Security Usability of Petname Systems[*]

Md. Sadek Ferdous[1,2,3], Audun Jøsang[2,3]
Kuldeep Singh[1,2,5] and Ravishankar Borgaonkar[2,5,6]

[1]NTNU, [2]UNIK, [3]University of Oslo, [4]University of Tartu, [5]TKK, [6]KTH
{sadek,josang,kuldeep,ravishankar}@unik.no

Abstract. To have certainty about identities is crucial for secure communication in digital environments. The number of digital identities that people and organizations need to manage is rapidly increasing, and proper management of these identities is essential for maintaining security in online markets and communities. Traditional Identity Management Systems are designed to facilitate the management of identities from the perspective of the service provider, but provide little support on the user side. The difficulty of managing identities on the user side causes vulnerabilities that open up for serious attacks such as identity theft and Phishing. Petname Systems have been proposed to provide more user friendly and secure identity management on the user side. This paper provides an analysis of the Petname Model by describing its history and background, properties, application domains and usability issues with emphasis on Security Usability. By covering a broad set of aspects, this paper is intended to provide a comprehensive reference for the Petname System.

1 Introduction

The purpose of digital communication protocols is to exchange information as efficiently and reliably as possible. Originally, these protocols were designed without authentication because the identities of communicating parties could be assumed, and did not have to be formally verified. Authentication was subsequently added for verifying the correctness of claimed and assumed identities. Authentication requires prior registration of identities, and is based on a set of security mechanisms combined with a credential or security token. As authentication became necessary for accessing many online services, more and more identities and credentials were issued, and their management became problematic, both for service providers and for users. Identity Management (IdM, in short) was introduced by the industry to facilitate server-side management of user identities. Initially, client-side management of user identities was not considered to be an issue. However, many people currently feel overloaded with identities and passwords that security policies require them to memorize. The growing number of identities that users need to handle and the inability of users to comply with

[*] Proceedings of the 14th Nordic Conference on Secure IT Systems (NordSec 2009), Oslo, October 2009

credentials management policies now makes client side IdM a critical issue. It is also important to consider that SP (Service Provider) identities also need to be managed, and this aspect of IdM has received very little attention. Petname Systems, which we will discuss here are precisely focused on client-side management of SP identities. We would like to point out here that we will use the term Petname Model to denote the abstract properties of Petname Systems. An implementation of the Petname Model is then a Petname System.

To understand the Petname Model it is essential to understand why Petname Systems were proposed in the first place. The Petname Model was formally described by Marc Stiegler in his 2005 paper [1]. The potential of the Petname Model, however, was discovered by several people in several successive steps. Elements of the idea behind Petname Systems are scattered among several papers and web articles, and the combined efforts of these authors have shaped the formulation of the Petname Model. Section 2 aims to summarize the existing literature.

Section 3 defines the Petname Model by outlining its different components and establishing the connections among them. A Petname System can have several properties and its potential applications can span over several disciplines of computing and networking. A long list of properties as well as several application scenarios were listed in [1]. Section 4 formalizes the properties in a more systematic way by dividing them into two broad categories: 1) Functional properties and 2) Security Usability properties, and also by adding new usability requirements. Security Usability of Petname Systems will be analyzed in Sect. 5. In Sect. 6, different applications of the Petname Model are explained. The current paper introduces some new application scenarios other than those discussed in [1]. Section 7 analyzes the usability issues of two applications that utilize the Petname Model. Section 8 provides some hints on potential future work on Petname Systems and concluding remarks are provided in Sect. 9.

2 Background and Rationales of Petname Systems

At first, it is important to note the relation between Entity, Identity and Identifier. In the scope of this paper, a person, an organization or a machine (computer) operated by any person or organization will be denoted as entity. Identity is the *fundamental property of any entity that declares the uniqueness or sameness of itself and makes it distinctive from other entities in a certain context* [2]. In general, an entity can have multiple identities, but an identity cannot be associated with more than one entity. Each identity can consist of multiple attributes that are also known as identifiers when used for identification purpose [3]. Here, the same attribute can be associated with multiple identities.

Three desirable properties of an identifier were defined by Zooko Wilcox-O'Hearn in his influential web article published in 2001. According to Wilcox-

O'Hearn an identifier should ideally be Global[1], Unique[2] and Memorable[3] [4, 5]. To be memorable, an identifier has to pass the so-called "moving bus test"[6]. That is, if one can correctly remember a name written on a moving bus for a definite amount of time, that name can be considered memorable. An identifier will be unique if it is collision-free within the domain [1] and has the property that it is strongly resistant against forgery. Wilcox-O'Hearn also claimed with supporting evidence that no identifier could have all the three desirable properties simultaneously, and suggested to choose any two of them according to different scenarios.

A triangle where the three properties are placed in the three corners is commonly known as Zooko's triangle, and represents the basic foundation for the Petname Model. Zooko's triangle is illustrated in Fig.1.

