TRUSTVOUCHER: AUTOMATING TRUST IN WEBSITES

by

KEVIN DEAN

B.S., Kansas State University, 2012

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing And Information Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

Approved by:

Major Professor
Eugene Vasserman

Copyright

Abstract

Since the early 2000s, Internet users have continuously fallen prey to the perils of identity theft and malware . A number of tools have been proposed and implemented to foster trust towards deserving websites and alert users of undeserving websites, including P3P and trust seals. Each of these has fallen short, with studies showing that users simply do not use them. TrustVoucher is a prototype system o forge bonds of trust between users and websites by automatically determining if the website is backed by a trusted third party. Inspiration is taken from the real life way of trusting businesses, in which one aggregates recommendations by friends. TrustVoucher protects users who are attentive to its messages by informing them of sites who have put forth the effort to be endorsed by a trusted third party. An experimental study was performed on the effectiveness of the chosen interface for doing this, and determined that users did not consistently trust the recommendations of TrustVoucher, so future work will explore options for gathering the trust of users to distribute among websites.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Dr. Eugene Vasserman, for all of his guidance in creating and testing my ideas. I would also like to thank Dr. Gary Brase for his assistance in the experimental aspects of this thesis.

Also, thank you to my family, friends, professors, and wife for their support in completing this work and reaching one of my life goals.

Chapter 1

Introduction

User privacy on a platform like the Internet is an inherently trust-based problem. Despite the use of privacy-enhancing technologies like SSL and Tor, the privacy of users is still left at the mercy of the websites they visit. No matter how many technological barriers are placed between the browser and a website, that website is always left to do what it will with the data it is given during browsing.

A soft option that many popular websites use is the privacy policy, a human readable document, usually easy to find, that outlines what data the company will collect and how it will use it. In addition, they often include sections on how the site utilizes cookies, small packets of data that websites can store on a users computer to read back later (they are often used for the purpose of tracking browsing habits, a controversial practice).[2][3] A number of laws in various countries require certain kinds of websites to publish a privacy policy detailing specific information, such as websites that collect data on children under 13 in the United States.[4] Textual privacy policies are generally trustworthy, if one has heard of the company, but users do not read them. Privacy policies are lengthy, difficult to comprehend, and sometimes legally binding documents,[4][5][6] so while they communicate intent effectively, most users never even glance at them.

A few technological solutions to privacy problems have been proposed, such as P3P

(Platform for Privacy Preferences).[7] P3P was a protocol that tried to mitigate privacy issues by having website owners publish machine readable versions of their privacy policies, in particular the portions on how they use cookies. P3P agents are able to interpret these and inform users when sites have policies that do not match their preferences. Work on P3P was suspended after its final version in 2006,[7] for a variety of reasons. One major reason was a lack of use. One study found that in 2003, a year after P3P's publication, only 23% of the top 500 websites implemented it.[8] Additionally, few users seemed to actually customize the settings of P3P agents, leaving them on defaults. This shows a need for tools that do not require initial user input to function securely. Fundamentally, P3P did not solve the issue of trust. It relied on the premise that published policies would match up with actual practices, by assuming the existence of an assuring party.[9] This suggests that a solution to privacy problems may lie in the initial trust that a user gives to websites.

Another option is trust seals, which have been supplied by companies like TRUSTe since the late nineties.[10] For a large part of the 2000s, trust seals could be found on many major business websites (particularly storefronts) as an assurance of the websites' privacy or security. Having a trust seal from a known provider shows the user that the third party vouches for some property of the website, though this property depends on the provider. Some seals imply that the website has and follows a public privacy policy. Others are an indication of website security, stating that regular scans are performed on site infrastructure. While they do not provide universal information about websites, trust seals do provide an assurance that the third party trusts them enough to endorse them.

Unfortunately, trust seals as they are have failed to truly catch on. In a study by Kirlappos et al.[11], it was found that more than a third of users did not notice trust seals at all, and that only 20% of users noticed all trust seals in the experiment. In addition, none of the interviewed users could correctly identify the meaning of a trust seal. Trust seals can be still be found on some major websites today, but are most often relegated to the privacy policy page where most users do not see them. Unlike P3P, however, the major shortcomings of trust seals are mostly in how they are used, so they may still be useful as a means of establishing more solid user-website trust. If the seals could be made easily verifiable, then users would be able to get more immediate value from them.

The system described in this thesis, TrustVoucher, attempts to correct the problems of trust seals by moving them out of web pages and into the browser itself. TrustVoucher uses a series of steps to detect, recognize, and verify the claims put forth by websites that they are trustworthy. Websites must be trusted by third party called the notary, similar to P3P's concept of an assuring party. If claims are valid, the user is notified of the trustworthiness of the website, and if it is not then they are warned of the possible deception.

As described in chapter 4, TrustVoucher is compatible with the current state of trust seals, as well as a future state in which embedded images are no longer used. This backwards compatibility ensures a smooth transition into an Internet where websites can gain the trust of the user, and malicious websites are unable to gather undeserved trust. In this thesis, the term "trust" will refer to the willingness to provide personal or identifying information to a third party.

An expanded description of trust seals and alternative privacy mechanisms follows in chapter 2. Chapter 3 is a high level description of the protocol used in TrustVoucher. Chapter 4 describes usability issues and the implementation of TrustVoucher. Chapter 5 discusses the experimental method and results for a user study on the implementation from chapter 4. Conclusions and future work are found in chapter 6.

Chapter 2

Background

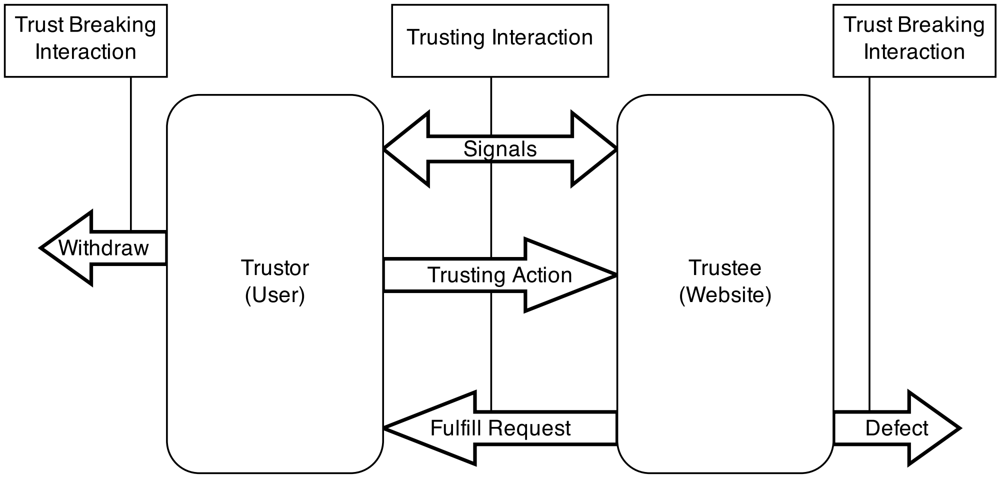


Figure 2.1: *The trustor-trustee interaction, adapted from Riegelsberger et al.*[1] *Using the browser example, the trustor signals the trustee of the desire to interact with an HTTP request, and the trustee replies with an HTTP response. The trustor may decide to withdraw or to make the trusting action, and evaluates the risk involved. The trustee then decides if they will respond with the desired fulfillment, or defect and break the trust chain.*

2.1 Trust Model

The basic model of trust interactions on the Internet is illustrated in Figure 2.1, adapted from Riegelsberger et al.[1] It shows the possible ways that an online interaction may play out.

First, the user (trustor) and website (trustee) send signals (HTTP requests and responses) to each other. The user evaluates these signals to determine how much they trust the website, and decides to withdraw or continue on. The website may similarly decide to provide the service they have offered, or be malicious and withdraw with the users information.

The mechanisms discussed in this chapter have been developed to ensure that both parties use trusting interactions, or to enable quicker and more reliable access to trust breaking interactions when they are needed. These solutions operate either inside or outside of the interaction boundaries.

Solutions that operate within the boundaries are online protection, and can only utilize information made available through the signal transfer. Usually online mechanisms are automated within the browser, and are transparent to the user. Do-not-track and P3P, discussed in sections 2.4 and 2.3, both provide online privacy protection. Online protection has the unfortunate side effect that it usually must assume honesty of websites.

Some mechanisms operate entirely outside the boundaries of the trust interaction, and provide offline protection. Offline mechanisms involve a third party authority of some kind that provides information on what websites it considers trustworthy or untrustworthy. This can be done as a whitelist, such as those maintained by trust seal operators in section 2.2, or a blacklist of sites not to communicate with, as in the Safe Browsing malware blocking system in the Google Chrome web browser.[12] While this works well for malware protection, new websites are created too often for an offline trust mechanism to be effective.

TrustVoucher uses a hybrid option to facilitate trusting interactions. In hybrid protection mechanisms, information about the trustworthiness of websites is gathered offline, and accessed during the trust interaction to evaluate whether the user is safe.

2.2 Trust Seals

A number of third-party companies provide trust seal services with varying kinds of guarantees. In Holst's survey of users,[13] the most trusted seal providers were identified as the Norton Secured seal, the McAfee Secure seal, the TRUSTe certification, and the BBB (Better Business Bureau) Accredited Business seal. All of these manifest as images in a web page, which the user may click to verify. The first two provide security seals, and the other two provide privacy seals.

The Norton Secured seal is the current incarnation of the former VeriSign seal. In 2010, VeriSign sold off its entire security unit to Symantec, including its SSL business and security certification services.[14] While these seals are still labeled with "Powered By VeriSign", Symantec plans to eventually switch them to read "Powered By Symantec" instead. According to Symantec's website, certified sites are scanned daily for malware, are verified to use high-grade encryption, and are entered into the "seal in search" program provided by some browsers and search engines. In the Holst survey, 35.6% of users (a large plurality) indicated they trusted Norton the most, likely thanks to its well-known antivirus software.

