a user to recover their password via email. Based on these results, the security of these web applications is largely based on the security of the email associated with the account. If that email account were to be compromised, then the attacker could easily compromise the user's other account. This is a potentially vulnerable practice. In addition, only 52 web applications use defense in depth during password recovery. If the password recovery mechanism is considerable weak for a web application, attackers could focus their attention on breaking the password recovery mechanism in order to access the accounts for users. Furthermore, 10% of web applications send a new or current password to the user via email, which implies that the password is not hashed when stored on the server. These are all problematic practices.

Based on our scoring for adherence to best practices, no web application out of 227 web applications received a perfect score of 4. We assumed that web applications did not design their systems considering this specific criterion so it is not surprising that no web applications followed all the best practices completely. Hence, about 7% of web applications follow a majority of the best practices for fallback authentication. The fact that 63 web applications (22.3%) of web applications received a score of 1 suggests that numerous web applications are considering some aspect of security when implementing password recovery. 199 web applications (88%) received a score of 0, which suggest that a majority of web applications do not adhere to any beneficial best practice or cancel out their use of best practices with insecure practices. On the other hand, 28 (12%) web applications had a negative score. This indicates that they implemented a vulnerable practice and did not implement any secure practices.

The categories of web applications that implement the most secure practices are Email Provider, Gaming, Online Shopping, and Software Service Provider. Entertainment, News and Journalism, Online Shopping, and Restaurants are categories where most web applications have a score of 0 or 1. The categories with the worst password recovery practices are Hotels and Motels, Online Shopping, Real Estate, and Restaurants. These results are interesting because we would expect that the Banks and Brokerages, Colleges, and Government categories to implement the most secure mechanisms to protect the value of their web applications. Yet, the results from Table 34 indicate that web applications in these categories vary with regard to following best practices. It is the category of Email Provider that adheres to password recovery best practices. Since many web applications rely on the security of emails for password recovery, our results imply that web applications that rely on the security of the email are more protected due to email provider’s adherence to best practices. While a large number of web applications do not follow or have vulnerable password recovery practices, the situation may not be as serious since many of those web applications fall into categories that may not require high level of security such as entertainment or restaurants.

Limitations

A few web applications have errors with their password recovery. The password recovery process may not work at all, may be initiated but never finish certain steps such as sending a password recovery email, or may have glitches such as claiming that a reset session is invalid after the email link was clicked. Some of the data gathered based on observations rather than specifically labeled options. For example, if a web application did not provide the option to invalidate sessions after a successful password reset, we manually checked if another browser had its session invalidated when the password reset was performed on a different browser.

CHAPTER 7

ALTERNATIVE AUTHENTICATION

Multistep and Multifactor Authentication

Multistep authentication is a mechanism for authentication where the user is granted access if they can provide at least two authentication credentials. Multifactor authentication is a mechanism for authentication where the user is granted access if they can provide at least two credentials for at least two different authentication credential types (knowledge, possession, and inherence). Hence, multifactor authentication is a subset of multistep authentication. For example, a scheme requiring a password (knowledge) and a one-time code sent to a phone (possession since it is a different device) is considered multifactor authentication. However, a scheme requiring password (knowledge) and a one-time password from email (knowledge since it is not a different device) is not multifactor authentication but multistep authentication.

We collected complete multistep authentication data for 230 web applications and collected any posted multistep for the remaining 52 web applications. First, we present the multistep statistics for the 230 web applications for which we created an account. Then, we provide the statistics on multistep data for all 282 accounts while acknowledging the limitation of not having an actual account to verify the data.

Of the 230 web applications, only 39 web applications (17%) had at least one option for multistep authentication with an average of 1.51 multistep methods per web application for web applications that offer multistep authentication. Six web applications (230 web applications: 3%, among multistep: 15%) allowed email one-time password multistep authentication. Four web applications (230 web applications: 2%, among multistep: 11%) allowed multistep using a PIN. Two web applications (230 web applications 1%, among multistep: 5%) allowed security question multistep authentication. Multistep with W2 information, information about credit cards or debit cards, information about 1040 one web application (230 web applications 0.44%, among multistep: 3%). Out of the 39 web applications that allow multistep authentication, 35 web applications (230 web applications: 15%, among multistep: 90%) used multifactor authentication. The average number of multifactor authentication options offered by web applications was 1.26.

34 web applications (230 web applications: 15%, among multistep: 87%, among multifactor: 97%) implemented phone or mobile app one-time password multistep authentication. Three web applications (230 web applications: 1%, among multistep: 8%, among multifactor: 9%) allowed hardware token multifactor authentication. Six web applications (230 web applications: 3%, among multistep 16%, among multifactor: 17%) implemented the Universal Two Factor protocol. One web application (230 web applications 0.44%, among multistep: 3%, among multifactor: 3%) used biometric fingerprints for multifactor authentication.

The following are the multistep statistics for all 282 web applications. The number of web applications that had at least one option for multistep authentication was 70 web applications (24%) with an average of 1.37 options provided per web application that offers multistep authentications. Eight web applications (282 web applications: 3%, among multistep: 11%) allowed email one-time password multistep authentication. Four web applications (282 web applications: 2%, among multistep: 11%) allowed multistep authentication using a PIN. Two web applications (282 web applications 1%, among multistep: 5%) allowed security question multistep authentication. Multistep with W2 information, information about credit cards or debit cards, and with 1040 information each was implemented by one web application (282 web applications 0.44%, among multistep: 3%), respectively. 63 web applications (282 web applications: 22%, among multistep: 90%) implement multifactor authentication with an average of 1.22 options per web application. 56 web applications (282 web applications: 20%, among multistep: 80%, among multifactor: 88%) implemented phone or mobile app one-time password multifactor authentication. 13 web applications (282 web applications: 5%, among multistep: 19%, among multifactor: 21%) allowed hardware token multifactor authentication. Seven web applications (282 web applications: 2%, among multistep: 10%, among multifactor: 11%) implemented the Universal Two Factor protocol. One web application (282 web applications 0.4%, among multistep: 3%, among multifactor: 2%) used biometric fingerprints for multifactor authentication.

