Zero Factor Authentication: A Four-Year Study of Simple Password-less Website Security via One-Time Emailed Tokens

Steven Andrés

Claremont Graduate University, School of Information Systems and Technology, 150 East Tenth Street, Claremont, California, United States of America

ARTICLE INFO

Article history:
Received __ July 2015
Received in revised form __ July 2015
Accepted __ August 2015

Keywords:
Authentication
Security
Multi-Factor
Passwords
Password Recovery

ABSTRACT

Static passwords are the easiest and most well known form of authentication for websites. But they carry with them some dangers. Users may select weak passwords that open them up for dictionary or guessing. Complex passwords are hard to remember and users may not follow best practices on the construction or secure storage of these complex passwords. Disclosure of a website's password database is disastrous and compounded if users have used the same password across many different websites. Our solution utilizes the same process as the "forgot password" recovery method, but removes the password entirely. In effect, the control of the user's inbox becomes their shibboleth to prove access to the website account. If the process is good enough for recovery, why not use it for authentication in the first place? Through longitudinal studies at a large university and a cybersecurity firm, we were able to demonstrate an increase in user satisfaction and security while reducing helpdesk support technician burden. The software is open source and available in a single-sign-on (SSO) compatible edition as well.

1. INTRODUCTION

With the rapid proliferation of websites and web-based applications, Internet users are performing a growing number of login actions on a regular basis. As the easiest and cheapest way of authenticating an end-user, password-based authentication methods have been consistently chosen by almost every new online service (Zhu et al. 2014). To protect against password guessing, users are asked to create complex strings of mixed upper- and lowercase letters, plus numerals, and special punctuation symbols. Creating a password with sufficient entropy is not a trivial task, so it is tempting for someone to re-use a complex but easy-to-remember password at multiple websites (Notoatmodjo and Thomborson 2009).

At the same time that users are asked to interact with more websites with increasingly complex password requirements (and requirements that shift between websites), they are also strongly advised to use unique passwords at every website to prevent catastrophic compromise of an individual's online persona by re-using credentials stolen from one database at another website (Grawemeyer and Johnson 2011; Honan 2012). The cognitive overload in such cases can be very stressful and have a negative impact on the users' perception of their information security and privacy (Adams and Sasse 1999). Individuals fail to comply with recommended best practices and use the same credentials because it is too challenging for them to remember so much account data (Ingle et al. 2014).

When a website does suffer a password breach (which have happened with some regularity in recent years), these passwords are immediately used at other websites in an attempt to discover accounts with password re-use. In this manner, a high-value website can be compromised by an unrelated breach at a low-value website (Bailey et al. 2014). There are numerous calls for improving the existing authentication mechanisms and making security more user-centered (Gaw and Felten 2006; Yan et al. 2004; Zurko and Simon 1996), yet not much has been done to explore the problem.

To address the issue of increased cognitive load from additional unique passwords and the increased risk of password data breach, this paper demonstrates how a password-less authentication system (cheekily named "ZeroFactorAuth") can be implemented to greatly simplify the user's authentication process on websites by sending securely generated one-time authentication tokens via email. At the same time, website operators enjoy the elimination of risk for a data breach of sensitive password information. The name is an intentional nod towards the higher security of two-factor authentication, which carries with it some drawbacks in terms of usability.

Design science principles (Hevner and Chatterjee 2010) are used to design, develop, and evaluate the system. We use qualitative methods to obtain feedback on the effectiveness of ZeroFactorAuth and demonstrate its utility and value to end-users and industry professionals. We conducted semi-structured interviews with faculty members in a technical interdisciplinary division of a very large public university in the Western United States who have used the system for over four years. For additional rigor, six information security experts participated in semi-structured interviews after reviewing the system and examining it for weaknesses. After presenting the prototype at a conference in December 2014, the feedback received was used to develop an improved version using agile software development methods. Version 2.0 was installed on the central project management web portal for a cybersecurity firm. Over the course of five months, all employees, business partners, and customers interacted with the prototype and their experiences were evaluated with semi-structured interviews.

Author's e-mail address: steven.andres@cgu.edu

This completed research makes several important contributions to science. First, it presents a novel approach to improve user authentication by relying on secure email communications and one-time authentication tokens rather than user-selected passwords with varying degrees of complexity. We demonstrate how the problem of cognitive overload can be avoided in a secure and efficient manner. Second, we showcase how design science principles can be successfully utilized to build and evaluate the artifact. And third, we provide a summary of existing literature on alternative password-less authentication by outlining the important features which need to be taken into consideration when offering users a safe, easy, and reliable method of accessing websites and applications.

2. RELATED WORK

2.1. No Password Management

The easiest and most insecure method of credential management is by using just a few passwords committed to memory across all websites. Fifteen years ago, surveys indicated "active web users have to manage about 15 passwords for daily use" (Kanaley 2000) while researchers recommend no more than four or five passwords "that users can be expected to cope with" (Adams and Sasse 1999). Of particular concern is when the same password is used on sites with differing "value" (Bailey et al. 2014) sensitivity levels such as a banking website and an online recipe database. Ives et al (2004) note that "a password, and all the accounts it provides access to, are no more secure than the weakest system using that password." Das and Sahoo (2011) discovered that over 75% of individuals use the same password for social networks and email. When studying "several hundred thousand leaked passwords from eleven web sites," Das et al (2014) found that "43-51% of users reuse the same password across multiple sites." This method is simple and easy to use but suffers from the catastrophic effects of widespread account compromise when any password data breach occurs.

2.2. Manual Password Management

After an estimated double-digit number of credentials, an individual will find it difficult to remember all of them. Invariably, the most likely management system will be a list of passwords stored in a centralized location (e.g., in a file named "password.txt" on the desktop or in a spreadsheet). However, this approach requires the users to manually input data and manage it in case of changes or updates (Ingle et al. 2014). In one study, 14% of self-selected and 66% of random passwords were written down (Yan et al. 2004) which caused additional security risks from securing the physical paper record. In one organization, a *majority* of users admitted to writing down passwords, with a tenuous approval from policies that only states that usernames and passwords should be kept separately (Inglesant and Sasse 2010). Even in the case of a mnemonic-based password where users might select a phrase and input the first letters of each word, the recall rate was low and 10% of users still selected weak passwords. To combat weak passwords, many organizations and government agencies (NIST 2009) mandate forced password changes at regular intervals. These password permutations add to the cognitive load on users and can reduce organizational security if users game the policy by incrementing a numeric counter at the end of a common password (Schneier 2005). Such tactics are very inefficient and create a single point of failure.