The McAfee Secure seal operates similarly to the Norton seal, but over a wider range of applications. McAfee, another well-known antivirus vendor, previously operated the HackerSafe seal. In 2008, the HackerSafe brand came under heavy scrutiny after it was revealed that dozens of certified sites contained significant security bugs.[15] As a result, the seal program was rebranded to McAfee Secure. McAfee's seal requires that the website be PCI compliant as a precaution.[16] The company claims that its seal increases confidence in 73% of shoppers.[17]

The Better Business Bureau (BBB) operates a seal program to complement their real-world recommendation program, intended specifically for online commerce websites.[18] The BBB already has an established accreditation business in the world of classical commerce. Customers use the BBB seal to ensure that merchants who they patronize conform to good business practices. The BBB web seal attempts to extend this to e-commerce by utilizing its brand recognition to gain clients and to help its clients gain customers. Despite criticisms of their ratings system[19], the BBB seal is much more focused on actual business practices than the other seals, rather than security or raw privacy.

TRUSTe provides privacy seals, rather than the security seals that Symantec and McAfee offer. Rather than vouching for policy content, however (as P3P does, described in the next subsection), TRUSTe offers a statement of policy consistency that users may choose to believe.[10] The seal that websites embed assures a user that TRUSTe has viewed and verified that the privacy policy displayed by the website discloses all relevant privacy information, and that the website's practices are consistent with this. No guarantees are made about what will be done with the information (except that the site follows TRUSTe's basic requirements), so users cannot use this as an indication that the site conforms to their privacy preferences, as with P3P. On the other hand, the seal does give a measure of authenticity, as a website with a TRUSTe seal fully discloses their information usage.

Current trust seals have not had the impact they aim for. Few users pay attention to them now, or realize the implications of fake seals.[11] To be effective, seals will need to become both more visible and more trustworthy. TrustVoucher will attempt to solve both of these issues by taking over presentation of seals from site operators.

2.3 Platform for Privacy Preferences

In 2002 an alternative to trust seals was ratified as a standard by the World Wide Web Consortium (W3C). The Platform for Privacy Preferences (P3P), was intended to let websites declare their privacy intentions in a machine-readable format. The hope was that if browsers could automatically determine what privacy policy a website followed before the user landed, then it could decide if the site was safe based on preferences set by the user.

P3P was based on machine-readable policies, encoded in XML format and conforming to a predefined schema, found in the 1.0 specification.[7] Policies consist of a series of statements, each applying to an aspect of the website's functionality. Each statement is made up by a series of components, chosen from the 11 defined. They define information such as contact information for the site owner, types of data collected by the site, how data is used, and with whom data is shared.[11] More information on these can be found in the specification or in Kirlappos et al.[11]

An extra feature of P3P is the compact policy. Compact policies let websites specify information about cookie usage in the form of short tokens instead of full statements. These policies have seen use in helping reduce tracking, and cookies that do not have an associated compact policy are blocked by default in some browsers.[11]

For a few years, P3P had momentum and enjoyed integrated user agents in both Netscape Navigator and Internet Explorer (the two biggest browsers at the time), as well support from AT&T with their Privacy Bird plugin. As usage increased, the flaws in P3P gradually became more apparent. One of the biggest problems, mentioned before, was the low adoption rate. In addition, P3P was somewhat limited. Stufflebeam et al.[20] complained about its inability to express "effective-on" policies, although it could express expiration. They also noted that P3P was unable to specify its entire privacy policy as an organization, and that it could only apply to a subset of privacy promises.

P3P never gained full acceptance, and was abandoned after its final 1.1 version in late 2006. Mozilla's browser initially included support, but it was dropped in late 2003.[21] Since then, few browsers have added support. Internet Explorer still includes it, but usage appears to have been mostly relegated to the affect compact policies on cookies.[22] Compact policies will likely be short-lived as well, with the rise of more lightweight options like do-not-track.[9]

2.4 Do Not Track

The "do-not-track" initiative, or DNT, is a simpler attempt at filtering websites by privacy practices. DNT is an option now provided by all major browsers to send headers informing websites not to track their browsing habits via data mining or cookies.[23] It has proven a mild success, with a number of social media websites, advertising networks, tag management platforms, and analytics services pledging to honor it.

Unfortunately, DNT suffers from the same honor-system drawbacks as P3P. The proposal for DNT does, however, claim that most major tracking entities on the Internet are large “law-abiding commercial enterprises”, who are more incentivized to honor the request not to be tracked than a dishonest web entity.[24] If most of the important players are honest, then perhaps an honor system could work.

Currently DNT still has a relatively small list of cooperating entities. In 2012, a number of major advertising players such as Google pledged to support DNT, but are still left off of the list of websites who honor it.[25] Since most websites have failed to even participate in the system, DNT has failed to make its mark.

2.5 Privacy Icons

In 2011 Mozilla proposed an alternative method of facilitating trust that functions as a lighter version of P3P, referred to as the Privacy Icons.[26] The icons acts as a form of compressed policy, expressing a website’s intent in the form of simple pictograms. Websites could lie, but are expected to be truthful about the policies.

Each of the four icons corresponds to a common privacy concern. The retention period defines how long a website will keep data, with options for 1, 3, 6, or 18 months, as well as an indefinite option. Third-party use is a binary option. A site may either state that they only share data in obvious ways or that they reuse it for certain purposes (such as statistical data). Ad networks states whether or not a site will share information with advertising networks. The final option, law enforcement, states that the site either cooperates with governments only when legally required, or that they will provide data to governments without being required.

While the icons each provide websites with low-privacy options, each one requires the website to be transparent. Sites may provide information to governments without being required, for example, but they must publish the process they will follow when doing such. The privacy icons do not allow any unpublished usage of personal data, although this is not technically enforceable.

Mozilla's solution is meant to be "bolt-on". As such, it limits what users can be told about policies (as opposed to P3P's abundance of information). The largest downside, as with other trust markers discussed, is the dependence on honesty. The privacy icons provide useful information, but they do not prevent unwarranted trust. In contrast, TrustVoucher will attempt to solve both problems.

2.6 Transport Layer Security

Transport Layer Security (TLS), and its predecessor Secure Sockets Layer (SSL), are sometimes suggested as a trust indicator, despite their focus on confidentialy and authentication. The TLS protocol is a very complex system, notorious for its difficulty to implement. For further reading and technical details, see the analysis by Wagner and Schneier[27] and the RFC describing the protocol.[28]

TLS/SSL is manifested in the HTTPS protocol, a secure version of the more common HTTP (Hypertext Transfer Protocol), and is triggered by placing the "https://" string before web addresses. It provides end-to-end encryption between users and websites, allowing mitigation of man-in-the-middle (MitM) attacks. A MitM attack is a form of impersonation, where a malicious entity on the web intercepts a users traffic and presents itself as the target website, gaining any information the user might provide to the intended website. The exact way this is done is described in the referenced papers. TLS/SSL only provides confidentiality of data, however. Once the data arrives at the destination server, the receiver may do whatever they wish with it.

A relevant detail is the concept of a Certificate Authority (CA), from the X.509 standard that modern TLS runs on. A CA is a web entity that holds a set of encryption keys trusted by the majority of popular web browsers. Technically a CA may not be trusted, but for our purposes it is assumed that the term CA implies inclusion in browsers. The CA provides certificates (the data structure used to start end-to-end encryption) to websites who do business with it. Those websites may then serve the certificate to users, both to start the encryption process and, as a side-effect, prove their identity.

The "lock icon", as it is often referred to as, is sometimes considered to be a way that a website may gain trust from users. The idea that users see the padlock and believe this makes them more secure is tempting, but studies have not supported it. In 2005 Jensen et al.[29] found that the lock icon contributed the least to user perceptions of trust from the features they tested (notably, trust seals topped the list). Jakobsson et al.[30] found that two copies of a web page, one with the lock icon and one without, received nearly identical ratings.

One recent innovation on the TLS/SSL frontier is the creation of HTTP Strict Transport Security (HSTS). HSTS lets websites request that the browser always load it with HTTPS, forcing encryption on every communication.[31] It has not yet been studied whether the use of HSTS has had an effect on TLS/SSL's use as a trust mark. Many of today's largest websites, such as Google and Twitter, now send HSTS requests,[32] so it is possible that users have started noticing it more due to saturation.

The CA model does bear some similarities to the TrustVoucher system described in this paper. Certificate authorities are analogous to notaries, in that users are provided a preconstructed list of entities that they are expected to trust. While TLS has not penetrated the area of user trust very far, it has revolutionized Internet safety. TrustVoucher is designed to bridge the gap and let users determine who they should be sharing data with in the first place.

Chapter 3

System Description

TrustVoucher is defined in two versions, referred to as A and B. The primary version, TrustVoucherA, is meant to be quicker, less intrusive, and does not require extra HTTP requests during page loading. The additional TrustVoucherB exists to facilitate the transition forward, by using the existing seal system (described in 2.2) for bootstrapping. TrustVoucherA is a more ideal solution once its mechanisms have been adopted by notaries and websites.

Both versions follow the same four-step process: detection, recognition, verification, notification. The differences arise in detection and recognition, as described in the following sections. Notification is the act of telling the user whether verification succeeded or failed, and is described further in chapter 4.

3.1 Terminology

The system description will use some new terms, and some specific technical terms.

- A claim is a piece of information presented by a website indicating that is it trusted by a third party.

- A notary is a trusted third party who certifies websites as trusted. These are expected to evolve from modern trust seal providers like TRUSTe and BBB.
- A client is the website being contacted, who puts forth a claim.
- Hostnames are the dot-separated addrees used to access a website. Ex: `http://www.google.com` has the hostname www.google.com
- Headers are key/value pairs of data sent by websites and read by browsers. Custom headers are preceded by an 'x-'.
- A hash is a uniform length piece of data that can be computed from any input. Long strings of text can be hashed to create a small, but quasi-unique identifier. Small changes in input produce large changes in a hash, so if changes occur between hashings, it is easy to tell.
- HTML is a language used to specify the content and layout of web pages.

3.2 Adversary Model

As a security system, TrustVoucher requires a defined adversary model to evaluate its effectiveness at protecting users. The only adversary considered in TrustVoucher is untrustworthy websites. An untrusthworthy website is one that claims undeserved trusthworthiness. Websites that are malicious, but do not claim the trust of a notary, cannot be identified and are out of scope. Eavesdropping adversaries are also out of scope, and are expected to be mitigated by the use of TLS/SSL. Notaries are assumed to be trustworthy if, and only if they appear within the list of trusted notaries distributed with TrustVoucher implementations. A notary that has not been vetted should not appear on this list, and websites claiming their trust will not validate. Notaries that have been comprimised are not considered here.