Single-Sign On

We provide the single-sin on statistics for the collected 230 web applications. We could not collect single-sign on data for the other 52 web applications so they will not be represented in our statistics.

Table 35

Number of Web Applications for Most Popular Single-Sign On Providers

Single-Sign On Providers	Number of Web Applications	Percent	Percent Among Single-Sign On
Facebook	101	44%	94%
Google	64	28%	60%
Twitter	29	13%	27%
LinkedIn	13	6%	12%
Yahoo	12	5%	11%
Microsoft	8	4%	7%

Out of 230 web applications, 107 (47%) allow some form of single-sign on for authentication. 101 web applications (overall: 44%, among single-sign on: 94%) that use Facebook as the single-sign on provider. 64 online web applications (overall: 28%, among single-sign on: 60%) use Google as the single-sign on provider.29 web applications (overall: 13%, among single-sign on: 27%) use Twitter as the single-sign on provider. The other most popular single-sign on providers are LinkedIn with 13 web applications (overall: 6%, among single-sign on: 12%), Yahoo with 12 web applications (overall: 5%, among single-sign on: 11%), and Microsoft with 8 web applications (overall: 4%, among single-sign on: 7%).

Continuous Authentication

For 230 web applications, 77 web applications (33%) employed continuous authentication.17 (overall: 7%, among continuous authentication: 22%) enforced re-entering the username and password to access the web application settings. One web application (230 web applications: 0.4%, among continuous authentication: 1%) used a one-time password to access web application settings. 11 web applications (230 web applications: 5%, among continuous authentication: 14%) used security questions for continuous authentication. 57 web applications (230 web applications: 25%, among continuous authentication: 74%) required the password to be submitted to change the email address. 6 web applications (230 web applications: 3%, among continuous authentication: 8%) required the password to be submitted in order to change the phone number.

Considering all 282 web applications, 79 (28%) web applications employed continuous authentication. 17 web applications (282 web applications: 6%, among continuous authentication: 22%) enforced re-entering the username and password to access the web application settings. Two web applications (282 web applications: 1%, among continuous authentication: 3%) used a one-time password to access web application settings.12 web applications (282 web applications: 4%, among continuous authentication: 15%) web applications used security questions for continuous authentication. 58 web applications (282 web applications: 20%, among continuous authentication: 72%) required the password to be submitted to change the email address. 6 (282 web applications: 2%, among continuous authentication: 8%) web applications required the password to be submitted in order to change the phone number.

Combination of Practices

Table 36

Number of Web Applications that Use a Combination of Practices

Practice	Number of Web Applications	Percent
Single-Sign On	107	47%
Continuous Auth.	77	33%
Store Financial Info	109	47%
Single-Sign On and Store Financial Info	37	16%
Continuous Auth. and Store Financial Info	45	20%

We examined the presence single sign-on authentication and the ability to store financial information on the web applications for the 230 web applications for which we could create an account. 37 web applications (16%) stored at least one form of financial information on the account. This indicates that web applications that use single-sign are not web applications that typically store financial information and hence do not store many assets related to monetary value.

Additionally, we examined the presence of continuous authentication and the ability to store financial information for the 230 web applications for which we could create an account. 45 web applications (20%) use continuous authentication and allows storage of at least one piece of financial information. If a web application does not implement continuous authentication, then an attacker who gains access to the session of the account without having the credentials for the account will be able to compromise the account since they will not be required to re-authenticate before account changes such as changing the password. By using continuous authentication, more effort is required by the attacker in order to compromise the account because the attacker would be asked to re-authenticate before accessing sensitive information or changing credentials. Thus, web applications that store valuable financial information would be more secure if they implemented continuous authentication. Since one-fifth of web applications use continuous authentication when they allow storage of financial information, only a portion of web applications follow this secure practice.

Limitations

For multistep authentication, we gathered data from publicly posted sources for 52 of the 282 web applications. Thus, we were unable to verify the use of multistep authentication methods for those web applications. We did not make an account using a single-sign on provider for authentication because we wanted to test each web application’s password policy, password recovery mechanism, and authentication mechanism. We checked the usage of continuous authentication for credential change, accessing financial information, and accessing the profile pages. If a web application uses continuous authentication on for a page or purpose that is not apparent, then we did not record that data.

CHAPTER 8

OTHER SECURITY PRACTICES

To have a more complete picture of security practices, we recorded the additional security practices that each web application provides. Specifically, we report on HTTPS usage statistics, session management, multistep credential recovery, CAPTCHAS, ability to change credentials, deletion of accounts, and storage of financial information.

HTTPS

For the 230 accounts that we created, 206 web applications (90%) used HTTPS on at least on page. 127 (overall: 55%, among any HTTPS: 62%) used HTTPS on all web pages. Five web applications (overall: 2%, among any HTTPS: 2%) only use HTTPS at login only. 59 accounts (overall: 27%, among any HTTPS: 29%) use the extended value SSL Certificates for HTTPS.

For 282 web applications, 258 web applications (91%) use HTTPS on at least one page. 51 web applications (overall: 53%, among those that use HTTPS: 59%) use HTTPS on all pages. Six web applications (overall: 2%, among those that use HTTPS: 2%) used HTTPS only at login. 71 web applications (overall: 25%, among those that use HTTPS: 28%) use the extended value SSL Certificates for HTTPS.

Session Management

We recorded data on the ability to invalidate active sessions for each web application and whether web applications allowed the session to persist on multiple tabs for the 230 web applications for which we could create an account. 22 (10%) of web applications allowed some form of session management. 225 web applications (98%) allowed the session to persist on multiple tabs for the same browser.

Multistep Credential Recovery

Another security feature we examined was multistep authentication credential recovery. This is often implemented using backup codes, which are a series of one-time codes that should be printed or saved for later authentication. Backup codes can be used for fallback authentication, yet, in this context, backup codes were used for authentication when multistep authentication is no longer working or accessible.