2.3. Automated Password Management

A more sophisticated method of credential management is to use a password manager. These automated systems save time by automatically suggesting complex passwords, storing passwords as the user authenticates, and automatically filling out login forms. It is entirely possible, with the use of a password manager, to have a unique password for each and every website and the practice comes highly recommended by a recent US-CERT publication (Huth et al. 2012). However, some users are uncomfortable with "relinquishing control" of their passwords to software, do not feel that they need the password managers, or worry that the password managers provide weaker security (Chiasson et al. 2006). Some web browser-based password manager implementations can actually promote poor security practices, such as "training" a user to enter their master password into an IFRAME injected into a webpage where the URL is not from the trusted password manager, a behavior that is similar to a malicious attack (Li et al. 2014).

2.4. Browser Password Management

Some browsers including Google Chrome, Microsoft Internet Explorer, and Mozilla Firefox have the option to automatically save user credentials. This method suffers from data integrity issues, as many times during reinstallations of the browser or the operating system, data can be lost, corrupted (Ingle et al. 2014) or leaked (Munson 2014). For example, Google Chrome uses the popular SQLite database format, stored in the user's profile directory, that "provides neither secrecy nor integrity" (Gasti and Rasmussen 2012). Mozilla Firefox uses the same file format, but introduces an optional password to improve upon secrecy and cloud synchronization services to address integrity concerns. In some cases, non-technical users "were not comfortable giving control of their passwords to an online entity" (Karole et al. 2011) and preferred a smartphone-based manager, separate from their web browser. Browser password management has been shown to be particularly dangerous in the case of public access computers, where an intrusive dialog box can mislead a naïve user into storing a private password on a public system (Bicakci et al. 2011).

2.5. Multifactor Authentication

For increased security, a multifactor authentication or two-factor authentication can also be used. Solutions from RSA include the SecurID hardware token that produces a random six-digit code that can be requested by a properly configured website as an additional security measure beyond the user's password (Schneier 2005). Even when employing expensive and complex hardware tokens, the database breach suffered by RSA by advanced persistent threat (APT) attackers in May 2011 (Coviello 2011) is widely believed to have disclosed the proprietary hash function (Biryukov 2004) and serial number to randomization seed mappings to cause the compromise of several customers (Rashid 2011). The Google Authenticator mobile app offers a similar second authentication factor in software rather than hardware. While two-factor authentication can fairly be judged to be more secure and trustworthy (De Cristofaro et al. 2013) than our proposal, it does not replace the need for a user to remember a password (initial factor) nor address concerns over password database disclosure. Rather than alleviate risk from the website operator, two-factor authentication requires additional software development efforts and ongoing maintenance on the part of the website operator and an increase in the friction of the login experience on the part of the end-user.

2.6. Federated Identity Single Sign-On

There are some solutions available that allow for "single sign-on" (SSO) with a federated identity (Groß 2003). A typical user example of this is websites that allow you to login using your Facebook or Google account. By invoking these methods, the user is prompted to establish a trust relationship between the SSO provider and the website. In many situations involving the workplace, using a social network (that may be against policy) to perform authentication could be self-defeating. An employee might also be rightly concerned that his employer would have access to his/her social profile as a result of the authentication action (despite the technology of OpenID that flies counter to this perception). Mozilla Persona (formally BrowserID) comes closer to our email-based authentication token system, by using email addresses as identities and issuing public-key certificates for these emails. However, Persona offers in-browser solutions and stores the public-key certificate in the local space of the browser which means it has to be set up multiple times, and this method is not very convenient for the end user (Zhu et al. 2014). Additionally, several attacks against the integrity and confidentiality of the BrowserID system were successfully carried out by researchers using identity forgery, login injection, and various cookie and key cleanup issues (Fett et al. 2014). To participate in a federated SSO, therefore, a user must establish a link between two systems that may cross the work life and personal life spheres, or enable technology currently available only on the Firefox platform and not accessible on public access systems (Hackett and Hawkey 2012)

2.7. Prototype Authentication Systems

Researchers have attempted to design and develop improved authentication systems. However, the majority of these systems are conceptual models and have very limited success in real settings or have not been tested or evaluated by end-users. Some of these software prototypes include Loxin (Zhu et al. 2014), Password-free (Ingle et al. 2014), Kamoflauge (Bojinov 2010), Single Password Protocol (Gouda et al. 2007) and Tapas (McCarney et al. 2012). Even more ambitious are hardware/software "clean slate" redesigns of authentication, such as the Pico solution (Stajano 2011) and the FIDO Alliance, backed by industry heavyweights Google, Intel, Lenovo, and Microsoft (Barrett and Kesanupalli 2013). While it is inspiring to see advancements and research into this area, without a clear path to end-user adoption, most new authentication prototypes will struggle with achieving a critical mass of early adopter users to drive market share, as shown below.

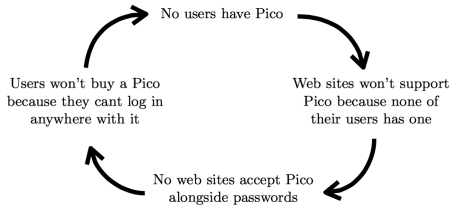

Figure 1. Vicious circle opposing Pico adoption (Stajano et al. 2014)

3. SYSTEM DESIGN

The key concept behind our proposed solution is that email is already used as a secure communication method through which password recovery (e.g., the "forgot password?" links on websites) is performed. Florencio and Herley (2007) show that users forget their passwords and need to reset them quite often. So if users are already using email to reset their passwords, we can remove the additional steps and use the password recovery mechanism as the primary authentication system. If the communication path is trustworthy enough to reset a password, it follows therefore that it should be trusted enough to be the primary authentication system. Moreover, in situations where a password database breach has occurred (Davies 2015; van Elderen 2015; Kan and Shear 2015), the standard practice is to invalidate all existing passwords and force all users to go through the password reset process to establish a new password. Again, if the process is secure enough for the reset, why not use it as the primary method?