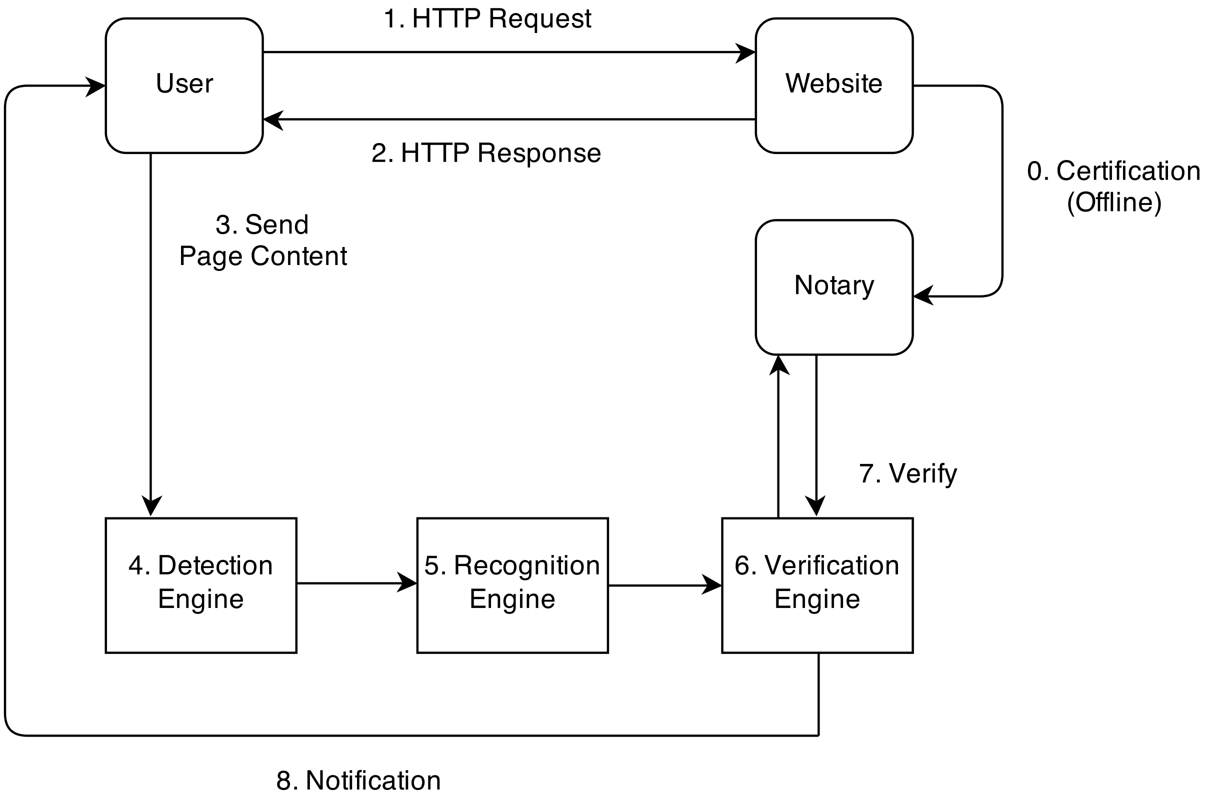


Figure 3.1: *The claim process. The user must perform a full page request, and TrustVoucher will process any claims made by the website in its detection, recognition, and verification engine. Communication with the notary refers to looking the website up in the notary's trust list, and the certification action is the behind-the-scenes interaction between notary and website.*

3.3 TrustVoucherA

3.3.1 Claims

In TrustVoucher, the idea of embedding seals into the HTML of a webpage is replaced by the concept of claims. A claim is a token provided by a website at load time, indicating that it is trusted by a notary. In TrustVoucherA, claims exist as headers; specifically, a custom x-trusted-by HTTP header, the contents of which inform the browser of what actions to take during verification.

A claim header is formed as the 2-tuple (D, S). D is the hostname of the notary, such as truste.com (the hostnames used are always fully qualified). S is a digital signature of the website's hostname, provided to the website out of band by the notary.

3.3.2 Trust Lists

To facilitate claim verification, TrustVoucher requires a trust list from each notary that is considered trusted. Each notary will have an entry in the notary list, with its corresponding trust list. Participating browsers will come with a default notary list, including well-known ones that exist today (see chapter 2). Other notaries may petition for inclusion on this list as well, and users may modify it through the settings to either add or remove notaries. If a notary is found to be misbehaving and giving out its trust unwarranted, then its status may be frozen and TrustVoucher will fail to verify associated claims.

A notary list is a set of hostnames that identifies the notaries. The trust list associated with a notary is a set of hostnames. A claim is considered valid if the hostname of the client is found in the trust list of the claimed notary. If one assumes that each website on a trust list will put forward the claim, then we can state that every website in each trust list is trustworthy.

Each trust list is paired with the public SSL certificate of the notary, and is considered valid for an arbitrary period of 24 hours. After 24 hours, the list is considered stale and a new one is requested from the notary. Since TrustVoucher is a live system, notaries that cannot be contacted for a new trust list will be temporarily removed from the trusted list in case of malicious activity. This should incentivize companies that run notaries to have maximum uptime, but may also lead to user annoyances during unforeseen outages. It is left as future work to determine if keeping old trust lists for a grace period is a better action.

To request a new list, the browser will send a request to the notary with the timestamp and a hash of the current list. If the notary determines the list is not current or that the hash is not consistent with what is expected, then a delta-encoding-based transfer will initiate to synchronize the local copy. This is done every 24 hours when the trust list becomes stale. If a seal expires, then the notary will not include it in resynchronizations. Notaries may also opt to append a "scheduled refresh" to their responses, which is a timestamp of when the browser should next try to refresh the list. This lets notaries deal with any traffic issues, and prevent themselves from experiencing denial of service from too many simultaneous requests. This is similar to TLS/SSL's revocation checking, which has its share of criticism.[33] Unfortunately, trust lists require more regular updates. Certificates only need revocation if the cryptographic properties are compromised, it will not stop authenticating the website for small reasons. Trust may be revoked for many reasons, though, and websites may need their trust revoked on the fly (for instance if the business decides to change their terms of service without verifying it with the notary).

3.3.3 Verifying Claims

Claims are verified using the trust list entry of the notary being claimed by a client site. Once a claim is parsed into the tuple (D, S), TrustVoucher will search the list of trusted notaries for D. Once D's trust list has been found, the list is searched for the hostname of the client site. If the site is found, then TrustVoucher uses the SSL certificate of the notary to verify the digital signature S. This serves primarily as a sanity check to prevent malicious modification of local trust lists to include an extra website (by appending an entry to a local trust list). At this point, the claim has been verified, and the user is notified.

At this time, TrustVoucher includes no facilities for checking what sort of relationship the notary and website share. Specific auditing practices of the notary, such as privacy policy validation and security scanning, may be identified and dealt with in future versions of TrustVoucher.

3.3.4 States

TrustVoucher has a limited number of states it end in after the claim process is complete. These are ’trusted client’, ’false claim’, and ’no claim’. Trusted clients are the desired state. This includes any site that makes a claim that is able to be verified. This will lead to an unobtrusive notification for users.

If at any point one of the verification steps fails, the system will transition to a warning state. The user will be notified that the site claims to be trusted but is not. This distinction from sites that do not claim trust at all is important in cases where trust has been revoked. If, for example, a merchant site has its trusted status revoked by its notary for breach of privacy policy, then users who normally would blindly trust the website should informed of the new untrusted status.

Websites which make no claims fall into a special category. Since TrustVoucher’s goal is to facilitate valid trust with as many websites as possible, users should be aware of a site which does not even claim to be trustworthy. An example may be a malicious merchant site which does not make any claims in the hope that a user defaults to giving blind trust. To counter this, TrustVoucher could trigger a warning on any website making no claims.

The opposing position is that some sites do not need to be trusted in any way, and so should not lead to any notifications at all. While this circle of the Internet is dwindling with the constant evolution of the web, sites that do not gather information from users (besides IP addresses and headers) clearly do not need vouching. This would include websites that do not have any user account system, and so could be generalized to websites with no password fields. However, some sites may collect data without user accounts, so this heuristic may require further research.

3.4 TrustVoucherB

3.4.1 Backward Compatibility

The second version of TrustVoucher is backward compatible for immediate usage (to allow for incremental deployment of TrustVoucherA). The reasons for this are threefold. First, this eases the transition for users, who will be unaffected as websites shift from embedded seals to claim tokens. Second, it gives website owners a wider window to upgrade their systems. At first, any system with embedded seals will still function, but operators will be incentivized to upgrade to take advantage of higher user trust gains, less clutter to include on their sites, and the possibility of notaries phasing out image seals. In this way, TrustVoucher becomes immediately useful without breaking user experience in any way. Finally, having a backward compatible system will give a window during which the notaries themselves can upgrade their systems. During the window the notary can generate their certificates, trust lists, and tokens, and distribute them in a convenient manner to the trusted parties, before making the image seals obsolete.

3.4.2 Detecting Seals

The first step of parsing a trust seal claim is to detect the existence of the claim. Rather than special headers, TrustVoucherB accomplishes this by scanning web pages for the image seals described in 2.2. This process is described further in chapter 4. If a website contains a known trust seal, it is considered to claim the notary that the seal belongs to.