Table 37

Type of Multistep Credential Recovery Mechanism Offered for 230 Web Applications

Multistep Credential Recovery	**Number of Web Applications**	**Percent Overall**	**Percent Among Multistep Recovery**
Backup Codes	12	5%	33%
Customer Support	14	6%	33%
Trusted Device	2	1%	5%
Recovery Code	4	2%	11%

For 230 web applications, 12 web applications (5%) used backup codes. 36 web applications (overall: 16%, among multistep: 85%) allowed for multistep credential recovery. 14 web applications (overall: 6%, among multistep: 33%) used help desk or customer service as multistep credential recovery. Two web applications out of 230 (overall: 1%, among multistep: 5%) required multistep credential recovery using a device that was already listed as trusted or logged in. 4 web applications (overall: 2%, among multistep: 11%) used only a single recovery code. 0 web applications of 230 used Duo's multistep credential recovery method.

For 282 web applications,15 web applications (5%) used backup codes. 57 (overall: 20%, among multistep: 81%) allowed multistep credential recovery. 22 web applications (overall: 8%, among multistep: 31%) used help desk or customer service as multistep credential recovery. 2 web applications (overall: 1%, among multistep: 3%) required multistep credential recovery using a device that was already listed as trusted or logged in. 5 web applications (overall: 2%, among multistep: 7%) used only a single recovery code. 13 web applications (overall: 5%, among multistep: 20%) used Duo's multistep credential recovery system

CAPTCHA

The usages of CAPTCHAS, challenge response mechanisms used to prevent automated attacks, were measured for the 230 web applications for which we could create an account.

Table 38

Number of Web Applications for Each Type of CAPTCHA Usage

CAPTCHA Usage	**Number of Web Applications**	**Percent**	**Percent Among CAPTHCA**
Login	10	4%	17%
Password Recovery	26	11%	43%
Registration	39	17%	65%

Out of 230 web applications, 60 web applications (26%) web applications used CAPTCHA in some form. 10 web applications (230 applications: 4%, among CAPTCHA: 17%) used CAPTCHAs during login. 26 (230 applications: 11%, among CAPTCHA: 43%) web applications implemented a CAPTCHA in password recovery. 39 web applications (230 applications: 17%, among CAPTCHA: 65%) used CAPTCHAS for registration.

Considering all 282 web applications, 69 web applications (25%) of web applications used CAPTCHA in some form. 11 web applications (282 applications: 4%, among CAPTCHA: 16%) of web applications used CAPTCHAs during login. 34 (282 applications: 12%, among CAPTCHA: 50%) web applications implemented a CAPTCHA in password recovery. 40 web applications (282 applications: 14%, among CAPTCHA: 58%) used CAPTCHAS for registration.

Login and Password Recovery Credentials

We examine whether the login credentials could be changed. If a login credential that cannot be changed is compromised, then an attacker will always have a valid credential for the web application and must only find a valid password in order to gain access into the account. On the other hand, the attacker could change the login credential and prevent the user from knowing the login credential, which can prevent the user from logging into their account.

Table 39

Number of Web Applications that Use Each Type of Login Credential

Login Credential Usage	Number of Web Applications	Percent
Email	197	86%
Username	61	26%
Account Number	14	6%
Phone Number	5	2%
At Least Two Login Credential Options	44	19%

Based on the 230 accounts that we created, the most common login credential was email with 197 web applications (86%) using email as the login credential. 61 web applications (26%) used username as the login credential. 14 web applications (6%) used non-changeable account numbers to be used for the login credential. Five web applications (2%) allowed the user to login given a registered phone number. One web application (0.4%) let a user login given their first name, last name, date of birth, and SSN. This web application was the FASFA government web application, and thus the web application already knows the date of births, SSN, and name of students who are able to have a FASFA. Some web applications allowed a user to use one of many credentials in order to login to an account. 44 web applications (19%) allowed a user to use one of at least two credentials in order to login.

166 web applications (72%) gave the user the ability to change the contents of the credential for login. Out of the 166 web applications that allow the user to change the login credential, 67 web applications (overall: 29%, among login credential change: 40%) required the user to authenticate themselves by providing their password in order change the login credential.

Table 40

Number of Web Applications that That Allow Modification of Login Credential

Changing Login Credential	Number of Web Applications	Percent Overall	Percent Among Credential
Email	158	69%	82%
Username	17	7%	25%
Phone	5	2%	100%

While 166 web applications allow the user to change at least one login credential, 44 web applications offer at least two login credential options such as email address and username, which explains why the number of web applications in Table 40 do not have a sum of 166. For web applications that use username as identity, only 17 (overall: 7%, among username login: 25%) allowed the username to be changed. For web applications that use email as the login credential, 158 web applications (overall: 69%, among email login: 82%) permit the user to change their email. 5 web applications (overall: 2%, among phone login: 100%) use phone as a login credential and let the user change their phone number.

Many web applications required some form of credential in order to initiate the password recovery process. Hence, we examined the password recovery credential required to initiate password recovery.

Table 41

Types of Credentials that Initiate Password Recovery for Web Applications

Password Recovery Credential	Number of Web Applications	Percent
Uses Email	208	90%
Email Sufficient	197	86%
Uses Username	48	21%
Username Sufficient	43	18%
Uses Phone	15	7%
Phone Sufficient	14	18%

Out of 230 web applications, 208 web applications (90%) required at least email as a credential to initiate password recovery with 197 web applications (overall: 86%, among email initiating recovery: 95%) only requiring email to initiate password recovery. Alternatively, 48 web applications (21%) allowed at least username to be used to initiate password recovery. 43 web applications (overall: 18%, among username initiating recovery: 90%) only required a username to be used to initiate password recovery. 15 web applications (7%) allowed a phone number to initiate password recovery with 14 web applications (overall: 18%, among phone initiating recovery: 93%) only requiring a phone number to initiate password recovery. In terms of requiring multiple types of credentials to initiate password recovery, 24 web applications (10%) allowed email or username, 8 (3%) allowed email or phone, 3 (1%) allowed email or account number, 3 (1%) allowed username or phone, and 5 (2%) allowed username or account number. For web applications that required multiple credentials to initiate password recovery, 7 (3%) required email and username while 4 (2%) required email and account number.