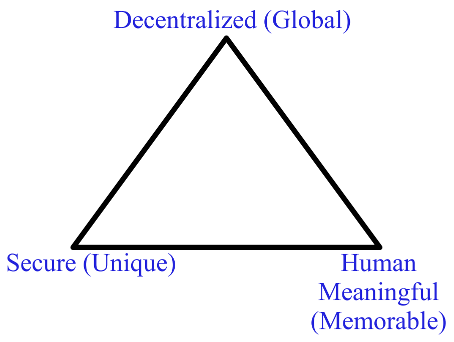


Fig. 1. Zooko's triangle

The idea of placing the three properties at the three corners of a triangle can be explained as follows. In a triangle the three corners are never connected by a single line, only pairs of corners are connected. Placing those three properties in the three corners of the triangle provides a visual analogy to the fact that an identifier can only achieve two of the desirable properties at any one time.

In 2000, Jonathan S. Shapiro, being inspired by the idea of Marc Miller *et al.* while at Electric Communities, described in a web article his scheme of adopting a system which utilized three types of naming conventions: *Petname*, *True Name* and *Nickname* [7]. He adopted this idea for a configuration management system. A True Name is synonymous to a globally unique identifier, the Nickname is a globally memorable name of an entity assigned by its creator, and the Petname is a memorable and locally unique user-assigned name for that entity.

A few months later, Mark Miller published another article [6] in which he, for the first time, documented the structure of the Petname Model with three

[1] Called "Decentralized" in [4]
[2] Called "Secure" in [4]
[3] Called "Human-Meaningful" in [4]

components: Petname, Key and Nickname. These three components are essentially equivalent to Shapiro's Petname, True Name and Nickname respectively. Miller suggested to use the term Key instead of True Name, and pointed out that the Petname Model satisfies all the three desirable properties of Zooko's triangle. Tyler Close suggested to adopt the term Pointer instead of Key [8] and we will also use the term Pointer in this paper.

In 2003, Tyler Close of Waterken Inc. pointed out the possibility of using Petname Systems for better trust management [9]. Waterken Inc. developed the Petname Toolbar for the Firefox web browser. The main motif was to show the potential application of Petname Systems to counter phishing attacks. According to Tyler Close, humans are not capable enough to manage the transition of trust from one entity to another in digital communication and this leads to identity-theft as a result of phishing attacks and suggested to use Petname Systems to enable manual trust evaluation by the user while the transition takes place.

In 2005, Marc Stiegler extended the Petname Model based on Mark Miller's suggestion and also explained the detailed interaction among the components of the Petname Model [1]. He also formalized the properties and requirements for the Petname Model and gave examples of some applications of Petname Systems.

3 The Petname Model

3.1 Rationale

As mentioned in the previous section, Zooko's triangle visualizes the hypothesis that no identifier can be Global, Memorable and Unique at the same time, but can only have two properties. Three unique pairs can be created using these three properties: 1) Global-Memorable, 2) Memorable-Unique and 3) Global-Unique. Even if no identifier can have all the three properties, a *naming system* can be designed to achieve all the three properties of Zooko's triangle. The Petname Model represents one such naming system.

3.2 Components

The Petname Model uses three different types of names that in our terminology are called: Pointer, Nickname and Petname. These three names actually represent the three sides of Zooko's triangle and hence are synonymous to the three pairs discussed above. Detailed explanation for each of them are given below.

Pointer. The Pointer was defined as "True Name" in Shapiro's interpretation and as "Key"in Miller's interpretation. A Pointer implies a globally unique and securely collision free identifier which can uniquely identify an entity. It interconnects the *Global* and *Unique* corners of Zooko's triangle. The security of the Petname Model largely depends on the difficulty to forge a Pointer. A public/private key pair and a fully qualified pathname of a file in an Internet file server are good examples of Pointers. They are globally unique and difficult to forge. However, a Pointer (e.g. a public key, IP address, etc.) may not be memorable to human.

Nickname. The Nickname inter-connects the *Global* and *Memorable* corners of Zooko's triangle. It is an optional non-unique name created by the owner of the Pointer. The purpose of the Nickname is to aid in identifying the entity easily. The title of a web page that is displayed in the title bar of the browser is an example of a Nickname. Users may remember that webpage by the title, but another website may have the same title and can create a collision on the user's mind. Thus a Nickname is not necessarily unique.

Petname. The Petname is a name created by the user to refer to a specific Pointer of an entity. Within the domain of a single user a bidirectional one-to-one mapping exists between Petnames and Pointers. A Petname connects the *Memorable* and *Unique* corners of the triangle. Petnames only have a local scope and may only be relevant for local jurisdiction. The same Petname can be used by different users to refer to either the same Pointer or to different Pointers. The security of a Petname System also depends on the privacy of Petnames and the difficulty to mimic a Petname. Here it is interesting to note that a Petname does not necessarily mean a text-based name. In addition to text, it can also be image and sound or any combination of them in different ways.