3.4.3 Verification

Once a seal is claimed by a website, the domain is passed on to the verifier. In the modern embedded seal system, users can verify a seal by clicking the image. They are taken to the notary's website and presented with an abundance of information. The process is demon-

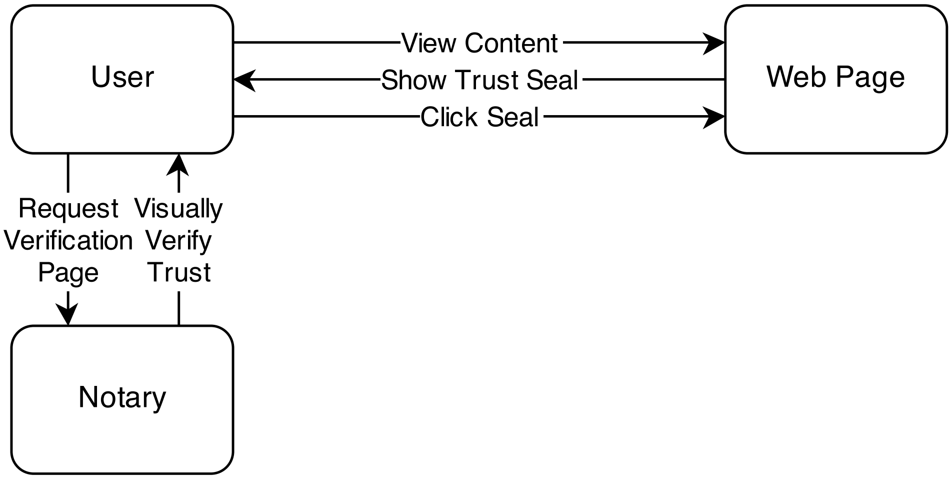


Figure 3.2: *A user's verification process. These steps must be followed for a user to reliably determine if a website is trustworthy under the trust seal model. Most users will not put in the effort to complete it, so automating the process will increase the trust gains of websites.*

strated in figure 3.2. TrustVoucherB simulates this process as an asynchronous request in the background. The response is scanned for an indication of success or failure, and the result is passed on to the notification engine implemented as part of chapter 4.

Any errors encountered during the verification process are considered failures, and are sent as such to the notification engine. This means that any accidents, like the notary website going down, will register as failures and cause a notification to the user. This could potentially cause false negatives on error reporting, but it is necessary to defend against fake seals. If unknown notaries are allowed or ignored, then users may give undeserved trust to the site.

Chapter 4

Implementation

4.1 Usability

The primary goal of TrustVoucher is to facilitate trust. Chapter 3 described the system architecture, but ignored user interactions. On the user's end, trust will be facilitated by providing information on the existence and validity of notaries for the websites they visit. Facilitating that trust follows directly from convincing the user to believe the recommendations TrustVoucher gives.

In the expanded trust model described by Hoffman et al, six factors are listed for gaining a users trust: security, usability, privacy, availability, audit and verification mechanisms, and user expectation.[34] The security and privacy factors are dealt with by the claims process. Audit mechanisms are left to notaries to perform, and verification mechanisms have been prescribed in chapter 3. User expectation (or product reputation) is discussed briefly in chapter 5. Availability is a problem left to notary services, and is out of scope of this paper. This leaves usability as the most pressing problem for ensuring that users actually take the advice that TrustVoucher provides.

Achieving a usable interface is not a trivial problem, and is still the subject of ongoing research. The approach used in the first version of TrustVoucher will be as simple as possible.

4.2 Notifications

The most immediate usability issue is how often the system interacts with the user in an unsolicited way. Generally a system that constantly bugs users is one that they will start to ignore (the “crying wolf” problem[35]), so a balance needs to be struck. The three options for TrustVoucher are to notify users when a website claiming trust fails to validate, when a website makes no claims at all, or when a website makes a claim that validates. The first option is necessary (users need to know if false trust is being extended), but not sufficient. Adding either of the other two options is sufficient, since both supply enough information for a user to evaluate their situation (as long as errors are reported with the first option). Using both would lead to some kind of notification for every website - far too many for a usable system.

Notifying users when no claim is made, the negative option, would give fewer notifications in an ideal world where most websites cooperate with a notary. Only those sites that are untrustworthy would be noteworthy. The positive option, on the other hand, would immediately reduce notifications, since few websites currently use a notary. As adoption increases, positive notifications would increase as well. The two ways of notifying are then technically equal.

According to the goals of TrustVoucher, however, positive notification is superior. The premise is to foster as much trust as possible, and positive notifications should theoretically increase a user’s trust each time they are displayed. If no notifications are given, then there is no increased trust. The possible downside is sensory overload, where a user begins to see so many notifications that they start ignoring the system entirely and extending trust to every website. Future work will test the sensory overload problem and decide whether negative notification might resolve it.

To reduce the number of notifications, TrustVoucher will check whether the domain in question has been visited before. Previously verified domains will be ignored, unless its claim becomes invalid. If a previously valid website becomes invalid, then the user will be notified.

4.3 Extension Architecture

The prototype implementation of TrustVoucher is an extension to the Mozilla Firefox web browser. The entire system is a JavaScript file, which is attached to every new web page using the DOMContentLoaded event provided by Firefox. The DOM (Document Object Model) is a JavaScript representation of a web page's structure. The rest of this section will assume familiarity with the DOM and basic JavaScript. For more information on these, see Mozilla's tutorials on the subject.[36][37]

The extension is split into four main parts: detection, recognition, verification, and notification. Detection is relevant only to the backward compatibility of TrustVoucherB, and the internal function of recognition changes depending on which version of TrustVoucher is triggered. Verification also depends on which version is being run. Notification is the same in both versions.

The decision as to which version of TrustVoucher is run is dependent on the presence of an x-trusted-by header. If this is absent, TrustVoucherB is used as the fallback mechanism.

4.3.1 Detection

Detection for TrustVoucherB is performed using two main components, MutationObservers and intervals, running in parallel. The interval is started when the web page is first loaded, set by default to run every five seconds (this value is arbitrary at the current time; more research is required into what makes an effective and efficient interval period). Every iteration of the interval, the web page is scanned in full for any images. Each image is checked against a list of sources that have already been sent through recognition, and is discarded if it is found, with the assumption that it is known not to be a seal. One weakness of this approach is that a malicious website could wait for TrustVoucher to scan its image, change the content served on its server, then force the web page to reload the image. TrustVoucher will see it as the same image, as the source still matches, but the user will see something else, possibly a false seal. If an image was not in the list of sources, then it is passed on to the recognition engine.

Unfortunately, running a full scan of the web page, even at five second intervals, is very inefficient. To mitigate this, a MutationObserver is spawned alongside the interval. MutationObservers are a feature of the DOM4 specification that trigger a callback function when mutation events occur (a mutation is any modification to the DOM tree).[38] The observer is initialized to recognize any and all mutations to the entire web page. If a mutation occurs, then a shared boolean value is set to true. Whenever the interval callback fires, it first checks whether the shared boolean is true, and only continues with the scan if it is.

For TrustVoucherA, header analysis is used. On the load of any webpage, a cache of recently examined domains is checked. If the current domain is either not found, or was added to the list more than 60 seconds before (this number is arbitrary, and can be altered for security), then an XMLHttpRequest is created to the page being visited.[39] This provides TrustVoucher with an easy way to access all headers for the website. These headers are sent to the recognition engine.

If the page is successfully verified at any time, the interval and observer are both disabled for the website. The rationale is that once a website has been verified as being trusted by a known notary, any further changes will never negate that fact. The use of the detection components is only to prevent malicious websites from inserting false seals after the page has loaded.

4.3.2 Recognition

Visual Seals

As described in chapter 3, TrustVoucherB includes backward compatibility that identifies claims by recognizing existing trust marks on a web page. In the implementation of the Firefox extension, this translates to a computer vision problem, where the extension must recognize whether a given page contains any of the known seals.

To solve this, two solutions were explored. First, a JavaScript library called Resemble.js was found and added to the extension.[40] On a given page, every image was compared to the known seal images, and a value representing the difference between the two was returned. Unfortunately, the values returned by Resemble.js can vary widely due to slight differences such as size. In addition, it was found that heavy image processing on every image of a page was far too much load on a given browser. Any website with an abundance of images would slow to a crawl after a few seconds. The initial issues were resolved by keeping lists of what images on a page had been checked already and removing items from the list when they were modified by JavaScript, but the performance problems of full feature detection on every image remained.

The second solution, and that which is used in TrustVoucher, is the perceptual hash. A perceptual hash is a way of deconstructing an image to its core features, rather than comparing by pixel. Generally it involves some method of reducing the content of the image using some form of low-pass filtering, such that similar images will look the same. Numerous algorithms exist for this purpose, including pHash,[41] Average Hash,[42], and feature extraction processes like the Generalized Hough Transform.[43] TrustVoucher opts for the dHash algorithm, chosen for its low number of operations per image (since it is used so many times per page).[44] Theoretically, any perceptual hash could be used for this purpose, if attacks against dHash were found to be a problem. Perhaps in the future, a more academically supported algorithm could replace it.

The dHash algorithm starts by converting the two images being compared to grayscale,

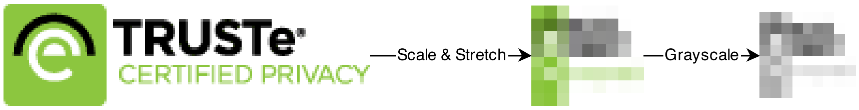


Figure 4.1: *The three steps of dHash performed on the TRUSTe logo. The final image is the hash value.*

mitigating the possibility of changing color values in a fake seal. Both images are then reduced to be 9x8 pixels (stretched), and antialiased. This has the effect of giving us a blurred version of the images, if they were to be ballooned back to their original sizes. At this point, a single pass is done over each image to create an 8x8 grid of boolean values. Each value is the result of comparing the pixel in that slot to the pixel on its right (since the grid is 8 wide, the last pixel is only compared to its left). Each row of the grid is compressed into a hexadecimal digit, and the digits are concatenated into a string, the perceptual hash. For the purposes of TrustVoucher, the hexadecimal hash is not used, but instead bitstrings of the comparisons are directly compared, with the resulting number of conflicts out of 64 possible being the comparison value. An experimental value of 10 was chosen as the threshold, with images with comparison values less than 10 (respective to a known seal) being considered matches.

Unfortunately, dHash by itself does not provide reliable results for the images being tested. It matches when it needs to, but provides many false positives. As an example, one of the common TRUSTe seals compares very well with a single blank pixel (used in many sites as a spacer), due to alpha transparency. To combat this, an original addition was made to the dHash algorithm. After the difference map has been computed, a “contrast map” is computed as well. The contrast map is a 216-bit list, a continuous list corresponding to an 8x9x3 RGB image. To construct the list, each pixel in the resized 8x9 image (the color version, not the grayscale version) is visited, and the max of the three color values is computed. For each of the three color values, if it is equal to the max, then a 1 is pushed to the list, otherwise a 0 is pushed. This has the effect of making a high-contrast image.