Table 42

Usage of Emails for Login and Password Recovery Practices

Usage	Number of Accounts	Percent
Email for Login and Email Initiates Password Recovery	193	93%
Email for Login and Email Password Recovery	192	83%
Email for Login and Only Email Password Recovery	175	76%

We examined the use of email as a sign in credential and usage of password recovery. Out of 230 web applications, 190 (83%) web applications used an email as a login credential and used email as a credential to initiate the password recovery process. 192 web applications (83%) allowed email as a login credential and used an email password recovery process such as emailing a reset link and sending a password to the registered email. Specifically examining web applications that only allow email password recovery, 175 web applications (76%) used email as the login credential.

Considering all 282 web applications, the most common login credential was email with 205 web applications (73%) using email as the login credential. web applications (38%) used username as the login credential. 15 web applications (5%) used non-changeable account numbers to be used for the login credential. Five (2%) of web applications allowed the user to login given a registered phone number. Four web applications (1%) let a user login given their first name, last name, date of birth, and SSN. 79 web applications (28%) allowed a user to use one of at least two credentials in order to login.

Since we could not create accounts for the other 52 web applications, we do not report on how many of those web applications allow the user to change credentials. We still examined the password recovery credential required to initiate password recovery based on the password recovery screen for each web application. Out of 282 web applications, 220 web applications (78%) required at least email as a credential to initiate password recovery with 208 web applications (overall: 74%, among email initiating recovery: 93%) only requiring email to initiate password recovery. For username, 85 web applications (30%) allowed at least username to be used to initiate password recovery with 79 web applications (overall: 28%, among username initiating recovery: 93%) only requiring username to initiate password. 17 web applications (6%) allowed a phone number to initiate password recovery with 14 web applications (overall: 18%, among phone initiating recovery: 82%) only requiring a phone number to initiate password recovery. In terms of requiring multiple options to initiate password recovery, 26 web applications (9%) allowed email or username, 8 (3%) allowed email or phone, 3 (1%) allowed email or account number, 3 (1%) allowed username or phone, and 6 (2%) allowed username or account number. For accounts that required multiple credentials to initiate password recovery, 7 (3%) required email and username while 4 (2%) required email and account number.

For the 282 accounts, we examined using email as the sign in credential and using email to initiate password recovery. 193 (68%) web applications used an email as a login credential and used email as a credential to initiate the password recovery process. 192 web applications (83%) allowed email as a login credential and used an email password recovery process such as emailing a reset link and sending a password to the registered email. Specifically examining web applications that only allow email password recovery, 175 web applications (76%) used email as the login credential.

Account Deletion

The ability to delete or deactivate account can help a user prevent attackers from accessing the account or can allow an attacker to deny the user access to the account. We collected the ability to delete an account via the user interface for 230 web applications. 85 web applications (37%) provided users an explicit method for deleting an account from the user interface. We were unable to collect this data for 52 of the 282 web applications because we were unable to create those accounts. One possible method for deleting an account involves contacting customer support. We did not examine this because an attacker using this method to compromise an account involves social engineering and we do not consider social engineering in our threat model. Examining the security strength of customer support would require an entirely different study.

Storage of Financial Information

For financial information, we collected data on whether accounts hold financial information and allowed the user to change billing and/or shipping addresses. Out of the 282 web applications, we collected this data for the 230 web applications for which we could create an account.

Table 43

Number of Web Applications that Store Each Type of Financial Information

Type of Financial Info	Number of Web Applications	Percent
Credit Card	104	45%
Debit Card	25	11%
Bank Account	9	4%
PayPal	16	7%

109 web applications (47%) stored at least one form of financial information on the web application. 104 web applications (45%) allow storage of credit cards. 25 web applications (11%) allowed storage of debit cards. 9 web applications (4%) allow linking a bank account to the account. 16 web applications allow linking a PayPal account to the account.

We also checked whether a web application allowed the user to change shipping and billing information in order to determine if an attacker can change package destinations by changing the shipping information.

Table 44

Statistics for Address Practices for Web Applications

Address Statistics	Number of Web Applications	Percent
Change Shipping Address	54	24%
Verify Shipping Address	4	2%
Change Billing Address	80	35%

54 web applications (24%) allow users to change their shipping address with 4 web applications (overall: 2%, among shipping addresses: 7%) verifying whether a shipping address is a valid address. 80 web applications (35%) allow a user to store a billing address to the account.

Discussion

As discussed previously, HTTPS is a useful mechanism for achieving privacy of user traffic and for preventing man-in-the-middle attacks. The fact that a majority of web applications employ HTTPS on all pages is a positive trend for web application security practices because user traffic and their session are protected. On the other hand, only 10% of web applications give users direct control of session management, and a majority of web applications allow a user’s login session to persist across multiple tabs of the same browser. If an attacker gains access to an existing user session and there is no session management, then it can be difficult to revoke an attacker’s session on the user’s account. Hence, by not employing proper session management, those web applications are facing a major security vulnerability.

When multistep authentication is implemented, the issue of account recovery for the multistep credential arises. Without a proper mechanism to recover a multistep credential, a multistep credential that is no longer valid could permanently deny the user access to their account. This situation particularly becomes more severe given the case where an attacker knows the login credential. Over three quarters of web applications that provide multistep authentication provide some method of multistep credential recovery. One interesting fact is that a decent portion of multistep credential recovery involves contacting customer support. The difficulty of compromising multistep authentication recovery directly affects the security of primary authentication. For future work, analyzing the security of multistep credential recovery processes would prove useful.

Another web application practice that we examined was the presence of CAPTCHAs. About 25% of web applications employed a CAPTCHA to in some way in their web application. Since CAPTCHAs can help reduce the number of automated and large-scale attacks on accounts, it behooves online web applications to utilize a CAPTCHA in either primary or fallback authentication. However, the number of web applications using CAPTCHA is relatively small, thus a large number of web applications are vulnerable to automated guessing attacks.