3.1. Existing Password Recovery Flow

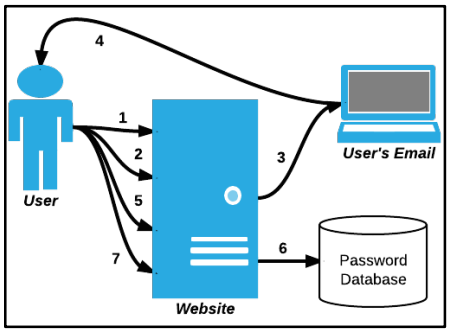

Figure 2. Existing Password-Based Authentication Recovery Method

In the typical existing password recovery method (illustrated above), a user accesses the target website (1) and attempts to login with a password. Due to the cognitive load of several passwords across hundreds of websites, the user might not remember the password and initiates the password recovery process by clicking on the "forgot password?" link (2). The existing website password-based authentication generates a secret token and embeds it within a link sent to the user's email (3). The user retrieves the email (4) by operating the email client software that has the email account preconfigured. The user acts upon the password recovery link (5) provided in the email by clicking (an action that trains the user to fall for phishing[1] links) and is asked to create a new password of suitable length and complexity. The new password is stored in the password database (6). Best practices dictate that only the hash[2] of the password is stored, but some authentication systems store cleartext passwords that make a breach of the database extremely dangerous. Once the new password is set, the website

[1] phishing is a "scam by which an e-mail user is duped into revealing personal or confidential information which the scammer can use illicitly" (Merriam-Webster 1997)

[2] A "hash" is a one-way cryptographic algorithm that represents data by a unique string of fixed-length that is extremely difficult to reverse back to the original cleartext or unencrypted content.

asks the user to login once again (7) with the new password. What is not shown is that the user must now update their password management system (e.g., text file, spreadsheet, web browser-based vault, or smartphone app) with the updated password to avoid the process in the future.

3.2. Proposed Password-less Authentication

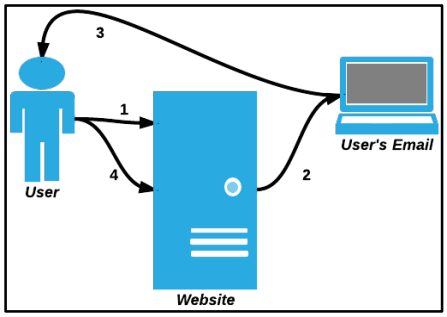

Figure 3. Password-less Authentication

Figure 3 represents the proposed model for password-less authentication. We follow some basic guidelines of successful email-based identification and authentication (Garfinkel 2003). The user accesses the website and provides their email address (1) that has been pre-authorized in a configuration file or database. The ZeroFactorAuth system generates an authentication token and returns half to the user's web browser as a session cookie and the other half via email to the user (2). The user retrieves the email (3) in the same fashion as before and copies the authentication token into the clipboard. The user switches back to the web browser and pastes in the token from the clipboard (4), and the server joins the token with the session token matching the web browser's cookie and IP address to complete the authentication. At no point does the user have to remember anything other than their email address. There is no password database to compromise.

3.3. Online Demonstration

A demo version of the current prototype is available for review at https://ZeroFactorAuth.com with some caveats. The demonstration system allows for end-user self-enrollment but in a real installation the database of users would be controlled by some administrator. A debugging view of cookies is available on the demonstration site by clicking the magnifying glass. The SAML and Shibboleth single-sign-on capabilities have been disabled for this demonstration. In practice, the authentication would happen within the same domain of the target website, so that users are trained to recognize the URL as an indicator of trust in the authentication process. The authentication pages are based on a template system, so practical applications of the technology could add the organization's branding, logo, and colors to make the authentication process more seamless.

3.4. Password Breach Risk Transfer

By not having a password database, this allows for "risk avoidance" for the website operator or more accurately "risk transfer". Currently, the risk of a password database breach is borne completely by the website operator. With ZeroFactorAuth, the risk is transferred to the email hosting provider because the website operator no longer has a database of passwords. Large email hosting providers such as Microsoft, Yahoo, Google, and AOL have elaborate safeguards for password databases and some even allow for enhancement of email password security with a multifactor authentication.[3] ZeroFactorAuth leverages that existing safety and trust around email authentication. Additionally, an attacker will have a much harder time attempting to breach the password database of a large email hosting provider that has dozens of information security staff defending its database.

3.5. Convenience Improvements

Compared to the existing password-based authentication method, ZeroFactorAuth appears to the user to be frictionless and not require any passwords. In reality, the end user's email password is leveraged as the secure communication method. Since most users will access the technology through personal devices, their email client is preconfigured with a stored password. The process of retrieving new emails appears to be password-less to the user because of this. The user does not have to remember or store any additional password for the new website, which reduces the cognitive load. Since the authentication token can be easily cut-and-paste on a mobile device, it improves upon the sometimes clumsy method of entering a complex password using a mobile virtual keyboard. And email is a built-in function of every modern operating system and thus does not require the purchase or installation of any hardware or software, in stark contrast to RSA SecurID, Google Authenticator, or mobile authentication systems[4] that require the installation of a separate application for each provider.

3.6. Multiple Persona Awareness

On many websites, a user is asked to select a username in addition to a password. In other cases, a username is assigned (e.g., on a company or university network). Different website operators may have differing rules for their username selection criteria. A user might be "JohnDoe" on SiteA but that same username represents another person on SiteB and thus our end-user must use "JohnDoe2" or "DoeJohn". During our prototype testing, many users expressed a surprising discomfort in the cognitive load necessary to remember usernames apart from passwords.

Many modern websites address this concern by using an email address as the "username". This has an added benefit of being unique across different websites. But another surprising result from our prototype testing is that expert users had sometimes four to seven email aliases that all forwarded to one inbox. This behavior was particularly prevalent with university faculty, who may have "john.doe@example.edu" as well as

[3] See https://www.google.com/landing/2step and https://help.yahoo.com/kb/SLN5013.html and https://account.live.com/proofs/Manage

[4] See https://www.duosecurity.com and https://www.authy.com and https://www.secureauth.com/Product/Two-Factor-Authentication.aspx

"john.doe@dept.example.edu" or "jdoe@lab.dept.example.edu" (and so forth, ad infinitum).