The contrast map is also not a perfect solution, but when paired with traditional dHash it provides reasonably sufficient protection against false positives, though further testing is required. TrustVoucher uses a threshold of 40 differences between contrast maps. An image must meet both thresholds against a known seal to be matched. If a match is found, it is sent further to the verification engine.

TrustVoucherB's recognition engine is still in the proof of concept stage, and requires further research to be secure in the wild. Currently, it will effectively match against known seals, but its major flaw is false negatives. It is very easy to fool the recognition engine into thinking a page has no claims, allowing a malicious web page to display seals without warnings from TrustVoucher. This problem is considered as future work.

Header Seals

The preferred version, TrustVoucherA, does not use visual seals. Instead claims are identified through header analysis, as described above. Once the recognition engine receives headers, it checks for any x-trusted-by headers. If the notary claimed by an x-trusted-by header does not exist in the list of known notaries, then a warning is sent to the notification engine. If it does exist, then the notary is sent to verification. If multiple x-trusted-by headers are found, then only the first one received is sent through the verification engine. If the first (or only) x-trusted-by header is not recognized, then an error notification is propagated to the user, informing them that the website claims trust from a non-existent notary.

4.3.3 Verification

Two options exist to verify a claim. First, we can use the trust list provided by the notary, described in chapter 3. If the website's hostname is found in the list, then the claim is valid. This is called offline verification, and is the preferred option. The second option, online verification, occurs if no trust list exists, indicating a trusted notary who has not yet conformed to the TrustVoucher system. Any notary in this set will have an associated matching URL, which the website's hostname can be sent to for online verification. The page returned by the notary will be scanned for search text that indicates success, determining the validity. TrustVoucherB will always use online verification. While online verification is no more secure than modern trust seals, it automates the process such that users have convenient access to trust information, and sets the stage for offline verification to replace it.

4.3.4 Notification

The final step for a webpage being parsed is to notify the user. In the case that no claim is ever recognized, this amounts to no action being taken. If a page is recognized and verified, the success notification described earlier is displayed. If a page is recognized but fails to verify, the error message is instead displayed.

Chapter 5

User Study

5.1 Experimental Method

To show how well the TrustVoucher system works as described, a user study was devised in the vein of Kirlappos et al.[11] The study was conceived as an online survey, administered through Qualtrics.[45] Participants were students from the General Psychology class at Kansas State University, a common source for experiment participants.

Participants were first asked a series of questions to divide them into the Westin Equivalence, described by Jensen et al.[29] This is similar to the Westin Privacy Segmentation used in Internet marketing research,[46] but is oriented more towards online privacy. The participants' responses allow for more detailed analysis as to who may be most affected by TrustVoucher.

After segmentation, participants were asked to rate a series of nine websites using a single visual interaction, accomplished with screenshots. They were unable to interact with the websites in any way, and were given a full screenshot taken in the Mozilla Firefox web browser. Since the pool of participants lacked a sufficient number of Firefox users, we instead restricted the study to Google Chrome, Internet Explorer, and Safari users, who should be equally unknowledgeable about the normal Firefox UI (this does not take into account users that prefer one of the indicated browsers but use Firefox often as well). Screenshots were of the entire page, created by compositing together the browser UI with a full page rendering. This allowed participants to see every feature of the website, as if they could scroll.

Each screenshot was pulled from one of three conditions. Websites identified as marked (by a trust seal) were exactly as a user would see when loading the website normally. Websites identified as stripped were shown with all trust marks removed. While the study was centered around the TRUSTe seal, other marks relating to security and trust were also removed to reduce noise. Removal was done in a fashion that preserved the structure of the web page, so that the screenshots still look natural. Websites with the final modification, augmented, were identical to stripped websites, except for the addition of the TrustVoucher indicator.

The screenshots were taken using an unaltered version of Mozilla Firefox 28. All standard interface features were left intact, such as TLS/SSL indicators. The only extension installed was a "save page as image" extension for creating the screenshots, which did not affect the visual appearance of the browser.

The set of websites shown to each participant contained an equal number of sites from each condition, and the distribution of site/modification pairs was made such that each one received approximately equal attention.

After seeing each screenshot, participants were asked to rate the website on a Likert scale of 1 (very untrustworthy) to 5 (very trustworthy). Additionally, they were asked to note what features of the website they used to come to their rating, whether they knew of the website previously, and what personal information they would be willing to provide the website if it were asked. The exact question can be found in appendix A.

5.2 Demographics

The online survey was completed successfully by 23 students. Each student was identified only by a random identifier associated with their answers. Nine students identified as "privacy pragmatists", and 14 identified as "privacy fundamentalists". No students answered in the category of "unconcerned". As per Jensen et al.,[29] pragmatists were those with four or more privacy-oriented answers, unconcerned were those with no privacy-oriented answers and at most one neutral, and pragmatists were those in neither group.

All students were four-year program undergraduates, with two sophomores, three juniors, one senior, and the remaining as freshmen. Five students were in the 21-23 age range, with the rest falling in the 18-20 range. One concern with this is that the data is only representative of users who generally grew up with access to the Internet and thus may have different perspectives than older users. 11 participants were female, and 12 were male. Five participants reported that they use Apple Safari as their browser of choice, six claimed Microsoft Internet Explorer, and 12 claimed Google Chrome.

5.3 Results and Analysis

The analysis of user ratings shows that the chosen interface for notifying users of trust marks is no more effective than existing trust seals, if not worse. When ratings from every site within each condition were combined together (table 5.1), it was found that users gave stripped sites a mean rating of 3.34, marked sites a 3.60, and augmented sites a 3.27. This implies that users found websites with visual trust markers slightly more trustworthy, but not to a statistically significant amount. As shown in table 5.2, none of the tests run on the data produced a $p \leq 0.05$, so it appears that the conditions had no notable effect on perceptions of trust. The median rating was similar, with marked sites having a median rating of 4 and both stripped and augmented sites having a median rating of only 3.

When ratings are split up by both condition and website, as in figure 5.1, we see the same

Condition	Mean	Std. Deviation	N
Augmented	3.3389	.89546	21
Marked	3.5992	.85535	21
Stripped	3.2690	.61609	21

Table 5.1: *Mean ratings for each condition across all ratings of that condition type. Different websites are collapsed together.*

Source	Type 3 Sum of Squares	Degrees Freedom	Mean Square	F	P	Partial Eta Sqr
Sphericity Assumed	1.272	2	0.636	.930	.403	.044
Greenhouse-Geisser	1.272	1.971	0.645	.930	.402	.044
Huynh-Feldt	1.272	2	0.636	.930	.403	.044
Lower-Bound	1.272	1	1.272	.930	.346	.044
Error(Sphericity Assumed)	27.350	40	.684			
Error(Greenhouse-Geisser)	27.350	39.419	.694			
Error(Huynh-Feldt)	27.350	40	.684			
Error(Lower-Bound)	27.350	20	1.368			

Table 5.2: *Assorted tests on the effects of conditions for each subject, and their error bounds.*

trend. Marked sites enjoy a consistently higher, but not significant, rating. Stripped websites suffer, as expected. The augmented ratings are completely unpredictable, sometimes high and sometimes low, but almost never rising above the marked ratings, and often falling below stripped ratings.

Table 5.4 shows the tests applied to the data on a per-site basis. As is expected, which site is being viewed by a participant has a large effect on the rating applied ($p \leq 0.01$). Table 5.3 shows the mean ratings for each site, with conditions collapsed together, and we can see that the culprit is sites 2 and 3, as they have significantly lower mean ratings than any of the other websites. This suggests that the design of websites has much more of an effect on perceieved trustworthiness than any sort of markings or indicators.

Figure 5.2 shows the number of participants who favored each site modification separated by the Westin Equivalence.[29] This was calculated by determining which condition had the highest average rating for each user, then counting how many users preferred each condition. Notably, pragmatists showed favor towards augmented websites featuring the TrustVoucher

Site	Mean	Std. Deviation	N
1	3.6818	.83873	22
2	2.6818	1.04136	22
3	2.9091	1.06499	22
4	3.7273	1.24142	22
5	3.4091	1.05375	22
6	3.5909	.95912	22
7	3.5455	1.14340	22
8	3.3182	1.08612	22
9	3.5000	.80178	22
Total	3.3737	1.06715	198

Table 5.3: *The mean rating of each website, with all conditions collapsed together.*

Source	Type III Sum of Squares	Degrees Freedom	Mean Square	F	P
Corrected Model	22.253	8	2.782	2.601	$p \leq 0.01$
Intercept	2253.657	1	2253.657	2107.671	$p \leq 0.01$
site	22.253	8	2.782	2.601	$p \leq 0.01$
Error	202.091	189	1.069		
Total	2478	198			
Corrected Total	224.343	197			

Table 5.4: *Assorted tests on the effects of websites across all participants, and their error bounds.*