Another source of potential concern is the use of emails as the main form of identity for online web application. Email addresses tend to be publicly known, which means that attackers do not need to exert much effort in guessing the login credential and can focus more effort on guessing the password. A large amount of web applications gave users the ability to change their login credentials. By doing so, users can change their login credential if they suspect that an attacker is performing a guessing attack on their account or has even gained access to their account. Yet, the attacker can change the login credential to their own personal credential if they compromise the account. While only about 29% of web applications authenticated the login credential change, it indicates that some web applications are following secure practices. 90% of web applications used the registered email address as one of the credentials that can initiate password recovery. This combined with the fact that 83% of web applications use email as the login credential and can use email to initiate password recover indicates that many web applications are susceptible to attackers who can compromise the user’s email account.

Many web applications do not implement session management, for usage of CAPTCHAs, and for login and password recovery credentials, which is may make them vulnerable to various attacks. With fewer of these proper security practices in place, an attacker has fewer obstacles in place to compromise an account and gain access to the account including sensitive data. Since a portion of web applications allow users to view and change their billing address, an attacker can gain sensitive personal data about the attacker or their financial information just by viewing the billing address. Furthermore, a quarter of web applications allow a user to change the shipping address on their account so an attacker could subtly change the shipping address and redirect packages to the attacker’s desired destination instead of the user’s destination. Overall, it is a good sign that a portion of web applications are implementing secure practices; however; a large majority of web applications do not employ the aforementioned security practices.

Limitations

We did not examine HTTPS network traffic to determine all requests used HTTPS until the final redirect occurred. We assumed that web applications did not only use HTTPS at the end of all the redirects as this would be strange practice that defeats the purpose of HTTPS.

Since we did not fully examine account lockouts and login attempts, we did not completely check if CAPTCHAs were used after a certain number of failed login attempts. Therefore, our CAPTCHA data is not 100% complete. However, the data we do have still has relevance and illustrates the effort by web applications to avoid large-scale guessing attacks for mechanisms such as initially logging in, password recovery, and account registration.

CHAPTER 9

CONCLUSION AND FUTURE WORK

Summary

We performed a survey of the security practices for 282 online web applications for various categories. Out of 282 web applications, we created 230 online accounts. Between June 2016 and April 2017, we examined user registration practices, password policies, multistep authentication options, single-sign on options, fallback authentication options, usage of security questions, presence of HTTPS, presence of continuous authentication, usage of CAPTCHAs, multistep authentication credential recovery, session management practices, and storage of financial information. In addition to those examinations, we compared password policy data, user traffic, phishing data, and advertising data from the research by Herley and Florencio (2010) to the data we collected between 2016 and 2017.

Our results indicate that a majority of web applications do not take advantage of the security mechanisms that can be used to secure online web applications and do not adhere to best practices. Password policies for web applications examined in 2010 have increased in strength in 2017, password policies are weaker for web applications that profit from user traffic, most security questions have issues with usability and guessability, a majority of web applications do not follow the recommended password recovery practices, and only a fraction of web applications use multistep authentication or continuous authentication.

However, there is hope for a more secure cyber future. The comparison of password policy research by Herley and Florencio (2010) to our data suggests that online web applications are making a shift to more secure password policies. Almost every web application category had an increase in password policy strength from 2010. The fact that there is a considerable fraction of web applications that are implementing multistep authentication and that resources such as TwoFactor.org (Davis 2017) suggests that online users are becoming conscious about the need for strong security mechanisms. There needs to be push for web application providers to address security concerns by providing usability security. As account breaches continue to occur, the pressure for web applications to adhere to secure practices will build. While web applications currently do not always implement secure practices, the trends observed in this work suggest that there can be an optimistic future were usable security practices are implemented across the cyber world.

Future Work

Because our work covered numerous aspects about web application security, we did not cover most security mechanisms extensively. This leaves room for a multitude of avenues for future work. For passwords, we noticed that web applications occasionally used password meters when registering passwords for an account. A reasonable study would be to examine the practices for password meters, the effectiveness of password meters, and develop a standardization of password meters. We were unable to project with adequate certainty the trends in password practices throughout the years. A formal study into password policy trends would prove to be an interesting study. Our correlations between password policy strength and presence of multistep authentication and between policy strength and presence of HTTPS led us to believe that formal hypotheses for these correlations could produce meaningful results.

Our classification method for security questions relied on subjective interpretation of the guessability, usability, and researchability of some of the security questions. This was a consequence of not having a user study to validate the degree to which each question can be guessed, configured, and researched. In order to improve our classification method, we recommend performing a user study on the guessability, applicability, ephemerality, applicability, and researchability of security questions used by online web applications.

As a part of our future work, we plan to refine our method for assessing a web application’s adherence to fallback authentication best practices. In addition, we recommend that there be a study of the security and usability of backup codes and other multistep authentication credential recovery mechanisms. On a final note, we did not have a chance to examine the security of single-sign on mechanisms. Since we found that reasonable number of web applications allow single-sign on, we believe that an examination of the usability and security of single-sign on mechanism will yield interesting results.

REFERENCES

“95% of HTTPS Servers Vulnerable to Trivial MITM Attacks”. (2016, May 17). Retrieved from https://news.netcraft.com/archives/2016/03/17/95-of-https-servers-vulnerable-to-trivial-mitm-attacks.html

“2004 annual password survey results”. (2005). Retrieved from http://www.safenetinc.com/news/view.asp?news ID=239

Adams, A., & Sasse, M. A. (1999). Users are not the enemy. *Communications of the ACM*, *42*(12), 40-46.

Asgharpour, F., & Jakobsson, M. (2007). Adaptive challenge questions algorithm in password reset/recovery. *First international workshop on security for spontaneous interaction: IWIISI*, *7*.

Beautement, A., Sasse, M. A., & Wonham, M. (2009, August). The compliance budget: managing security behaviour in organisations. In *Proceedings of the 2008 workshop on New security paradigms* (pp. 47-58). ACM.

Bellovin, S. (2008). Security by checklist. *IEEE Security & Privacy*, *6*(2), 88-88.

Bonneau, J., & Preibusch, S. (2010, June). The Password Thicket: Technical and Market Failures in Human Authentication on the Web. In *WEIS*.