Even though all addresses eventually arrived in their inbox, they experienced website login failures by forgetting which email address to use for authentication. Information security best practices state that a login failure should not inform whether the username or password or both were invalid. In our testing, the participants informed us that many times they had to "guess" between several emails and several password combinations before arriving at a successful authentication.

Armed with this problem, we designed ZeroFactorAuth with "multiple persona" awareness and allowed several email addresses to map back to one user "object" (which we refer to as the "username" for simplicity). Our university faculty member could use any of their registered email addresses to initiate the ZeroFactorAuth login process. If the email they enter is unknown, they will be provided feedback immediately. Because the user is never prompted for a password at the same time as the email, we do not violate best practices by providing this feedback.

3.7. Security Enhancements

Authentication tokens are for one-time use only, and expire within an inactivity period if not used. Tokens are linked to sessions within a particular initiating browser at the initiating source IP address. In contrast, regular passwords are valid indefinitely. Unless the website employs complex geolocation risk analysis (out of reach for many small sites), a regular single-factor authentication password can be used from any browser and from any source IP address, not just the one that started the login process. Any attempt to brute force the 40-character authentication token used by ZeroFactorAuth would be futile because it would take far too long before all the possibilities of the 160 bit size of U.S. Secure Hash Algorithm 1 (SHA-1) (Eastlake and Jones 2001) could be explored.

Many users, despite policies to the contrary, will send a password over email without encryption. An attacker that can intercept that message or view the email inbox at a later date can use that password for months or years after the email is discovered, provided the user has not changed passwords. ZeroFactorAuth tokens immediately expire upon use, so any subsequent discovery of an authentication token would not provide access.

3.8. Login attempt accountability

Every login attempt with ZeroFactorAuth generates an email to the account owner, whether initiated legitimately or by an attacker. This provides a unique audit log for account owners to police the security of their account. Should an account owner receive an email with an authentication token outside of when they were attempting a login, they would know that someone else is trying to login to their account. The authentication token email provides the IP address of the requestor and instructions to forward the message to a system administrator if they were not the initiator. Armed with that information, an administrator could employ preventative and protective actions such as blocking that malicious IP address. In contrast, traditional password-based authentication normally does not generate email alerts of login activity. On some sensitive websites, an account owner may receive an email after a certain threshold of failed logins have been attempted, but this feature is rare. Successful logins are almost never reported to account owners.

3.9. Caveats

With any authentication system, there are caveats to the design and implementation. This proposal does not seek to be the ideal authentication mechanism in all cases—only to improve on the current status quo of single-factor password authentication. Certainly there are ultra sensitive applications such as banking or military secrets that would not be a target for this system. In those circumstances the organization would prefer to control the entire process and own the entirety of the risk of a password database breach. For these ultra sensitive situations, an air-gapped network disconnected from the Internet with multiple authentication factors is more appropriate.

A slight delay is introduced to the login process as the user waits for the one-time authentication token to arrive in their inbox. This delay is comparable to the delay in receiving an SMS text message from a two-factor authentication system or the time needed to retrieve a one-time token from a hardware or software device and manually input into the authentication system. In our testing, this delay was minimal but we received feedback from our prototype that expert end-users perceived the delay to be substantial. In our expanded research this year, we have more closely benchmarked the delays (as discussed later).

While we acknowledge that email accounts can be compromised through phishing or even password guessing, the status quo situation with single-factor authentication and password reset functionality *already* provides fertile ground for an attacker to compromise all third-party website logins once the email account is hijacked. Said differently, there already exists the possibility of a motivated attacker to *compromise any website* offering a "forgot password" reset function by infiltrating the email account of a victim. Strong protections against email account compromise is outside the scope of this study. ZeroFactorAuth uses this inherent trust in email authentication to the advantage of the user in the form of frictionless authentication and increased accountability.

3.10. Dependency on Email

Because the one-time authentication token is sent to the user's email address, the user's email server becomes a critical part of the authentication mechanism. A user without an email account cannot use ZeroFactorAuth, but will also find the use of the Internet in general to be limited by the lack of an email account. If the user cannot access his/her email due to authentication issues with their email provider, they will be unable to uses ZeroFactorAuth. However, it is assumed by the design that a user with a malfunctioning email or forgotten email password will seek to remedy the situation through existing mechanisms, such as phoning a helpdesk, to reestablish their communications.

During initial testing, expert users expressed concern about email outages. While email outages are extremely rare, they still may occur. Because email is such a vital function in a user's interaction with websites (e.g., receiving confirmation messages), it is assumed that the likelihood of a user wishing to interact with a protected website while their email was malfunctioning to be small compared to the benefits provided by the system. It is important to note that average and novice users were unaware of substantial email outages in the previous few years.

3.11. Comparison

The following table will summarize the benefits and caveats of ZeroFactorAuth compared to other authentication methods.

Table 1. Comparison between existing password authentication (H_0), software/hardware two-factor auth (2FA), and ZeroFactorAuth (ZFA)

Attribute	H_0	2FA	ZFA
Popularity	High	Low	—
Ease of Use	Average	Low	High
Implementation Cost	Low	High	Low
Risk of Password DB Breach	High	High	None
End-User Hardware Req'd	No	Yes	No
End-User Smartphone Req'd	No	Yes	No
Promotes Unique Auth Per Site	No	Yes	Yes
Time-limited and Expiring Auth	No	Yes	Yes
Login Attempt Accountability	Low	Low	High
Auth Linked to IP Address	No	No	Yes
Auth Linked to User Agent	No	No	Yes
Time Delay to Perform Auth	None	Seconds	Seconds

From this table, one can see that ZeroFactorAuth excels in almost every attribute with the exception of popularity and delay in performing authentication. In the latter, the delay introduced is on par with any two-factor authentication mechanism.

4. SYSTEM IMPLEMENTATION

ZeroFactorAuth is implemented as software that runs on a web application server such as Microsoft Internet Information Server (IIS) or the Apache Foundation HTTPD (commonly referred to as "apache"). The software is written in PHP and interpreted at runtime by a support module dynamically linked or loaded with the web application server. This PHP module is widely distributed and sometimes enabled by default, so setup is extremely simple. By placing a few files in a directory and editing a flat configuration file, a website operator can have ZeroFactorAuth working within minutes. Because it does not depend on an underlying RDBMS, the technology can adapt to many different hosting environments.