3.3 Relationship among the Components

There is a bidirectional one-to-one mapping between Pointers and Petnames within the domain of each user. A Nickname has a one-to-many relationship to the set of Pointers. A Pointer is assumed to map to a single Nickname, but can map to several Alleged Names in the global domain. An *Alleged Name*, like the Nickname, is the introductory/referred name for an entity given by a third party. The distinction between a Nickname and an Alleged Name is that the Nickname is created by the owner of the entity and the Pointer whereas the Alleged Name is provided by a third party. The relationship between Petnames and Nicknames can be confusing sometimes. In some situations, a Nickname can be used as a Petname or in other situations a Petname can be derived from the Nickname. A single Nickname can always be uniquely resolved from the Petname, but the Nickname is not necessarily unique for the Petname. For that reason, a Petname can not be uniquely resolved from a Nickname. Figure 2 illustrates this relationship. As seen from the figure, the Petname Model is actually a naming convention built on top of Zooko's triangle.

4 Properties of Petname Systems

The properties of a Petname System can be divided into two broad categories: Functional properties and Security Usability properties.

4.1 Functional Properties

Functional properties are those basic properties that are mandatory for a Petname System. The functional properties are [1]:

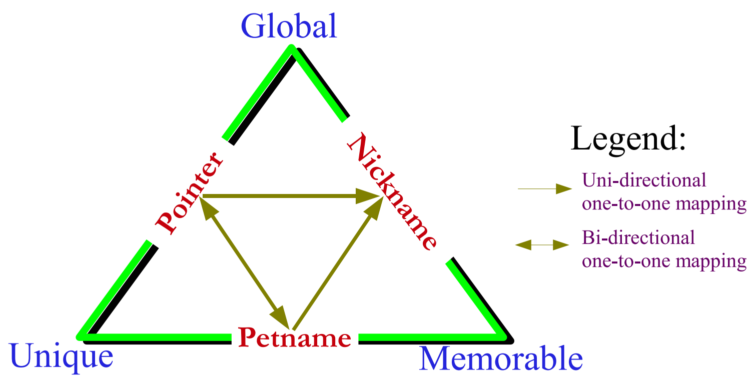

Global
Unique
Memorable
Pointer
Nickname
Petname
Legend:
Uni-directional one-to-one mapping
Bi-directional one-to-one mapping

Fig. 2. The Petname Model

F1. A Petname System must consist of at least a Pointer and a Petname.
F2. Nickname is optional.
F3. Pointers must be strongly resistant against forgery so that the Pointer can not be used to identify a false entity.
F4. For every user there must be a bi-directional one-to-one mapping between the Pointer and the Petname of each entity.

4.2 Security Usability Properties

Security usability will ensure the reliability of using the system and enables the user to draw conclusion on the actual security of the system. These properties will ensure that the Petname System is not affected by usability vulnerabilities. Usability properties can again be categorized in two types [10]:

1. A security action is when users are required to produce information and security tokens, or to trigger some security relevant mechanisms. Security action enables a user to interact securely with an entity. For example, typing and submitting a password is a security action. Properties related to the security action in the Petname System are [1]:

   SA1. It is the user who must assign the Petname for each Pointer.
   SA2. Users must assign the Petname for the Pointer with explicit action.
   SA3. As the relationship between the user and other entities evolve, the user should be able to edit the previously applied Petname for a Pointer to a new Petname.
   SA4. Suggestion on the Petname based on the Nickname can be provided as an aid for the user to select a Petname for a Pointer. If the Nickname is missing, other criteria could be chosen for the suggestion.
   SA5. If a suggestion is provided and the user wants to accept it as the Petname, then he must do so with explicit action.

SA6. Petname Systems must make sure that the user-selected, created or suggested Petname is sufficiently distinct from the Nickname so that the user does not confuse them with each other.

SA7. Petname Systems must make sure that the user-selected, created or suggested Petname must be sufficiently different from existing Petnames so that the user does not confuse them. This is needed to reduce the risk of mimicry of the Petname upon which the security of the Petname System largely depends.

SA8. If the user chooses a Petname that may resemble a Nickname or other Petnames, he should be warned explicitly.

SA9. The User should be alerted to apply a Petname for the entity that involves in highly sensitive data transmission.

2. A security conclusion is when users observe and assess security relevant evidence in order to derive the security state of systems. Security conclusions enable the user to conclude on the security state of the system by observing security relevant evidence and assessing this together with assumptions. For example, observing a closed padlock on a browser, and concluding that the communication is protected by TLS is a security conclusion. Properties related to the security conclusion are [1]:

SC1. The Pointer and the corresponding Petname must be displayed at all times through the user interface of the Petname System. This will make the user confident about his interaction and help to draw the security conclusion easily.

SC2. The Petname for a Pointer should be displayed with enough clarity at the user interface so that it can attract the user's attention easily.