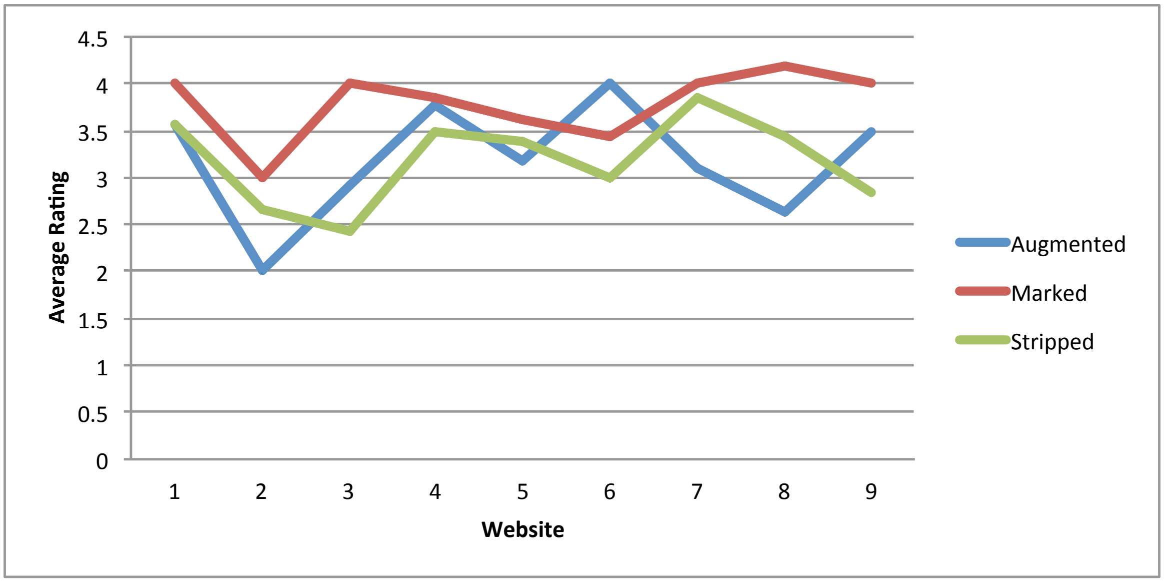


Figure 5.1: *The average rating of each website, separated by condition.*

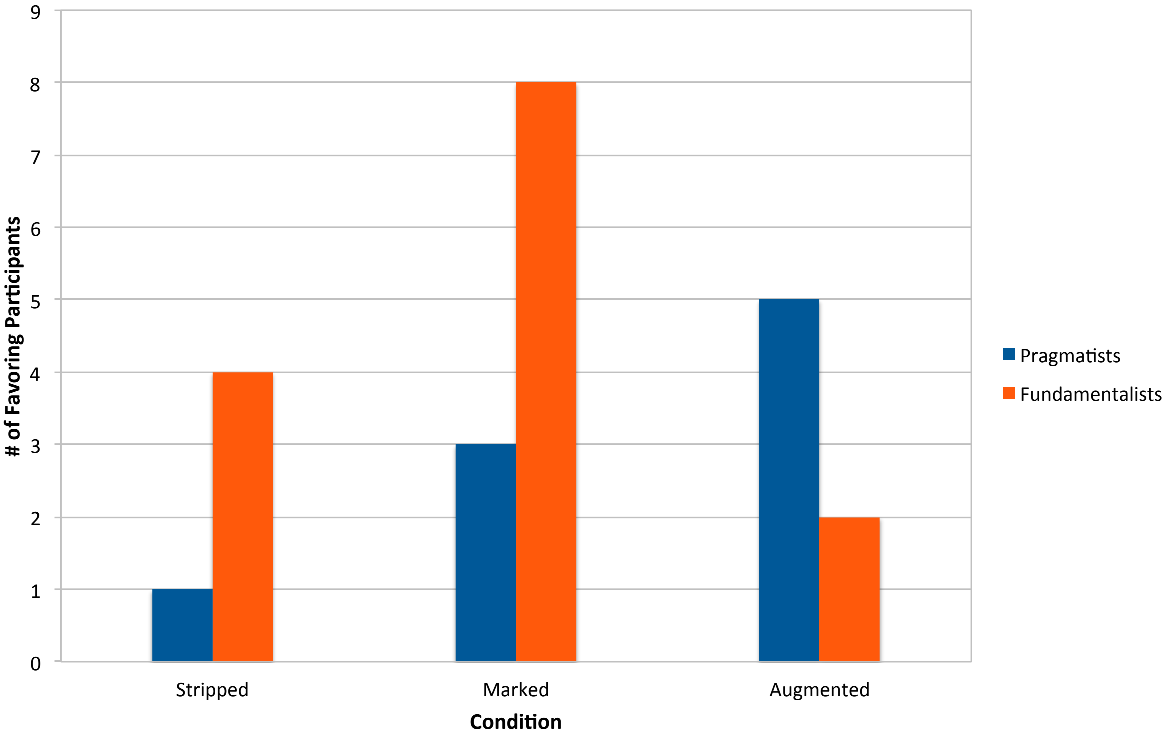


Figure 5.2: *Pragmatists vs. Fundamentalists - The y axis represents how many users gave the highest average rating to each condition. It is noteworthy that more pragamatists actually favored augmented websites than fundamentalists. These ratings were overwhelmed by the distrust of augmented sites by fundamentalists.*

indicator, while fundamentalists favored the marked websites. One explanation of this is that pragmatists are more open to the idea of new indicators, while fundamentalists, who are more protective of their privacy, may be more wary of possible attacks on their privacy.

When the privacy groups are split further, such that the average per site is applied only within privacy groups, this effect is more apparent, as demonstrated in figures 5.3 and 5.4. Within the fundamentalist group, 88% of augmented websites rated lower than marked websites, and 55% rated lower than stripped websites. Within the pragmatist group, however, only 55% of augmented sites rated lower than the marked version, and only 22% rated lower than stripped versions. This further supports that privacy pragmatists may be more open to the idea of the new trust indicator.

These results suggest that the prototype of TrustVoucher does not gather enough user attention. The next section will discuss user perceptions of the two trust markers presented to users, and attempt to justify the results in relation to them.

5.4 User Perceptions

5.4.1 TRUSTe Logo

At the end of the experiment, users were asked whether or not they had seen the TRUSTe logo before, and what they believe it indicates, followed by the same questions about the TrustVoucher indicator. These answers are more satisfying, and suggest that results may be more consistent with expectations with a little bit of publicity of the system, rather than the blind testing performed here.

Only two participants failed to notice either indicator. One participants noticed the TrustVoucher indicator but not the TRUSTe logo. Three participants noticed the TRUSTe logo but not the TrustVoucher indicator. These latter three participants said that the TRUSTe logo indicated some form of higher security or privacy, but two of them believed that the TrustVoucher indicator was not trustworthy, with one guessing that it was falsifying

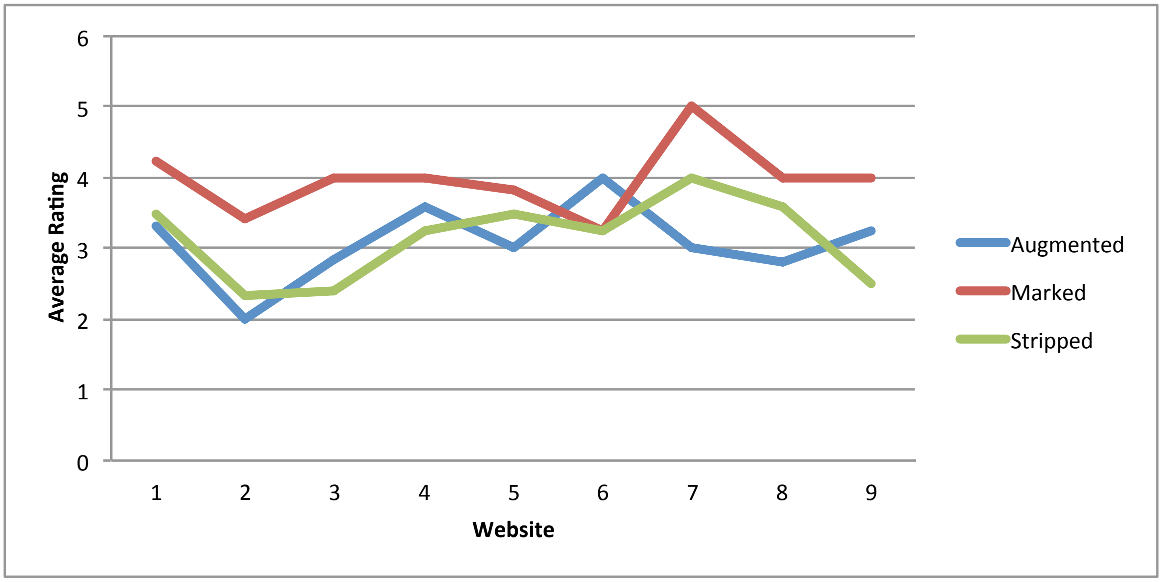


Figure 5.3: *The average rating of each website rated by a privacy fundamentalist, separated by condition.*

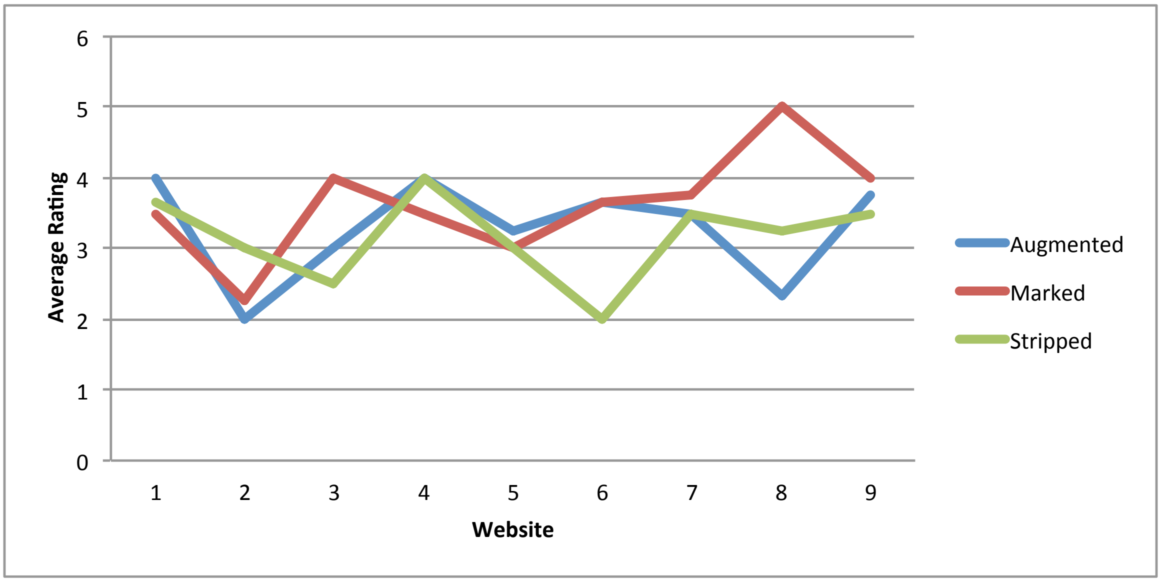


Figure 5.4: *The average rating of each website rated by a privacy pragmatist, separated by condition.*

the TRUSTe relationship. One participant chose not to answer these questions at all.

The remaining participants all answered that they noticed both. Eight participants correctly identified the TRUSTe logo as part of an online certification program. Ten of the participants guessed that the logo indicated heightened security, such as encryption, firewalls, or privacy guards. The remaining three simply believed that the logo "makes the site real" (real being the descriptor used by participants), demonstrating a belief that sites with the logo are more trustworthy. Some participants had completely incorrect assumptions about the meaning of seals, with one proposing that the seal vouches for the websites loan reliability.