Bonneau, J., Bursztein, E., Caron, I., Jackson, R., & Williamson, M. (2015, May). Secrets, lies, and account recovery: Lessons from the use of personal knowledge questions at google. In Proceedings of the 24th International Conference on World Wide Web (pp. 141-150). ACM.

Brostoff, S., & Sasse, M. A. (2003). “Ten strikes and you’re out”: Increasing the number of login attempts can improve password usability. In Proceedings of CHI 2003 Workshop on HCI and Security Systems.

Clair, L. S., Johansen, L., Enck, W., Pirretti, M., Traynor, P., McDaniel, P., & Jaeger, T. (2006, December). Password exhaustion: Predicting the end of password usefulness. In *International Conference on Information Systems Security* (pp. 37-55). Springer Berlin Heidelberg.

Davis, J. (2017). Two Factor Auth (2FA). Retrieved from https://twofactorauth.org/

Dhamija, R., & Perrig, A. (2000, August). Deja Vu-A User Study: Using Images for Authentication. In USENIX Security Symposium (Vol. 9, pp. 4-4)

Ellison, C., Hall, C., Milbert, R., & Schneier, B. (2000). Protecting secret keys with personal entropy. *Future Generation Computer Systems*, *16*(4), 311-318.

Florêncio, D., & Herley, C. (2006). Klassp: Entering passwords on a spyware infected machine using a shared-secret proxy.

Florencio, D., & Herley, C. (2007, May). A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web* (pp. 657-666). ACM.

Florêncio, D., & Herley, C. (2010, July). Where do security policies come from?. In Proceedings of the Sixth Symposium on Usable Privacy and Security (p. 10). ACM

“Forgot Password Cheat Sheet”. (2016, July 11). Retrieved from https://www.owasp.org/index.php/Forgot_Password_Cheat_Sheet

Frykholm, N., & Juels, A. (2001, November). Error-tolerant password recovery. In *Proceedings of the 8th ACM conference on Computer and Communications Security* (pp. 1-9). ACM.

Gaw, S., & Felten, E. W. (2006, July). Password management strategies for online accounts. In *Proceedings of the second symposium on Usable privacy and security* (pp. 44-55). ACM.

Google.com Traffic Statisitics. (2017, May 6). Retrieved from http://www.alexa.com/siteinfo/google.com.

Griffith, V., & Jakobsson, M. (2005, June). Messin’with Texas deriving mother’s maiden names using public records. In *International Conference on Applied Cryptography and Network Security* (pp. 91-103). Springer Berlin Heidelberg.

Haga, W. J., & Zviran, M. (1991). Question-and-answer passwords: an empirical evaluation. *Information systems*, *16*(3), 335-343.

Haller, N. (1995). The S/KEY One-Time Password System. Proc. ISOC Symposium on Network and Distributed System Security.

Herley, C. (2009, September). So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 workshop on New security paradigms workshop* (pp. 133-144). ACM.

Herley, C., Van Oorschot, P. C., & Patrick, A. S. (2009, February). Passwords: If we’re so smart, why are we still using them?. In *International Conference on Financial Cryptography and Data Security* (pp. 230-237). Springer Berlin Heidelberg.

Holding Companies with Assets Greater Than $10 Billion. (2017, September 30). FFIEC. Retrieved from https://www.ffiec.gov/nicpubweb/nicweb/HCSGreaterThan10B.aspx

Inglesant, P. G., & Sasse, M. A. (2010, April). The true cost of unusable password policies: password use in the wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 383-392). ACM.

Jakobsson, M., Stolterman, E., Wetzel, S., & Yang, L. (2008, April). Love and authentication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 197-200). ACM.

Jermyn, I., Mayer, A. J., Monrose, F., Reiter, M. K., & Rubin, A. D. (1999, August). The design and analysis of graphical passwords. In *Usenix Security* (pp. 1-14).

John the Ripper. (n.d.). Retrieved from http://www.openwall.com/john/

Johnson, B. (2010, May 27). Sarah Palin vs the hacker. The Telegraph. Retrieved from http://www.telegraph.co.uk/news/worldnews/sarah-palin/7750050/Sarah-Palin-vs-the-hacker.html

Just, M. (2004). Designing and evaluating challenge-question systems. IEEE Security & Privacy, 2(5), 32-39

Just, M., & Aspinall, D. (2009, July). Personal choice and challenge questions: a security and usability assessment. In Proceedings of the 5th Symposium on Usable Privacy and Security (p. 8). ACM

Kleucker, M. (2013). Fallback Authentication. Beyond the Desktop, 71 - 78

Most Phished Brands 'Missed' by AntiVirus Based on Big Data Security Intelligence – Q3 2013. (2013). Retrieved from http://cdn2.hubspot.net/hub/241665/file-353945934-pdf/Top_Phished_BrandsQ3_Malcovery_Security.pdf

O'Gorman, L., Bagga, A., & Bentley, J. (2004, February). Call center customer verification by query-directed passwords. In *International Conference on Financial Cryptography* (pp. 54-67). Springer Berlin Heidelberg.

Office of the Privacy Commissioner of Canada. (2006, October). "Guidelines for Identification and Authentication". Available Online. Retrieved from https://www.priv.gc.ca/en/privacy-topics/identity-and-privacy/identification-and-authentication/auth_061013/

Online brands most affected by phishing attacks as of 1st quarter 2016, by share of attacks. (2017). Retrieved from https://www.statista.com/statistics/266359/online¬brands¬most¬affected¬by¬phishing¬attacks/

Pagliery, J. (2013, December 04). 2 million Facebook, Gmail and Twitter passwords stolen in massive hack. CNN News. Retrieved from http://money.cnn.com/2013/12/04/technology/security/passwords-stolen/

PayPal's annual payment volume from 2012 to 2016 (in billion U.S. dollars). (2017). Retrieved from https://www.statista.com/statistics/419783/paypals-annual-payment-volume

Podd, J., Bunnell, J., & Henderson, R. (1996, November). Cost-effective computer security: Cognitive and associative passwords. In *Computer-Human Interaction, 1996. Proceedings., Sixth Australian Conference on* (pp. 304-305). IEEE.