SC3. The absence of a Petname for a Pointer should be clearly and visually indicated at the user interface so that the user is surely informed about its absence.

SC4. The visual indication for suggested Petnames and Nicknames should be unambiguous enough so that the user does not confuse them with each other.

SC5. The warning message that will be provided when there is a direct violation of any of the above properties should be clear enough so that the user can understand the problem and take the necessary security action.

5 Evaluation of Security Usability for Petname Systems

The usability of security is crucial for the overall security of the system, but is still a relatively poorly understood element of IT security. Therefore it is important to evaluate the Security Usability of Petname Systems as it is directly related to the security of client-side Identity Management. A set of general Security Usability principles related to Identity Management were proposed in [10]. We will use these principles as a basis to evaluate the Security Usability of the Petname System by analyzing if the Security Usability properties of the Petname System satisfy these principles. The Security Usability principles are described below:

Security Action Usability Principles:

A1. Users must understand which security actions are required of them.
A2. Users must have sufficient knowledge and the ability to take the correct security action.
A3. The mental and physical load of a security action must be tolerable.
A4. The mental and physical load of making repeated security actions for any practical number of instances must be tolerable.

Security Conclusion Usability Principles:

C1. Users must understand the security conclusion that is required for making an informed decision.
C2. The system must provide the user with sufficient information for deriving the security conclusion.
C3. The mental load of deriving the security conclusion must be tolerable.
C4. The mental load of deriving security conclusions for any practical number of instances must be tolerable.

The Security Usability properties of Petname Systems can now be analyzed according to these security principles. When a Petname System satisfies SA1-SA3 and SA6-SA9 of the Security Action properties, it implicitly implies that principles A1 and A2 are also satisfied, because the former properties enable a user to select a unique and unambiguous Petname for a Pointer. This selection of a unique and unambiguous Petname for a Pointer can be thought of as the correct security action as it enables the user to securely identify an entity. Security Action properties SA4-SA8 will act as the aid for the user to select a Petname for a Pointer. We believe that selecting an unambiguous Petname will pose the most significant mental load for the user in the Petname System when repeated for several entities. Such mental load will be reduced significantly if these five properties are satisfied in a Petname System because users do not have to think about the ambiguity of the new Petname with other existing Petnames. Automated suggestion could also be a great aid in such selection. Therefore satisfying these five properties will implicitly lead to the principles A3 and A4 also being satisfied.

To analyze the Security Conclusion properties of the Petname System, we have to first define Security Conclusion in the Identity Management perspective. Security Conclusion in the Identity Management perspective is to correctly identify a specific entity. Displaying the Petname for a Pointer that points to the desired entity at the user interface will enable the user to draw conclusion that this Pointer and in turn the entity the user is interacting with is the intended one. The presence and absence of the Petname will provide the user with enough information to draw the security conclusion easily. So whenever a Petname System satisfies SC1-SC3, it will explicitly satisfy C1 and C2. Different visual techniques should be applied to help the user reduce their mental load in deriving security conclusion. Using different eye-catching colors to indicate the presence or absence of a Petname for a specific Pointer can be an example of one such visual technique. The security conclusion properties SC2-SC5 should be applied to enable a user to draw conclusion with ease and thus if followed will satisfy principles C3 and C4.

From the above analysis we can conclude that a complete implementation of all the properties of a Petname System will satisfy all the security usability principles.

6 Application Domains

The presence of the Petname Model is so ubiquitous that people may sometimes be unaware of its existence. Here we will highlight the possible domains in which the Petname Model is used, intentionally or unintentionally, or could be used. For each of the applications we will try to determine the suitability of applying the Petname Model [1].

Real World. The principle of the Petname Model is so naturally integrated in the real world that we do not notice its existence. Let us first analyze how people actually recognize each other. This process is very simple and natural to us: through several physical attributes like face, voice, physique or maybe combinations of them. These combinations can be thought of as the Pointer to uniquely identify a single person. That single person introduces himself to us by stating his name XYZ which is actually a Nickname in the Petname Model terminology. From then on we may perceive that man's identity as Mr. XYZ, which actually represents a Petname. Now if another person also introduces himself as XYZ, then our mind does not only assign that name as his Petname because it was already assigned to another person. Here things may evolve in different directions. One possible direction can be that our mind distinguishes between those two persons and changes the Petname for the first person as Mr. XYZ of London and Mr. XYZ of Paris for the second person or whatever seems practical.

Phone/E-mail Contact List. A phone/email contact list is another classic example of a Petname System. The phone number with international format (preceding the number with + or 00 and country code) may represent the Pointer and it is unforgeable and globally unique. We save the number in our contact book by placing a name for it which is nothing but a Petname for that number. Nicknames are absent here. The same analogy applies for email contact lists. Email addresses represent Pointers. A *From-field* in an email header may contain only the email address: xyz@yahoo.com or a given name by the sender with his email address: Mr. XYZ <xyz@yahoo.com>. Here the given name (Mr. XYZ) represents the Nickname. After receiving a mail from a new sender one can save the sender's email address in the email contact list. At that time a Petname is created by inserting a name suitable to identify that person, or by simply keeping the Nickname.