5.4.2 TrustVoucher Indicator

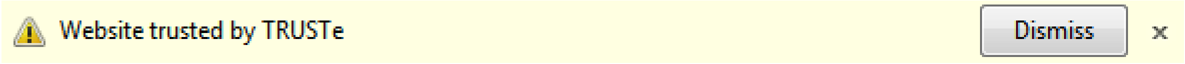

Website trusted by TRUSTe

Figure 5.5: *The TrustVoucher indicator.*

Ten participants guessed that the TrustVoucher indicator, pictured in figure 5.5, implies certification by an organization. Only four participants believed the indication of heightened security, but another four users chose the new implication that it was less trustworthy. This is an interesting issue, since the images provided by TrustVoucher are meant to be more trustworthy. The remaining three users were either unsure of its meaning or thought it meant the site is "real".

5.4.3 Improving Responses

A number of conclusions can be drawn about the proposed TrustVoucher interface from user perceptions. The most noticeable difference between the two marks shown to users seems to be the belief that TrustVoucher is fake. This is understandable, since users have not seen it before. The premise of the experiment was to see how users responded with no prior

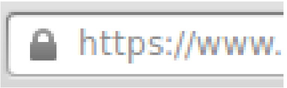


Figure 5.6: *The TLS/SSL icon in Mozilla Firefox.*

information about the interface, and only about half of participants were trusting based on the information provided. Since they had no cues, it is a good sign that any users responded well at all. If the system were publicized beforehand, and perhaps a short informational popup was shown when first installed in the browser, users might respond more positively.

The differences in responses between fundamentalists and pragmatists further supports this. Fundamentalist users, who have no idea what the TrustVoucher indicator is, should be expected to respond negatively to a new browser feature with no prior information about it. Pragmatists, who are less immediately concerned about privacy, were more receptive, since they have less reason to discriminate against strange features. If fundamentalists can be convinced that the indicator is a natural browser feature, then perhaps the ratings would start to converge.

Despite the lack of information about the indicator available to participants, it can be concluded that the indicator was not naturally trustworthy. The indicator that was used in the experiment is a standard notification box exposed by Firefox. One possibility is that participants saw it as too generic, and that it could have been created by the website. To fix this, a new indicator that is more integrated into the browser may be the solution. A good inspiration may be the markers used to show that a website is using TLS/SSL. Future work includes finding this more effective indicator.

Chapter 6

Conclusions and Future Work

The TrustVoucher implementation described is still a prototype and not suitable to actually protect online users, but has the potential to further improve the world of commerce and information exchange.

The most significant problem with TrustVoucher is in the notaries themselves. Since users cannot audit each notary personally, they will have to depend on the curator of trust lists to do this for them. A potential issue is rogue notaries, similar to the rogue certificate authorities that have recently been a problem in the world of TLS/SSL.[47] While the effects wouldn't be as wide-reaching, detecting that a notary has signed off for a malicious site may be more difficult than determining wrongly distributed certificates.

The most pressing issue of TrustVoucher's implementation is its internal security. The problem of false negatives mentioned before is a difficult issue to solve, and may be intractable as long as visual trust seals are still in wide use. If websites stop using them, then visual seals will become an oddity, and malicious websites will be unable to lure users with the familiar brands. In addition, TrustVoucher needs to be fully audited for security flaws in its implementation.

Another area of research is the user interface. The user study showed little value in the chosen method of notification. While a TrustVoucher user is more secure against malicious sites displaying seals, there is little gain if they do not use the information provided. A more engaging interface is thus desirable, and so more studies should be performed on what interface features users will pay attention to and trust.

A desirable, but tangential piece of future work is to expand TrustVoucher into more granular areas. In the form described, TrustVoucher will only tell users that the website is backed by a known third party notary. This is useful, but a more powerful function, previously attempted by P3P, is to make guarantees about privacy policies. One possible direction is to expand claims to include information about human-readable privacy policies, and a signature-based mechanism to verify that the policy is the one which was approved by the notary. Combining this with P3P's customization could provide a more reliable way for websites to advertise their privacy policies, as the problem of lying websites is reduced. This would, of course, carry the original P3P problem of users not customizing the system, but it is a step forward.

The main goal of TrustVoucher is to facilitate more trust between users and websites. As described, the mechanism for doing this is to automate the existing trust seal system into a claim/verify system, and to make the visual indications more noticeable to users. Hopefully, users will begin to pay attention to and use trust seals, so that websites can more easily bootstrap relationships with users.

Bibliography

[1] J. Riegelsberger, M. Angela Sasse, and J.D. McCarthy. The mechanics of trust: a framework for research and design. *International Journal of Human-Computer Studies*, 62(3):381–422, 2005.

[2] Grant Brunner. Mozilla drags its feet on blocking third-party tracking cookies. http://www.extremetech.com/internet/156188-mozilla-drags-its-feet-on-blocking-cookies-from-unvisited-websites, May 2013. Accessed: 2014-04-21.

[3] Robert L. Mitchell. Ad tracking: Is anything being done? http://www.computerworld.com.au/article/541921/ad_tracking_anything_being_done_/, Apr. 2014. Accessed: 2014-04-21.

[4] Federal Trade Commission. Childrens online privacy protection rule. 70, Apr. 2005.

[5] Understanding health information privacy. http://www.hhs.gov/ocr/privacy/hipaa/understanding/index.html. Accessed: 2014-04-21.

[6] Personal information protection and electronic documents act. http://laws-lois.justice.gc.ca/eng/acts/P-8.6/, Apr. 2011. Accessed: 2014-04-21.

[7] Platform for privacy preferences (P3P) project. http://www.w3.org/P3P/, Nov. 2007. Accessed: 2013-11-18.

[8] Lorrie Faith Cranor. P3P: Making privacy policies more useful. *Security & Privacy, IEEE 1.6*, pages 50–55, 2003.

[9] Cem Paya. Do-not-track and P3P: New privacy standard, weaker approach. http://randomoracle.wordpress.com/2013/04/13/do-not-track-and-p3p-new-privacy-standard-weaker-approach/, Apr. 2013. Accessed: 2013-11-20.

[10] Paola Benassi. TRUSTe: an online privacy seal program. *Communications of the ACM*, 42(2):56–59, Feb. 1999.

[11] Iacovos Kirlappos, M. Angela Sasse, and Nigel Harvey. Why trust seals dont work: A study of user perceptions and behavior. *Trust and Trustworthy Computing*, pages 308–324, 2012.

[12] Niels Provos. Safe browsing - protecting web users for 5 years and counting. http://googleonlinesecurity.blogspot.com/2012/06/safe-browsing-protecting-web-users-for.html, June 2012. Accessed: 2014-04-21.

[13] Christian Holst. Which site seal do people trust the most? (2013 survey results). http://baymard.com/blog/site-seal-trust, Jan. 2013. Accessed: 2013-11-20.

[14] The VeriSign trust seal is now the Norton Secured seal — Symantec. http://www.symantec.com/page.jsp?id=seal-transition. Accessed: 2013-11-21.

[15] Dan Goodin. McAfee ”Hacker Safe” cert sheds more cred - The Register. http://www.theregister.co.uk/2008/04/29/mcafee_hacker_safe_sites_vulnerable/, Apr. 2008. Accessed: 2013-10-28.

[16] Welcome to the PCI security standards council. https://www.pcisecuritystandards.org/. Accessed: 2013-11-21.

[17] McAfee Inc. McAfee SECURE: Web security and website protection product features. http://www.mcafee.com/us/mcafeesecure/products/mcafee-secure.html. Accessed: 2013-11-21.

[18] Council of Better Business Bureaus. BBB accredited business seal for the web. http://www.bbb.org/us/bbb-online-business/. Accessed: 2013-12-01.

[19] Jan Fletcher. BBB ratings under scrutiny again. http://blog.intuit.com/marketing/bbb-ratings-under-scrutiny-again/, Sept. 2011. Accessed: 2013-10-01.

[20] William H Stufflebeam, Annie I. Anton, and Neha Jain Qingfeng He. Specifying privacy policies with P3P and EPAL: lessons learned. *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, 2004.

[21] Bug 225287 - remove P3P from the default build. https://bugzilla.mozilla.org/show_bug.cgi?id=225287, Nov. 2003. Accessed: 2013-11-20.

[22] Internet Explorer 10 for Windows 7 privacy statement - Microsoft Windows. http://windows.microsoft.com/en-US/internet-explorer/ie10-win7-privacy-statement, Dec. 2012. Accessed: 2013-11-20.

[23] Jonathan Mayer and Arvind Narayanan. Do not track - implementations. http://donottrack.us/implementations, . Accessed: 2014-04-10.

[24] Jonathan Mayer and Arvind Narayanan. Do not track - universal web tracking opt out. http://tools.ietf.org/html/draft-mayer-do-not-track-00, . Accessed: 2014-04-10.

[25] Rainey Reitman. White House, Google, and other advertising companies commit to supporting do not track. https://www.eff.org/deeplinks/2012/02/white-house-google-and-other-advertising-companies-commit-supporting-do-not-track. Accessed: 2014-04-10.

[26] Ben Moskowitz and Aza Raskin. MozillaWiki - privacy icons. https://wiki.mozilla.org/Privacy_Icons, June 2011. Accessed: 2014-01-06.

[27] David Wagner and Bruce Schneier. Analysis of the SSL 3.0 protocol. *The Second USENIX Workshop on Electronic Commerce Proceedings*, page 2940, Nov. 1996.

[28] A. Freier, P. Karlton, and P. Kocher. The secure sockets layer (SSL) protocol version 3.0. https://tools.ietf.org/html/rfc6101, Aug. 2011. Accessed: 2014-04-08.

[29] Carlos Jensen, Colin Potts, and Christian Jensen. Privacy practices of internet users: self-reports versus observed behavior. *International Journal of Human-Computer Studies*, 63(1):203–227, 2005.

[30] Markus Jakobsson, Alex Tsow, Ankur Shah, Eli Blevis, and Youn-Kyung Lim. What instills trust? a qualitative study of phishing. 4886:356–361, 2007.

[31] J. Hodges, C. Jackson, and A. Barth. RFC 6797 - HTTP Strict Transport Security (HSTS). https://tools.ietf.org/html/rfc6797, Nov. 2012. Accessed: 2014-04-21.