Rabkin, A. (2008, July). Personal knowledge questions for fallback authentication: Security questions in the era of Facebook. In *Proceedings of the 4th symposium on Usable privacy and security* (pp. 13-23). ACM.

Reeder, R., & Schechter, S. (2011). When the password doesn't work: Secondary authentication for websites. IEEE Security & Privacy, 9(2), 43-49.

Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. *BT technology journal*, *19*(3), 122-131.

Schechter, S., Brush, A. B., & Egelman, S. (2009, May). It's no secret. measuring the security and reliability of authentication via "secret" questions. In Security and Privacy, 2009 30th IEEE Symposium on (pp. 375-390). IEEE

Schechter, S., Egelman, S., & Reeder, R. W. (2009, April). It's not what you know, but who you know: a social approach to last-resort authentication. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1983-1992). ACM.

Smartphone thefts drop as kill switch usage grows But Android users are still waiting for the technology. (2015, June 11). Retrieved from http://www.consumerreports.org/cro/news/2015/06/smartphone-thefts-on-the-decline/index.htm

The Most Phished Brands of 2009. (2009, December 19). Avira TechBlog. Retrieved from http://techblog.avira.com/2009/12/19/ the-most-phished-brands-of-2009/en/

Tiwari, A., Sanyal, S., Abraham, A., Knapskog, S. J., & Sanyal, S. (2011). A multi-factor security protocol for wireless payment-secure web authentication using mobile devices

Toomim, M., Zhang, X., Fogarty, J., & Landay, J. A. (2008, April). Access control by testing for shared knowledge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 193-196). ACM.

Vu, K. P. L., Proctor, R. W., Bhargav-Spantzel, A., Tai, B. L. B., Cook, J., & Schultz, E. E. (2007). Improving password security and memorability to protect personal and organizational information. International Journal of Human-Computer Studies, 65(8), 744-757.

“Why you may not see your ad”. (2017). Retrieved from https://support.google.com/adwords/troubleshooter/1711301?hl=en.

Zurko, M. E., & Simon, R. T. (1996, September). User-centered security. In *Proceedings of the 1996 workshop on New security paradigms* (pp. 27-33). ACM.

APPENDIX A

AVERAGE AND MEDIAN PASSWORD STRENGTH FOR EACH ACCOUNT CATEGORY

Category	***Median Password Strength***	***Average Password Strength***	***Number of Accounts***
Airline	*28.20264*	*35.261*	*11*
Banks and Brokerage	*31.01955*	*36.47735*	*11*
College	*47.63357*	*43.76181*	*29*
Computer	*26.57542*	*32.15947*	*9*
Education	*19.93157*	*21.48566*	*5*
Email Provider	*41.3594*	*35.9594*	*11*
Entertainment	*19.93157*	*19.68252*	*28*
Gaming	*33.21928*	*34.32686*	*19*
Government	*49.03426*	*55.1718*	*15*
Health	*30.64548*	*30.64548*	*2*
Hotels and Motels	*37.60352*	*33.29323*	*7*
Insurance	*23.2535*	*23.2535*	*2*
News and Journalism	*19.93157*	*24.39875*	*19*
Online Shopping	*26.57542*	*28.15667*	*43*
Real Estate	*4.70044*	*12.45908*	*5*
Research	*19.93157*	*19.93157*	*1*
Restaurant	*26.57542*	*30.29015*	*20*
Social Media	*19.93157*	*20.44263*	*13*
Software Service Provider	*28.20264*	*32.514*	*14*
Travel	*26.57542*	*23.4343*	*9*

APPENDIX B

ACCOUNT DATA FOR 70 ACCOUNTS FOR HYPOTHESIS TESTING

Website	Password Strength	Accept External Advertising	Adwords	User Choice
Adobe	47.63357	1	1	1
Amazon	19.93157	1	1	1
Answers	19.93157	1	0	1
AOL	45.60352	1	0	1
Arizona State University	49.5186	0	1	0
Bank of America	41.3594	0	0	0
CA Jobs	47.63357	0	0	0
CapitalONe	41.3594	0	1	0
Carnegie Mellon University	52.6797	0	0	0
CBS Sports	13.28771	1	0	1
Census Harvester	73.79697	0	0	0
Charles Schwab	31.01955	1	0	0
Citi Banks	31.01955	1	1	0
Colubmia University	47.63357	0	0	0
Cornell University	47.63357	0	0	0
Craigslist	26.57542	0	0	1
eBay	31.01955	1	0	1
Facebook	26.57542	1	0	1
FASFA ED	49.98342	0	0	0
Fidelity	47.63357	0	0	0
Georgia Tech	65.49616	0	0	1
Google	26.57542	1	0	1
Hotmail	41.3594	1	0	1
Internal Revenue Service e-File	48.8682	0	0	0
Intuit (TurboTax, Mint)	52.6797	1	0	1
iPhoto	41.3594	1	0	1
JP Morgan	31.01955	0	0	0
LA Times	19.93157	1	0	1
Michigan Institute of Technology	36.18948	1	0	0
Michigan State University	47.63357	0	0	0
MySpace	19.93157	1	0	1
NASA Hurley	71.45036	0	0	0
National Institute of Health	44.43671	0	0	0
NOAA Hurley	77.11518	0	0	0
NOAA Weather.gov	71.45036	0	0	0
Northwestern University	40.6997	0	0	0
Ohio State University	43.41012	0	0	0
OkCupid	13.28771	1	0	1
Overstock	26.57542	1	1	1
Paypal	26.57542	1	1	1
Pennsylvania State University	41.3594	0	0	0
PGA Tour	19.93157	1	0	1
Princeton University	65.54589	0	0	0
Rock Star	41.3594	0	0	1
SSA	48.8682	0	1	0
Stanford	52.6797	0	0	0
Texas A & M	47.63357	0	1	0
The Weather Channel	19.93157	1	0	1
TreasuryDirect	49.03426	0	0	0
Twitter	19.93157	1	0	1
TypePad	19.93157	0	0	1
UC Berkeley	53.58777	0	0	0
University of Central Florida Hurley	49.03426	0	0	0
University of Florida	47.63357	0	0	0
University of Illinois at Urbana–Champaign	47.63357	0	0	0
University of Minnesota	45.15085	0	0	0
University of Phoenix	47.63357	1	1	0
University of Southern Florida	47.63357	0	0	0
University of Texas	41.3594	0	0	0
University of Washington	45.60352	0	0	0
USAJobs	49.3594	0	0	0
USPS	47.63357	0	1	0
Vanguard	26.57542	1	1	0
Veterans Affair	47.63357	0	0	0
Virginia State University	55.23562	0	0	0
Wells Fargo	19.93157	1	1	0
Wikia	13.28771	1	0	1
Wikipedia.org	3.321928	0	0	1
Yahoo	46.09474	1	0	1
YouTube	26.57542	1	0	1