IM Buddy List. In the domain of a particular Instant Messaging Service each entity has a unique Id (email Id for yahoo, hotmail or passport service) which represents the Pointer for that entity. But sometimes those Ids can have quite close resemblance (logicman and 1ogicman, the second one actually is a 1 not a small L) to each other and thus can be quite confusing for the user to differentiate. A better option is used in the interface of the Instant Messenger where one can put a name for each of the IDs. Such name is actually a Petname. In the user interface all the interactions with the ID is usually done with the Petname and thus making the IM Buddy list a good example of a Petname System. Nicknames are absent here.

DNS & Anti-Phishing Tool. As mentioned earlier, that two domain names can be quite close to each other (typo squatting), intentionally or unintentionally, which can lead to phishing or pharming attacks, Petname Systems can be a useful tool to thwart this type of attack. The domain name itself represents the Pointer. The title in the title bar of the browser for that domain name is the given Nickname. In the user interface (the browser), the user can provide a Petname for each domain name. All the interactions with that domain will be indicated by the Petname in the user interface. Providing a Petname for each domain name will impose a trust relationship to that domain name. Absence of Petname will indicate the absence of a trust relationship.

On the background of the above scenarios, the typical e-commerce transaction scenario can be analyzed. A user frequently shops online and places his trust in PayPal to process his online transaction. Now to safely process his transaction he can define a Petname for PayPal in his browser. Assume that the user visits an e-commerce site that offers an item he wants to buy, but the users does not trust the site to know his credit card details. Luckily the site allows him to pay through PayPal, so he is redirected to `www.paypal.com` when the transaction enters the payment phase. Assuming that he has already defined a Petname for PayPal, his browser should indicate the Petname for it and he feels confident that it really is PayPal, and authorizes the transaction. Assuming that the e-commerce site is fraudulent, and redirects him to `www.paypa1.com` (note that it is “1”, not a small “L”) to phish him, his browser will not find a corresponding Petname because the domain name does not match. The missing Petname will alert him that the PayPal site is fraudulent, and that he should abort the transaction.

This idea has been utilized in three Firefox extensions to develop a defense mechanism against Phishing Attack. These are: the Petname Tool [11], developed by Tyler Close, TrustBar [12], developed by the TrustBar team at the Dept. of Computer Science in the Bar Ilan University, Israel and Passpet [13], developed at the CYLAB of the Carnegie Mellon University.

IP Address. Not all IP addresses have domain names. If one would like to communicate only utilizing IP address, a Petname Model can be applied locally as a substitute for domain names. IP addresses are hard to remember, and Petnames will make it easy to refer to them. IP addresses will represent the

Pointer, and the corresponding Petname will be used at the user interface. All communication from the user's side will be based on Petnames.

Process Handling. Every modern OS runs a number of processes simultaneously. *ps -e* command in Linux or the process tab in the task manager for Windows shows a long list of processes. Some of the process names are so obscure that it is impossible for the user to understand their functionality. A Petname Model can be applied to improve the situation significantly. When a process runs for the first time it will present a short description of what it will do. Then the user can create an informative Petname for that process. This Petname will be displayed in the memory map, for example in the process tab in task manager or with *ps -e* command. In this case the Pointer does not have to be global. It is simply the unique process name or unique command used to run the process.

7 Evaluation of Security Usability for Petname System applications

Having formalized the properties of Petname Systems, and having analyzed security usability issues on a general level, the security usability for two existing Petname System applications are analyzed with the Cognitive Walkthrough method. The applications to be analyzed are : 1) Petname Tool and 2) TrustBar. Both toolbars are designed only to work with the Firefox browser, and are aimed at simplifying client-side management of SP identities and at providing a better defense mechanism against Phishing attacks. Though the application domains for the Petname System is much broader, as described in Sect. 6, we have decided to confine our evaluation only to these two in order to focus on managing SP identities at the client side. These two particular applications exactly meet this criterion.

The Cognitive Walkthrough method is a usability evaluation method in which an evaluator or a group of evaluators participate to identify the usability issues of an application by visually inspecting the user interface. Because Petname Systems affect the user interface, Cognitive Walkthrough is a suitable method for evaluating their usability. While performing the Cognitive Walkthrough for each application, we will try to note if the application satisfies the usability properties discussed in Sect. 4. The degree of compliance with the specified security usability properties will give an indication of the level of security usability of each application. For the evaluation we will be using Firefox version 3.0.7 with Nightly Tester Tool, a Firefox add-on, installed.