[32] HTTP Strict Transport Security - the chromium projects. http://dev.chromium.org/sts. Accessed: 2014-04-21.

[33] P. Guttman. PKI: it's not dead, just resting. *Computer*, 35(8):41–49, Aug. 2002.

[34] Lance J. Hoffman, Kim Lawson-Jenkins, and Jeremy Blum. Trust beyond security: an expanded trust model. *Communications of the ACM*, 49(7):94–101, 2006.

[35] Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. Crying wolf: an empirical study of SSL warning effectiveness. *USENIX Security Symposium*, pages 339–416, Aug. 2009.

[36] JavaScript guide - JavaScript — MDN. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide. Accessed: 2014-04-21.

[37] DOM developer guide — MDN. https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM. Accessed: 2014-04-21.

[38] W3C DOM4. http://www.w3.org/TR/2014/WD-dom-20140204/, Feb. 2014. Accessed: 2014-04-09.

[39] XMLHttpRequest level 1. http://www.w3.org/TR/XMLHttpRequest/, Jan. 2014. Accessed: 2014-04-23.

[40] Huddle. Resemble.js : Image analysis and comparison. http://huddle.github.io/Resemble.js/. Accessed: 2014-04-08.

[41] Evan Klinger and David Starkweather. pHash.org: Home of pHash, the open source perceptual hash library.

[42] Neal Krawetz. Looks like it - the hacker factor blog. http://www.hackerfactor.com/blog/index.php?/archives/529-Kind-of-Like-That.html, May. 2011. Accessed: 2014-04-08.

[43] D.H. Ballard. Generalizing the Hough Transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[44] Neal Krawetz. Kind of like that - the hacker factor blog. http://www.hackerfactor.com/blog/index.php?/archives/529-Kind-of-Like-That.html, Jan. 2013. Accessed: 2014-04-08.

[45] Qualtrics: Online survery software & insight platform. http://www.qualtrics.com/. Accessed: 2014-04-08.

[46] Humphrey Taylor. Most people are privacy pragmatists who, while concerned about privacy, will sometimes trade it off for other benefits. March 2003.

[47] Dennis Fisher. Final report on DigiNotar hack shows total compromise of CA servers. http://threatpost.com/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112, Oct. 2012. Accessed: 2013-10-28.

Appendix A

Participant Questionnaire

A.1 Introduction

1. You will be shown images of nine online loan websites. After each one, you will be asked to rate how much you would trust the website at face value on a scale of 1 to 5, with 1 being a very untrustworthy site and 5 being a very trustworthy site. You will be unable to interact with the website itself in any way, so do not rate it lower simply because you cannot follow a link or click a button.

   If your screen is not large enough to view a website, you may scroll it left and right to view more. Please view the entire website before selecting your rating.

2. Please confirm that you understand the ratings by selected the most trustworthy rating.[11]

   - 1
   - 2
   - 3
   - 4
   - 5

A.2 Demographics

1. What type of school are you enrolled in?

   - 2-Year College
   - 4-Year College
   - Graduate School
   - Other

2. What is your current class year?

   - Freshman
   - Sophomore
   - Junior
   - Senior
   - Graduate Student

3. What is your age?

   - Under 18
   - 18-20
   - 21-23
   - 24-25
   - Over 25

4. What is your gender?

   - Male
   - Female

5. What is your preferred web browser?

    - Microsoft Internet Explorer
    - Mozilla Firefox
    - Google Chrome
    - Apple Safari
    - Opera
    - Other / Don't Know

A.3 Privacy Segments

1. Please answer each of the following as truthfully as possible.

2. I am concerned about online identity theft.

    - Strongly Agree
    - Agree
    - Not Sure
    - Disagree
    - Strongly Disagree

3. I am concerned about my privacy online.

    - Strongly Agree
    - Agree
    - Not Sure
    - Disagree
    - Strongly Disagree

4. I am concerned about my privacy in everyday life.

   - Strongly Agree
   - Agree
   - Not Sure
   - Disagree
   - Strongly Disagree

5. I am likely to read the privacy policy of an ecommerce site before buying anything.

   - Strongly Agree
   - Agree
   - Not Sure
   - Disagree
   - Strongly Disagree

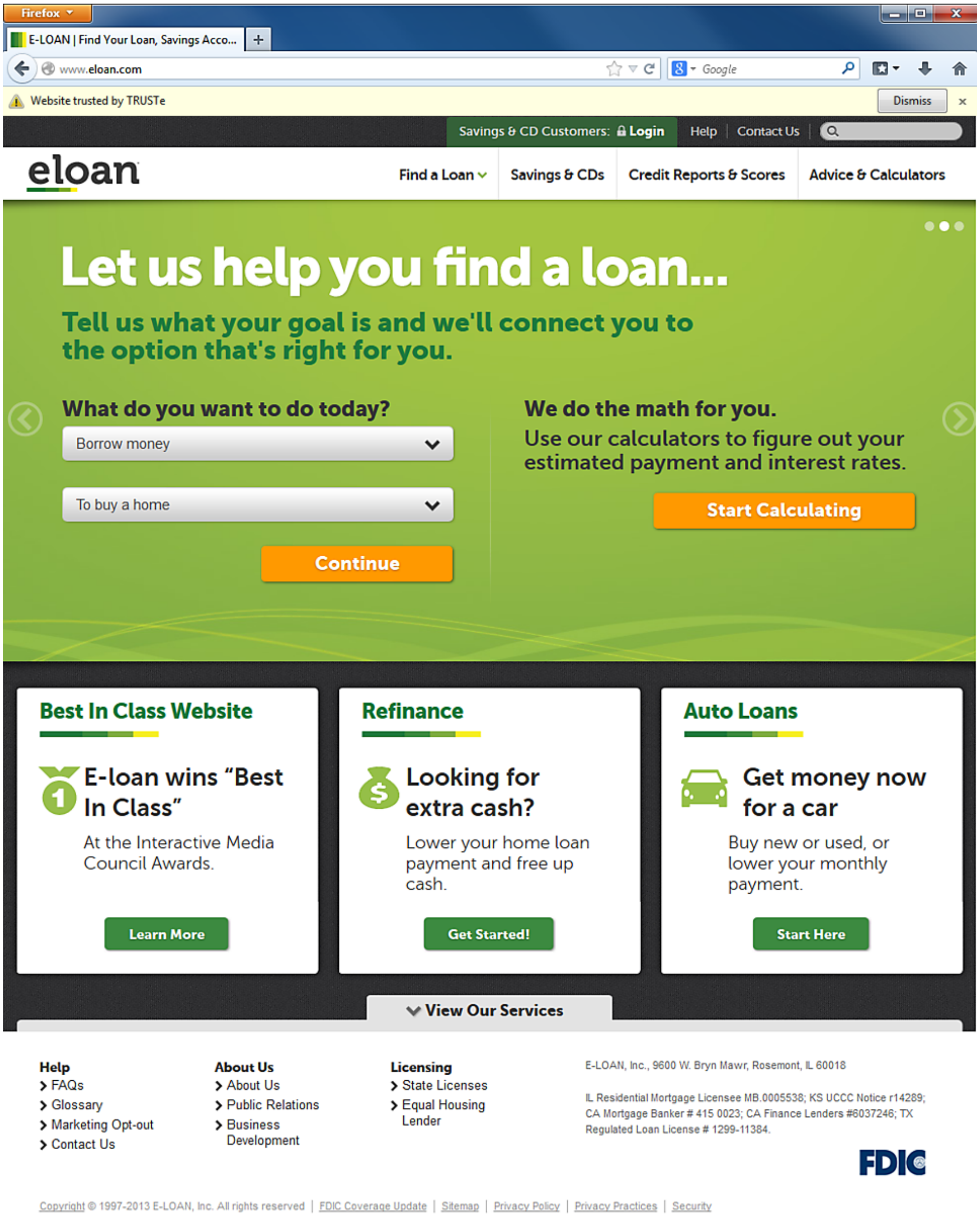
6. Privacy policies accurately reflect what companies do.

   - Strongly Agree
   - Agree
   - Not Sure
   - Disagree
   - Strongly Disagree

A.4 Trial

1. Please analyze the screenshot below, then answer the questions that follow.

   [User is shown screenshot of website with condition. Example in figure A.1.]



Let us help you find a loan...

Tell us what your goal is and we'll connect you to the option that's right for you.

What do you want to do today?

Borrow money

To buy a home

We do the math for you.

Use our calculators to figure out your estimated payment and interest rates.

Best In Class Website

E-loan wins "Best In Class"

At the Interactive Media Council Awards.

Refinance

Looking for extra cash?

Lower your home loan payment and free up cash.

Auto Loans

Get money now for a car

Buy new or used, or lower your monthly payment.

E-LOAN, Inc., 9600 W. Bryn Mawr, Rosemont, IL 60018

IL Residential Mortgage Licensee MB.0005538; KS UCCC Notice r14289; CA Mortgage Banker # 415 0023; CA Finance Lenders #6037246; TX Regulated Loan License # 1299-11384.

Copyright © 1997-2013 E-LOAN, Inc. All rights reserved | FDIC Coverage Update | Sitemap | Privacy Policy | Privacy Practices | Security

Figure A.1: *An example of the screenshots shown to users. This website has the augmented condition applied.*

2. How much do you trust this website? Please rate on a scale of 1 to 5.

    - 1 (Very Untrustworthy)
    - 2 (Untrustworthy)
    - 3 (Neutral)
    - 4 (Trustworthy)
    - 5 (Very Trustworthy)

3. What features of the website did you use to assign your rating?

    [Open Response]

4. Have you visited or heard of the website before this interaction?

    - Yes
    - No

5. If it was requested, what information would you be willing to provide to this website? Select any that apply.

    - Location
    - Age or birthday
    - Real name
    - Telephone number
    - Home address
    - Credit card information
    - Banking information
    - Financial history
    - Social security number

A.5 Post-Analysis Questions

- Did you notice this image in any of the websites?

  [User is shown TRUSTe Logo, as seen in 4.1]

  - Yes
  - No

- What do you think the above image indicates?

  [Open Response]

- Did you notice this image in any of the websites?

  [User is shown TrustVoucher indicator, as in figure 5.5]

  - Yes
  - No

- What do you think the above image indicates?

  [Open Response]