When a web request is received for a page that requires authentication, the ZeroFactorAuth software is queried for the current authenticated username. If no username exists in the current session, an authentication prompt (login) is displayed using a customizable template. The end-user is asked for his/her email address rather than a username. The configuration file is then queried for a matching username entry (but importantly, no password is ever stored in the configuration file) for the provided email address. Several email addresses may be mapped back to one username object.

Once a valid username object is located, ZeroFactorAuth must create the authentication token. This token is comprised of two components and two indicators, as shown in Figure 4. The indicators (email address and source IP address) are taken from the initiating web browser session, which may be the genuine user or an attacker. The authentication token is comprised of a session identifier and a "validation code" sent to the user via email.

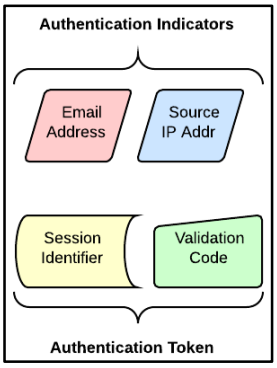

Figure 4. Authentication Components

To avoid the need to perform authentication on every web page request, the concept of a "session" is needed. ZeroFactorAuth implements sessions in PHP by sending a temporary session cookie to the browser using a pseudorandom session identifier, computed using SHA-1 (see Figure 5). This cookie is what links the browser instance to the session object on the server and is an extremely common and well-understood method of establishing a session across any authentication system. By default, the cookies and the session expire within 30 minutes of inactivity (customizable), which provides an additional layer of security when compared to static passwords.

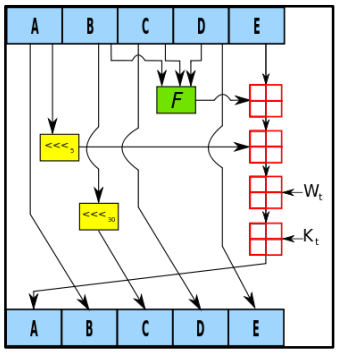

Figure 5. SHA-1 Hash Derivation (Crypto 2015)

Separately, the authentication system generates a unique identifier for the one-time validation code. This is accomplished by computing another SHA-1 hash of the PHP function *uniqid* (Bakken 1999) with enhanced entropy that generates a 23 byte unique string based on the current time of day in microseconds. Additional cryptographic security may be implemented but is deemed unnecessary at this point. The authentication system must only match the token with the browser session and IP address.

Critics may point out that the validation code is theoretically predictable if an attacker could determine the precise microsecond that the victim was intending to login. However, this is only one half of the authentication token. The attacker would also need to predict the SHA-1 hash that identifies the PHP session object for the victim. The probability that the attacker could predict both halves of the authentication token with such precision and could also spoof the network transport layer to appear to originate traffic from the same source IP address is highly unlikely.

5. METHODOLOGY

We identify concepts and theories from prior literature to create the proposed system. Following the design science research cycles proposed by Hevner and Chatterjee (2010), the current study attempts to answer the calls for designing and developing a functional and efficient password-less authentication system. After building the artifact, we evaluate it to ensure it has value to end-users as suggested by Boztepe (2007). Our evaluation centered upon the *utility*, *quality*, and *efficacy* of the artifact (Hevner et al 2004). We made several iterations of the artifact to ensure it is of high quality and meets user needs. We also draw upon the notion that the researcher-practitioner collaboration is key to user adoption and buy-in (Österle and Otto 2010).

We used semi-structured interviews to collect data and to evaluate the proposed authentication system. We conducted the interviews via phone and video conferencing. We used a qualitative method to gather more information about the needs of the end-users and their experience interacting with current password-based systems. This approach helped us to identify practical perspectives towards our artifact and ensure it can be successfully implemented in any organization regardless of its size and objectives.

5.1. Design and Prototype

We selected a very large public university in the Western United States to design and implement the system. For the testing and evaluation we chose a relatively small department to have more interaction with the respondents and to be able to support their needs. The password-less authentication system was designed four years ago (2011) and during that time users provided valuable feedback that led to multiple iterations of the system. This long-term longitudinal approach also helped to reveal any patterns in user behavior and identify areas for improvement.

5.2. InfoSec Expert Panel

In addition to the academic users, we contacted key industry experts with extensive experience related to authentication systems. To provide more consistency, we used an interview guide based on the Delone and McLean model of IS success measures (2003). We used Atlas.ti v7 to analyze the transcripts and to identify the main themes. To address inter-rater reliability issues, the two researchers first agreed on the main themes by reviewing one of the interview transcripts together. Then they analyzed the rest of the data independently. At the end they compared notes and reviewed the codes identified by each researcher to ensure consistency of the results.

5.3. Academic Feedback

The prototype authentication system was presented to the ICIS 2014 Workshop on Information Security and Privacy in New Zealand. Based on the valuable feedback provided by the reviewers and audience, the prototype was enhanced using agile software development methods. The revised software version was implemented on the central project management web portal for a cybersecurity firm in the Western United States. Over the course of five months, all employees, business partners, and customers interacted with the prototype and their experiences were evaluated with the same semi-structured interview protocol as previously employed.

6. RESULTS

The proposed authentication system was positively evaluated by some end-users and a panel of security experts with over four decades of industry experience. We were able to identify several themes that came up in all of the interviews and we grouped them into four main categories: issues, usefulness, quality, and features. It is interesting to note that end-users were more concerned with the usability and simplicity of the system, while the security experts were focused on its features and technical performance. These results are consistent with findings in prior literature which outline the different needs and perceptions of end-users and information security managers (Adams and Sasse 1999; Albrechtsen and Hovden 2009; Werlinger et al. 2009).

6.1. End-User Research

All end-users admitted that they are experiencing serious cognitive overload issues when dealing with password management and often they consider the tradeoffs and whether it is worth creating a new account at all ("I'm not going to do it because I can't remember it."). Such an attitude makes it easy to understand the need for a simple and easy to use authentication system. The end-users emphasized in their responses that ZeroFactorAuth possesses all these features ("so simple", "really usable", "totally intuitive", "I am now more focused on my actual work and not how to get to it"). In terms of quality, the end-users categorized the proposed system as "exceedingly protected" and "excellent". Further, one respondent said: "This [system] may be something that raises the standard of both the quality and the ease of use in not having to write all this stuff down or memorize it."