7.1 Evaluation of the Petname Tool

The Petname Tool is available as a Firefox add-on in [14]. Once installed the toolbar will look like Fig.3. The Petname Tool is very simple, however, one may almost feel that it is too simple. It does not come with any text label; only a text field to enter Petnames. Absence of a text label can confuse unfamiliar

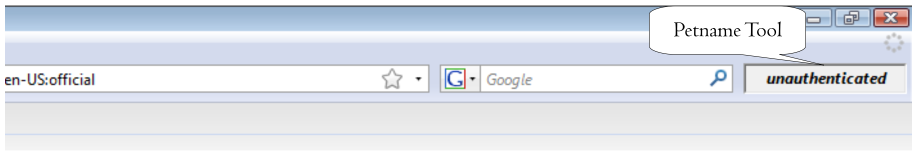


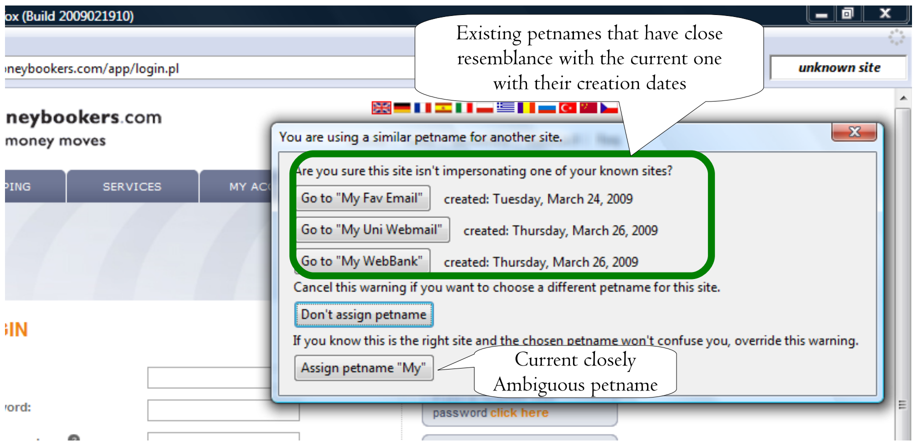
Fig. 3. The Petname Tool in Firefox

users because they might not understand its purpose. The Petname Tool does not work for non-https sites, therefore it will not be possible for a user to assign Petnames to non-https sites. Another thing is worth to note that the Petname Tool uses the hash of the public key of a website as a Pointer. Therefore if the site receives a new certificate and thus a new public key, the Petname Tool will fail to map between the already assigned Petname and the Pointer. A possible solution could be to let URL or domain name be the Pointer that will also remove the restriction of applying Petnames for https sites only.

In the following, the Petname Tool will be analyzed for compliance with the Petname System properties. The Petname Tool, obviously, deploys Petnames. The hash of the public key, derived from the certificate, represents the Pointer and is strongly resistant against forgery. Therefore we conclude that the Petname Tool satisfies F1 and F3. But a serious restriction of the Petname Tool is that it allows users to assign exactly the same Petname for different entities, thus violates the principle of bi-directional one-to-one mapping for each entity and therefore also violates F4. It does not deploy Nicknames and therefore does not satisfy the optional property F2.

The Petname Tool enables users to explicitly assign a Petname for each entity, e.g. to select the text field, write down a Petname and hit the Enter key. This satisfies SA1 and SA2. Users can change any Petname any time, thereby satisfying SA3. No suggestion is provided for aiding the user to select a Petname, which is not compliant with SA4 and SA5. Also Nicknames are not used in the Petname Tool, resulting in non-compliance with SA6. Whenever a user selects a Petname that closely resembles or similar to the existing Petnames, the user is alerted with an informative dialog box (Fig.4). The dialog box displays the existing Petnames to which the current Petname has close resemblance. The user can ignore the alert by clicking the *Assign petname* button or he can cancel this current Petname by clicking the *Don't assign petname* button. If the new Petname is similar to the existing one and he clicks Assign Petname button, the same Petname will be displayed for both websites when he visits them later. Therefore, the Petname Tool is compliant with SA8 (showing the two dialog boxes with the warning), but directly violates SA7. The Petname Tool does not show any alert when there is highly sensitive data transmission and therefore indicates the absence of SA9.

The Petname, if already supplied by the user, is displayed on the Petname Tool toolbar, thereby satisfying SC1. Different typefaces, tooltips and colors have been used in the Petname Tool to catch the user attention to indicate the


Existing petnames that have close resemblance with the current one with their creation dates
unknown site
Go to "My Fav Email"
Go to "My Uni Webmail"
Go to "My WebBank"
Don't assign petname
Assign petname "My"
Current closely Ambiguous petname

ox (Build 2009021910)

neybookers.com/app/login.pl

You are using a similar petname for another site.

Are you sure this site isn't impersonating one of your known sites?

created: Tuesday, March 24, 2009

created: Thursday, March 26, 2009

created: Thursday, March 26, 2009

Cancel this warning if you want to choose a different petname for this site.

If you know this is the right site and the chosen petname won't confuse you, override this warning.