APPENDIX C

TYTYPES OF SECUIRTY QUESTIONS AND ANSWERS

Website Name	Number of Fixed Question	Number of Open Question	Number of Controlled Question Add Text	Number of Controlled Question Hint	Fixed Answer Question	Open Answer Question	Controlled Answer Set	Controlled Answer Format
Activision	1	0	0	0	0	1	0	0
Alibaba	3	0	0	0	0	3	0	0
American Air.	3	0	0	0	0	3	0	0
AOL	1	0	0	0	0	1	0	0
Apple	3	0	0	0	0	3	0	0
Barnes & No.	1	0	0	0	0	1	0	0
Bethesda	1	0	0	0	0	1	0	0
Blizzard	1	0	0	0	0	1	0	0
Buffalo Wild	0	1	0	0	0	1	0	0
CA Jobs	1	0	0	0	0	0	0	0
Census	1	0	0	0	0	0	0	0
Delta Airlines	2	0	0	0	0	2	0	0
Denny’s	1	0	0	0	0	1	0	0
EA	1	0	0	0	0	3	0	0
eBay	3	3	0	0	0	3	0	0
Facebook	0	0	0	0	0	0	0	0
FASFA ED	3	2	0	0	0	4	0	1
GMX	1	1	0	0	0	1	0	0
Hawaiian Air	3	0	0	0	0	3	0	0
Intel	1	0	0	0	0	1	0	0
IRS e-File	4	0	0	0	0	4	0	0
IRS e-Serv	3	0	0	0	0	3	0	0
Intuit	1	0	0	0	0	1	0	0
Jet Blue	2	0	0	0	0	2	0	0
Kay Jewelers	3	0	0	0	0	3	0	0
Mail.com	1	1	0	0	0	1	0	0
Nexon	0	0	0	2	0	2	0	0
Olive Garden	1	0	0	0	0	1	0	0
Outback Ste.	1	0	0	0	0	1	0	0
Paypal	2	0	0	0	0	2	0	0
Pizza Hut	1	0	0	0	0	1	0	0
QVC	3	0	0	0	0	3	0	0
Rakuten JP	1	1	0	0	0	1	0	0
Rakuten	0	0	0	0	0	0	0	0
Southwest	2	0	0	0	0	2	0	0
Spirit Airlines	0	0	0	0	0	0	0	0
StackOverfl.	0	0	0	0	0	0	0	0
Uber	0	0	0	0	0	0	0	0
USA Today	0	0	0	0	0	0	0	0
USAJobs	3	0	0	0	0	3	0	0
USPS	2	0	0	0	0	2	0	0
Virgin Air.	0	0	0	0	0	0	0	0
Wendy’s	1	0	0	0	0	1	0	0
Wikia	0	0	0	0	0	0	0	0
Yahoo	0	0	0	0	0	0	0	0
Yelp	0	0	0	0	0	0	0	0
Zynga	0	0	0	0	0	0	0	0

APPENDIX D

SECURITY QUESTION PRACTICES

Website Name	Edit Answer Visible before Typing	Edit Answer Visible while Typing	Edit Answer Visible after Typing	Edit Show Current Question	Enter Answer Visible While Typing	Enter Answer Visible after Typing	Optional to Show Answer	Asks for Question and Answer	Number of Questions Asked	Cannot Use Old Answers	Can Have Duplicate Questions	Offline Security Question Recovery	Online Recover Security Questions	Advice Provided	Does Not Prompt Question Every Time	Never Prompts for Question
Activision	0	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0
Alibaba	0	1	1	0	null	null	0	0	0	0	0	0	0	0	1	1
American	0	1	0	1	1	1	0	0	3	0	0	0	0	0	0	0
AOL	1	1	1	1	1	1	0	0	1	0	0	0	1	0	0	0
Apple	0	1	1	0	1	1	0	0	2	1	0	0	1	0	0	0
Barn. & N	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0
Bethesda	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0
Blizzard	0	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0
Buff. WW	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0
CA Jobs	0	1	1	0	null	null	0	0	1	0	0	0	0	0	1	1
Census	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Delta Air	1	1	1	1	1	1	0	0	2	1	0	0	0	0	0	0
Denny’s	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
EA	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0
eBay	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0
Facebook	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FASFA ED	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0
GMX	0	1	1	1	1	1	0	0	1	0	0	0	0	1	0	0
Hawaiian	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
Intel	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0
IRS e-File	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRS e-Ser	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Intuit	0	1	1	1	null	null	0	0	1	0	0	0	0	0	1	1
Jet Blue	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0
Kay Jewl.	0	0	0	0	null	null	0	0	3	0	0	0	0	0	1	0
Mail.com	0	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0
Nexon	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Olive Gar.	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0
Outback	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0
PayPal	0	1	1	0	1	1	0	0	2	0	0	0	0	0	0	0
Pizza Hut	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1	0
QVC	1	1	1	1	1	1	0	0	1	0	1	0	0	0	0	0
RakutenJP	0	0	0	0	null	null	0	0	0	0	0	0	0	0	1	1
Rakuten	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Southwest	0	0	0	0	1	1	0	0	2	0	0	0	0	0	0	0
Spirit Air.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
StackOv,	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Uber	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
USAToday	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
USAJobs	1	1	1	1	0	0	0	0	0	0	0	0	1	0	1	1
USPS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Virgin Air	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wendy’s	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0
Wikia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Yahoo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Yelp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zynga	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0