6.2. Security Experts' Feedback

On the other side, security experts are more concerned with the performance and security features of the system, rather than with the user interaction or cognitive load issues. Experts are mostly looking for reliability, availability, and cost of the security algorithms being implemented. Cost of security is related to investments in security mechanisms, but as one participant said: "The cost is much lower when you don't have to worry about maintaining a database with user passwords, yourself." In terms of performance, experts rated the system as: "far superior to traditional authentication models requiring passwords" and "more secure and easier than most other systems". All participants described the features of ZeroFactorAuth as: "excellent", "great" and "easy to implement." Further, one participant said: "Most sites would benefit from the system because they need good security which prevents data breaches." After conducting a functional review, the experts also provided some valuable suggestions for further improvements, such as considering a more sophisticated algorithm like SHA-2 for increased hash complexity.

6.3. Academic Audience Response

The prototype authentication system was presented to an academic audience in December 2014 during a conference workshop. The technology was generally well received, as with the previous study groups. Strong initial reactions from some in attendance pointed to a perception of a greatly reduced level of security. These reactions were withdrawn, with some surprise, after it was explained that the vulnerability highlighted by the respondent was *already present* on any website that employs the popular

"forgot password" recovery system. After this realization addressed some concerns, audience members mentioned "how simple it appears" (almost to a fault), trying to find a fatal flaw. Expert users who employed password managers (also known as password vaults) were uninterested in the technology. Adoption of password managers is low and we believe this addresses a current need. If adoption of password managers greatly increases, the need for ZeroFactorAuth is inversely reduced.

"Why hasn't this been already implemented?" was a recurring theme with audience members, to which we could only respond that it appears the perception gap might be too difficult to overcome, given the response that we just received in the room. If users perceive ZeroFactorAuth as less secure than current methods, it will have little success gaining a wider adoption. It is important to note that audience members who self-identified as "non-technical" disagreed with the perception that security was reduced and welcomed the removal of some of their cognitive load.

A closing theme of remarks surrounded the availability of email systems and unacceptable delays in receiving authentication messages via email. Two respondents believed use of ZeroFactorAuth would introduce delays of "45 seconds to over a minute" which would prove unacceptable to their user community. We investigated the email delivery concern in early 2015 with a series of benchmarks.

6.4. Email Delivery Benchmarks

The following table summarizes our benchmarks of email delivery time using ZeroFactorAuth with various major email providers and the two private email systems of our university participants and cybersecurity firm participants. We sent ten authentication requests to each email provider at five different times of day. At each time period, we sent two requests within one minute of the other, to remove any temporary network congestion bias. By using different times of day, we could address the bias during times when more users were communicating over the network.

Table 2. Email delivery benchmarks for various providers

Email Provider	Avg. Delay
Google Gmail	3 seconds
Yahoo Mail	3
Microsoft Hotmail	4
AOL Mail	5
large university email	5
cybersecurity firm email	3
arithmetic mean	3.833

6.5. Cybersecurity Firm

During the first half of 2015, our research was expanded greatly by installing ZeroFactorAuth on the central project management web portal for a cybersecurity firm. Over the course of five months, all employees, business partners, and customers interacted with the prototype and their experiences were evaluated using semi-structured interviews after one week, two months, and at the end of the five-month beta program. Again, Atlas.ti v7 was used to analyze the interview transcripts and to identify the main themes.

We were interested in knowing if ZeroFactorAuth could be a "drop-in" replacement for authentication, so we intentionally did not provide extensive end-user education about the change of authentication. The project web portal changed from asking for the typical username and password pairing to simply asking for an email address. Once entered, ZeroFactorAuth would present the prompt for the authentication token using a customized template that included information about the authentication process and a hint to refresh their inbox and look in the "junk mail" folder if the end-user did not receive the email within a few seconds.

During the five-month research period, 4723 logins were attempted and 4691 were successful. Because the authentication tokens authorize a session, most users performed authentication every workday (and some on weekends to check the status of projects). We investigated the 32 failed logins by asking follow-up questions over email to the accounts listed in the log files. Of the 32, nine failed as a result of not entering a complete email address; they entered the username portion up until the @ symbol but not the domain name. Fourteen failures were attributed to six users entering their previous static passwords. All six users admitted they did not read the information presented on the screen and just "blindly typed their password."

The remaining nine failures were all traced to just one user. A self described "anti-technology luddite," the user was a business partner who had inadvertently requested a second authentication token before checking her email. By issuing $token_2$ the prior $token_1$ was now invalidated. The end-user copied-and-pasted $token_1$ when ZeroFactorAuth was expecting $token_2$. When this failed, the user requested another authorization $token_3$ and then checked her email to find $token_2$. The process continued until the user gave up and called the helpdesk. In future iterations, ZeroFactorAuth can be modified to store the most recent x tokens and accept any in this subset for authentication. This configurable leniency is a tradeoff between security and usability, but one that makes sense once we saw the issue in the field.

Although quantitative data shows a very successful test, end-user perception was slightly less enthusiastic. ZeroFactorAuth performed at a 99.32% success rate for authentication compared to a 92.64% success rate with traditional static passwords. The helpdesk at the cybersecurity firm reported 39 trouble incidents for password reset during the same five-month period in the prior year, resulting in a total of 20 hours and 15 minutes of support technician attention. During the beta period, there were zero password reset requests (by design) since there are no passwords stored and thus nothing to reset. There were three support incidents indirectly related to ZeroFactorAuth that resulted in a total of 20 minutes of support technician attention. The support costs on the helpdesk was decreased by 98.36% resulting in greater attention to more challenging technical issues. Using a median hourly wage of $19.72 USD for the geographic region (PayScale 2015), support costs would likely be reduced significantly.

6.6. Single Sign On (SSO) Mode

During our research period with the cybersecurity firm, it was suggested that we add single sign-on (SSO) capabilities to ZeroFactorAuth so that it can interoperate with federated identity infrastructures that power websites of the firm's large Fortune 500 enterprise-sized clients. Through the use of

the SimpleSAMLphp[5] library of tools, we were able to add support for SAML 2.0 and Shibboleth 1.3 with ZeroFactorAuth acting as an Identity Provider (IdP) and coexisting in a larger federated identity program. Additional expansion to Central Authentication Service (CAS), Sun Federated Access Manager (SFAM), and OAuth2 are available but not yet implemented. SSO capabilities are disabled on the online demonstration system and were not part of the five-month beta program at the cybersecurity firm.