Fig. 4. Dialog box that warns the close ambiguity/similarity among different Petnames

presence or absence of a Petname. In our opinion, white and light green as used by the Petname Tool is less visible than Red, Yellow or Green, as suggested in [15]. In addition, blinking text or different text colors could be used to draw more user attention. Nevertheless, we can conclude that the Petname Tool is compliant with SC2 and SC3. As there is no suggested Petname or Nickname in the Petname Tool, it does not satisfy SC4. The Petname Tool provides warning through dialog boxes when there are conflicts with other Petnames or if there is ambiguity between Petnames and thus satisfies SC5. However, it does not provide a warning message when there is a violation for other properties.

Apart from security usability issues, there are some other weaknesses in the Petname Tool. For example, there is no help button that could explain what the user has to do to utilize it properly. It does not provide the standard *About* menu item that could explain the purpose of the Petname Tool.

7.2 Evaluation of the TrustBar

The TrustBar Tool is available as a Firefox add-on in [12]. The current version of TrustBar is not compatible with the latest Firefox version. Therefore the Nightly Tester Tool, another Firefox add-on, was used to resolve the compatibility issues. Once installed the toolbar looks like Fig.5. TrustBar overcomes some of the shortcomings of the Petname Tool. For example, it works for non-https sites, provides an excellent *Help* feature and also comes with the standard *About* menu item that provides a short description of what it does.

The following simple analysis of TrustBar gives an indication of how it satisfies the properties of the Petname Model. TrustBar utilizes Petnames, and thereby complies with F1. The domain name or URL represents the Pointer and is strongly resistant against forgery. Therefore we conclude that TrustBar satisfies F1 and F3. TrustBar also displays a Nickname in the form of the orga-

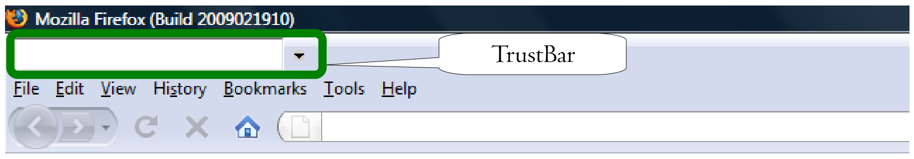

Mozilla Firefox (Build 2009021910)
TrustBar
File Edit View History Bookmarks Tools Help

Fig. 5. TrustBar installed in Firefox

nization name, if a certificate is available or in the form of the domain name for non-https sites and thus satisfies F2. However, a serious restriction of TrustBar is that it allows users to assign exactly the same Petname for different entities as demonstrated in the next paragraph, thus violates the principle of bi-directional one-to-one mapping for each entity and therefore violates F4.

TrustBar enables a user to assign a Petname for each entity so he has to act explicitly, e.g. select the text field, write down a Petname and hit the Enter key, to enable the Petname and this satisfies SA1 and SA2. Users can change any Petname any time and thus TrustBar meets the requirement of SA3. A suggestion is provided in the form of a Nickname for aiding the user to select a Petname if a server certificate is available and this satisfies SA4 partially, and the user has to act explicitly, e.g. by hitting the Enter key so the text field turns to light green (an indication for accepting the Petname) to accept the Nickname as the Petname. This satisfies SA5 too. A serious restriction of TrustBar is that it allows users to assign ambiguous Petnames or even equal Petnames for different entities. It does not provide any sort of warning to users about the ambiguity or similarity of the Petnames and thus directly violates SA7 and SA8. TrustBar allows a user to assign a Petname at his will whenever he feels and does not show any alert when there is highly sensitive data transmission and therefore violates SA9.

The Pointer and the related Petname in the TrustBar, if already supplied by the user, are displayed all the time in the browser toolbar, thereby satisfying SC1. Different icons, tooltips and colors have been used in the TrustBar to catch the user attention to indicate the presence or absence of Petnames. In our opinion it would have been better to use more flashy colors like Red, Yellow or Green instead of pale yellow and light green. Blinking text or different text colors could be used to draw more user attention to potential security problems in websites. Nevertheless, we can conclude that TrustBar satisfies SC2 and SC3. White, pale yellow or light green color has been used to differentiate among non-https Nicknames, https Nicknames and Petnames respectively, thereby satisfying SC4. TrustBar does not provide any sort of warning to the user and this indicates the complete absence of SC5.

7.3 Summery

Table 1 clarifies the distinction between the Petname Tool and TrustBar in terms of the properties of Petname Systems. It can be noted that TrustBar satisfies

Table 1. Comparison between the Petname Tool and TrustBar

	F				SA									SC				
ToolName	1	2	3	4	1	2	3	4	5	6	7	8	9	1	2	3	4	5
Petname Tool	**Y**	N	**Y**	N	**Y**	**Y**	**Y**	N	N	N	N	**Y**	N	**Y**	**Y**	**Y**	N	**Y**
TrustBar	**Y**	**Y**	**Y**	N	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	N	N	N	**Y**	**Y**	**Y**	**Y**	N

more properties of the Petname model than the Petname Tool, though TrustBar has one major shortcoming: absence of any type of warning message. Both tools suffer from the absence of the crucial property F4. As neither of them satisfy all the main properties of Petname Systems, we can conclude that none of them fully satisfies the security usability principles.