6.7. Future Research

Although the current study presents a secure and successful method for password-less authentication, our implementation was limited to two websites and a small population of users. Future research can benefit from exploring the problem into more detail and expanding upon our work. We encourage our colleagues to apply the authentication model using a much larger sample size with more diverse skillset and knowledge on information security. For improved usability, we recommend an email sender implementation that provides DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF) to improve the likelihood that the emails containing the tokens will not be marked as spam. Design science principles suggest continuous work and a number of iterations, so future research can also look into aspects of the problem related to measuring the cognitive overload of using ZeroFactorAuth in comparison to other systems, as well as usability testing and human-computer interaction improvements.

7. CONCLUSION

The current study addresses important security flaws in existing authentication models by applying design science principles for the development of a new and improved password-less authentication system. We propose a viable solution to the problem of remembering a growing number of unique and strong passwords. To avoid the cognitive overload, users often times choose weak passwords or simply reuse them for many different accounts. This creates a significant vulnerability issue which can provide unauthorized access to individuals with malicious intent.

This research makes several important contributions. First, the password-less system aims to solve existing authentication and cognitive load problems by offering users an efficient, simple, and easy to use system. Through two studies, one long-term longitudinal and the other short-term and broad, ZeroFactorAuth has performed well in real world situations. With a reduction in helpdesk support burden and an increase in the rate of successful authentications, the artifact increases security while reducing the burden on end-users and administrators alike.

Second, ZeroFactorAuth addresses gaps in previous research on authentication systems and demonstrates how practitioners can benefit from utilizing a more rigorous scientific approach to improve information security. Third, the study summarizes various authentication methods and describes their application and shortcomings. This comprehensive approach allows us to identify existing issues and tailor ZeroFactorAuth to meet users' needs more successfully. Fourth, we extend existing knowledge by identifying a variance in perceptions and approaches of end-users and security managers regarding important authentication concepts. And finally, we demonstrate how information security practices on user authentication can be improved utilizing established concepts of design science. The study bridges the gap between theory and practice and outlines important guidelines and principles for secure password-less user authentication.

[5] See https://simplesamlphp.org and https://wiki.oasis-open.org/security and https://shibboleth.net

Acknowledgements

The author would like to thank Dr. Miloslava Plachkinova for her assistance with the initial studies in 2014 and joint presentation to the ICIS 2014 Workshop on Information Security and Privacy. Additional gratitude is bestowed upon the faculty of the university who participated in the prototype study, as well as the employees, partners, and customers of the cybersecurity firm who participated in the revised and expanded research study. Jody Marc Cohn, my mentor from undergraduate studies at UCLA, provided technical review and copy editing for which I am grateful. Information security reviews were graciously performed by Josh Lemos and Justin Dolly and I thank them for their support during this study.

REFERENCES

Adams, A., and Sasse, M.A. 1999. "Users Are Not the Enemy," *Communications of the ACM* (42:12), pp. 40-46.

Albrechtsen, E., and Hovden, J. 2009. "The Information Security Digital Divide between Information Security Managers and Users," *Computers & Security* (28:6), pp. 476-490.

Bailey, D., Dürmuth, M., and Paar, C. 2014. Statistics on Password Re-use and Adaptive Strength for Financial Accounts. *Security and Cryptography for Networks* (8642), pp. 218-235. doi:10.1007/9783319108797

Bakken, S. 1999. "uniqid.c" *PHP Source*. Retrieved from https://github.com/php/php-src/blob/master/ext/standard/uniqid.c

Barrett, M. and Kesanupalli, R. 2013. "History of FIDO Alliance" *FIDO Alliance*. Retrieved from https://fidoalliance.org/about/history

Bicakci, K., Atalay, N. B., & Kiziloz, H. E. 2011. Johnny in internet café: user study and exploration of password autocomplete in web browsers. In *Proceedings of the 7th ACM workshop on Digital identity management* (pp. 33-42). ACM.

Biryukov, A., Lano, J., & Preneel, B. 2004. Cryptanalysis of the alleged SecurID hash function. In *Selected areas in cryptography* (pp. 130-144). Springer Berlin Heidelberg.

Bojinov, H., Bursztein, E., Boyen, X., & Boneh, D. 2010. Kamouflage: Loss-resistant password management. In *Computer Security–*

ESORICS 2010 (pp. 286-302). Springer Berlin Heidelberg.

Boztepe, S. 2007. "User Value: Competing Theories and Models," *International journal of design* (1:2), pp. 55-63.

Chiasson, S., van Oorschot, P.C., and Biddle, R. 2006. "A Usability Study and Critique of Two Password Managers," *Usenix Security*.

Coviello, A. W. 2011. Open letter to RSA customers. *RSA*. Retrieved from http://www.sec.gov/Archives/edgar/data/790070/000119312511070159/dex991.htm

Crypto, M. 2015. "SHA-1.svg" *Wikimedia Commons*. Retrieved from https://commons.wikimedia.org/wiki/File:SHA-1.svg

Davies, Craig. 2015. "HipChat Security Notice and Password Reset," *HipChat Blog*. Retrieved from https://blog.hipchat.com/2015/02/01/hipchat-security-notice-and-password-reset/

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

Das, B., and Sahoo, J.S. 2011. "Social Networking Sites–a Critical Analysis of Its Impact on Personal and Social Life," *International Journal of Business and Social Science* (2:14), pp. 222-228.

De Cristofaro, E., Du, H., Freudiger, J., & Norcie, G. 2013. A comparative usability study of two-factor authentication. *arXiv preprint arXiv:1309.5344*.

Delone, W.H., and McLean, E.R. 2003. "The Delone and Mclean Model of Information Systems Success: A Ten-Year Update," *Journal of Management Information Systems* (19:4), pp. 9-30.

Eastlake, D., and Jones, P. 2001. "Us Secure Hash Algorithm 1 (Sha1)." RFC 3174, September.

Fett, D., Kusters, R., and Schmitz, G. 2014. expressive model for the Web infrastructure: Definition and application to the Browser ID SSO system. In *Security and Privacy (SP), 2014 IEEE Symposium on* (pp. 673-688). IEEE.

Florencio, D., and Herley, C. 2007. "A Large-Scale Study of Web Password Habits," *Proceedings of the 16th international conference on World Wide Web*: ACM, pp. 657-666.

Garfinkel, S.L. 2003. "Email-Based Identification and Authentication: An Alternative to Pki?," *IEEE Security & Privacy* (1:6), pp. 20-26.

Gasti, P., and Rasmussen, K. B. 2012. On the security of password manager database formats. In *Computer Security–ESORICS 2012* (pp. 770-787). Springer Berlin Heidelberg.

Gaw, S., and Felten, E.W. 2006. "Password Management Strategies for Online Accounts," *Proceedings of the second symposium on Usable privacy and security*: ACM, pp. 44-55.

Gouda, M., Liu, A., Leung, L., Alam, M. 2007. SPP: An anti-phishing single password protocol. *Computer Networks*. doi:10.1016/j.comnet.2007.03.007

Grawemeyer, B., and Johnson, H. 2011. "Using and Managing Multiple Passwords: A Week to a View," *Interacting with Computers* (23:3), pp. 256-267.

Groß, T. 2003. "Security Analysis of the Saml Single Sign-on Browser/Artifact Profile," *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*: IEEE, pp. 298-307.

Hackett, M., and Hawkey, K. 2012. Investigating User Concerns with Federated Identity Systems. Retrieved from http://dcsi.cs.dal.ca/papers2012/paper17_hackett.pdf

Hevner, A., and Chatterjee, S. 2010. *Design Research in Information Systems: Theory and Practice*. Springer.

Honan, M. 2012. "Mat Honan: How I Resurrected My Digital Life after an Epic Hacking." Wired.

Huth, A., Orlando, M., and Pesante, L. 2012. Password security, protection, and management. *United States Computer Emergency Readiness Team*.

Ingle, D., Patil, V., and Talbat, S. 2014. "Password-Free Login," *International Journal of Computer Applications* (87:17), pp. 26-30.

Inglesant, P. G., & Sasse, M. A. 2010. 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.

Ives, B., Walsh, K. R., & Schneider, H. 2004. The Domino Effect of Password Reuse. *Communications Of The ACM*, *47*(4), 75-78. doi:10.1145/975817.975820

Kanaley, R. 2000. "Locked Out There's Nothing Like Forgetting Your Password. Some People See An Opportunity In Helping Computer Users Avoid The Frustration." *The Philadelphia Inquirer*. Retrieved from http://articles.philly.com/2000-12-28/business/25579923_1_password-web-sites-user-registration

Karole, A., Saxena, N., & Christin, N. 2011. A comparative usability evaluation of traditional password managers. In *Information Security and Cryptology-ICISC 2010*. pp. 233-251.

Kan, J., and Shear, E. 2015. "Important Notice About Your Twitch Account," *Twitch: The Official Blog*. Retrieved from http://blog.twitch.tv/2015/03/important-notice-about-your-twitch-account/

Li, Z., He, W., Akhawe, D., & Song, D. 2014. The emperor's new password manager: Security analysis of web-based password managers. In *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association.

McCarney, D., Barrera, D., Clark, J., Chiasson, S., & van Oorschot, P. C. 2012. Tapas: design, implementation, and usability evaluation of a password manager. In *Proceedings of the 28th Annual Computer Security Applications Conference* (pp. 89-98). ACM.

Munson, L. 2014. "97,000 Bugzilla Email Addresses and Passwords Exposed in Another Mozilla Leak." Retrieved from https://nakedsecurity.sophos.com/2014/08/29/97000-bugzilla-email-addresses-and-passwords-exposed-in-another-mozilla-leak/

National Institute of Science and Technology. 2009. NIST Special Publication 800-118: Guide to Enterprise Password Management (Draft): Recommendations of the National Institute of Standards and

Technology. Retrieved from http://csrc.nist.gov/publications/drafts/800-118/draft- sp800-118.pdf

Notoatmodjo, G., and Thomborson, C. 2009. "Passwords and Perceptions," *Proceedings of the Seventh Australasian Conference on Information Security-Volume 98*: Australian Computer Society, Inc., pp. 71-78.

Österle, H., and Otto, B. 2010. "Consortium Research," *Business & Information Systems Engineering* (2:5), pp. 283-293.

PayScale Research. 2015. "Help Desk Technician Hourly Salary." Retrieved from http://www.payscale.com/research/US/Job=Help_Desk_Technician/Hourly_Rate/72a1fca0

Rashid, F. Y. 2011. Northrop Grumman, L-3 Communications Hacked via Cloned RSA SecurID Tokens. *eWeek*, *6*(2).

Schneier, B. 2005. "Two-Factor Authentication: Too Little, Too Late," *Communications of the ACM* (48:4), p. 136.

Schneier, B. 2005. Write Down Your Password. Retrieved from http://www.schneier.com/blofg/archives/2005/06/write_down_your.html

Stajano, F. 2011. Pico: No more passwords!. In *Security Protocols XIX* (pp. 49-81). Springer Berlin Heidelberg.

Stajano, F., Jenkinson, G., Payne, J., Spencer, M., Stafford-Fraser, Q., & Warrington, C. 2014. Bootstrapping adoption of the pico password replacement system. In *Security Protocols XXII* (pp. 172-186). Springer International Publishing.

van Elderen, T. 2015. "Statement Data Breach," *Brabantia*. Retrieved from http://www.brabantia.com/uk/statement-data-breach

Werlinger, R., Hawkey, K., and Beznosov, K. 2009. "An Integrated View of Human, Organizational, and Technological Challenges of It Security Management," *Information Management & Computer Security* (17:1), pp. 4-19.

Yan, J.J., Blackwell, A.F., Anderson, R.J., and Grant, A. 2004. "Password Memorability and Security: Empirical Results," *IEEE Security & privacy* (2:5), pp. 25-31.

Zhu, B., Fan, X., and Gong, G. 2014. "Loxin–a Solution to Password-Less Universal Login," *2014 IEEE INFOCOM Workshop on Security and Privacy in Big Data (BigSecurity 2014)*, Toronto, Canada, pp. 494-499.

Zurko, M.E., and Simon, R.T. 1996. "User-Centered Security," *Proceedings of the 1996 workshop on New security paradigms*: ACM, pp. 27-33.