8 Future Work

No tool is currently available for Petname based identity management that satisfies all the properties of Petname Model and the corresponding security usability principles, as indicated by our analysis in the previous section. Developing a Petname Model based tool that satisfies the security usability principles should be a priority in future research and development.

As the Petname Model is based on Zooko's triangle, any shortcut in the triangle may collapse the relationships among the components of the Petname Model or may create a new dimension of relationship. Bob Wyman in his web blog proposed to update Zooko's triangle into a pyramid by inserting a new attribute called "Persistent"and connecting it to the other corners. The new attribute was proposed to signify the longevity of each name [16]. This proposal to change the shape of Zooko's triangle can be another potential topic for research which could give Petname Systems additional security properties.

Smart phones are becoming increasingly popular and the number of people that access the Internet from their smart phones is growing every day. Investigations on how the Petname Model can be implemented and adapted for the tiny screen of a mobile phone can be a challenging task and thereby another scope for future research.

9 Conclusions

The Petname Model is naturally embedded in human perception to identify different entities. Implementing it in computer networks and system is a natural extension of human cognitive capabilities and represents a great aid for humans in digital environments. This fact has been demonstrated through several applications, experiments and proposals. A large scale adaptation of the Petname Model is therefore timely.

This paper has focused on providing a brief overview of Petname Systems, formally defined the properties of Petname Systems and explained how these properties can satisfy essential security usability principles. We believe that if these properties are followed in developing applications based on the Petname Model, it will improve the overall security by removing security vulnerabilities related to poor usability. The paper has also analyzed two available Petname-based applications and shown that they represent an improvement in usability, but unfortunately do not satisfy all the specified security usability principles.

References

1. Stiegler, M.: Petname systems (August), 2005. Last visit on May 20, 2009 `http://www.skyhunter.com/marcs/petnames/IntroPetNames.html`.
2. Thanh, D.V., Jørstad, I.J.: The ambiguity of identity. Telektronikk issue on Identity Management **103**(No.3/4-2007, ISSN:0085-7130) (2007) 3–10
3. Jøsang, A., Pope, S.: User centric identity management. In: Asia Pacific Information Technology Security Conference, AusCERT2005, Austrailia. (2005) 77–89
4. Wilcox-O'Hearn, Z.: Names: Decentralized, secure, human-meaningful: Choose two, 2005. Last visit on May 30, 2009 `http://www.zooko.com/distnames.html`.
5. Internet archive wayback machine:snapshot on zooko's writing, 2008 `http://web.archive.org/web/*/http://zooko.com/distnames.html`.
6. Miller, M.: Lambda for humans, 2000. Last visit on May 30, 2009 `http://www.erights.org/elib/capability/pnml.html`.
7. Shapiro, J.S.: Pet names, true names, and nicknames, 2000. Last visit on May 30, 2009 `http://www.eros-os.org/~majordomo/dcms-dev/0036.html`.
8. Close, T.: Naming vs. pointing, 2003. Last visit on May 30, 2009 `http://www.waterken.com/dev/YURL/Analogy/`.
9. Close, T.: Waterken YURL:trust management for humans, 2003. Last visit on May 30, 2009 `http://www.waterken.com/dev/YURL/Name/`.
10. Jøsang, A., Al Zomai, M., Suriadi, S.: Usability and privacy in identity management architectures. In Brankovic, L., Steketee, C., eds.: Fifth Australasian Information Security Workshop (Privacy Enhancing Technologies) (AISW 2007). Volume 68 of CRPIT., Ballarat, Australia, ACS (2007) 143–152
11. Close, T.: Petname tool: Enabling web site recognition using the existing SSL infrastructure, 2006. Last visit on May 30, 2009 `http://www.w3.org/2005/Security/usability-ws/papers/02-hp-petname/`.
12. Trustbar Firefox addon. Last visit on May 30, 2009 `http://u.cs.biu.ac.il/~herzbea/TrustBar/`.
13. Yee, K.P., Sitaker, K.: Passpet: convenient password management and phishing protection. In: SOUPS. (2006) 32–43
14. Close, T.: Petname tool 1.6. Last visit on May 30, 2009 `https://addons.mozilla.org/en-US/firefox/addon/957`.
15. Drelie Gelasca, E., Tomasic, D., Ebrahimi, T.: Which Colors Best Catch Your Eyes: a Subjective Study of Color Saliency. In: Fisrt International Workshop on Video Processing and Quality Metrics for Consumer Electronics, Scottsdale, Arizona, USA, 2005.
16. Wyman, B.: The persistence of identity, 2006. Last visit on May 30, 2009 `http://www.wyman.us/main/2006/12/the_persistence.html`.