Usable Security Management with Host Identity Protocol

Kristiina Karvonen Miika Komu Andrei Gurtov
Helsinki Institute for Information Technology
firstname.lastname@hiit.fi

ABSTRACT

Host Identity Protocol (HIP) proposes a change to the Internet architecture by introducing cryptographically-secured names, called Host Identities (HIs), for hosts. Applications use HIs instead of IP addresses in transport layer connections, which allows applications to tolerate host-based mobility better. HIP provides IPsec-based, lower-layer security, but the problem is that this type of security is invisible for most applications and users. Our main contribution is the implementation and user evaluation of several security indicators which inform the user when HIP and IPsec are securing the connections of the user. We experimented with application and system level security indicators at the client-side, as well as with server-side indicators. In this paper, we present implementation experience on integrating the identity management Graphical User Interface (GUI) to HIP and results of usability tests with actual users.

I. INTRODUCTION

A host and its location are identified using Internet Protocol (IP) addresses in the current Internet architecture. However, IP addresses can serve only as short-term identifiers as a considerable amount of hosts are portable devices and they change their IP addresses when moved from one network to another. Short-term identifiers disrupt long-term transport layer connections, such as Internet phone calls, and make locating the peer host more difficult. Impersonation attacks are possible because IP addresses are relatively easy to forge.
The Host Identity Protocol (HIP) architecture [26, 23] leverages a so-called *identity/locator split* to address these challenges in an integrated approach. It separates the identity of a host from its location as illustrated in Figure 1. The identity is called the *Host Identity (HI)* and it is used as a long-term identifier on the upper layers of the network stack. The location of host is bound to IP addresses and used for routing packets to the host in the same way as in the current Internet architecture.

The HI namespace consists of *Host Identifiers*, each of which consists of the public key component of a private-public key pair. Each host is responsible for creating one or more public/private key pairs to provide identities for itself. As the HIs are based on public-key cryptography, they are computationally difficult to forge. HIs are location-independent identifiers which allow a mobile host to preserve its transport layer connections upon changes in the network. On the other hand, the HI can be used for looking up the current location of a host because the HI is a long-term identifier. A client host obtains the HI of a server typically from the DNS. However, the infrastructure may not support this in certain scenarios, such as in peer-to-peer and ad-hoc environments. In such cases, *opportunistic* HIP can be used for contacting a peer without prior information of the peer's identity. Opportunistic HIP is based on a “leap-of-faith”, which means that it is prone to man-in-the-middle attacks for the initial connection. It is similar to SSH, where the client caches the public key of the server after the first successful connection.

There are many challenges in making HIP understandable to the end users. As an example, users have developed an automatic response to press “OK” to SSH software prompts (meant to verify and accept the key) without consulting the prompts properly, if at all [3]. This is due to many prompts being uninformative to most users that do not increase the user's security awareness even when read [11]. As we implemented a prompting mechanism for HIP-related connections using our publicly available HIP Firefox2 add-on (available also as a firefox3 extension), we witnessed the same phenomenon.

Most Internet applications can run unmodified over HIP [16], although only HIP-aware (new) applications utilizing the extended socket interface [20] can take better advantage of the new features provided by HIP. As HIP secures application data traffic with IPsec that is located logically "deep" within the networking stack, the challenge is to provide proper and understandable security indicators to the user to convince her that the connection, e.g., to a banking web site, is secured. Such indicators can be developed as extensions to applications (e.g., a security add-on to Firefox browser) or within a host-wide HIP management utility that controls all applications.

When designing the security indicators, it is important to decide how and when to apply users’ existing security habits, and when to break them. For HTTPS, a browser typically illustrates the access to a secure site by a padlock icon or by changing the color of the address bar. However, recent research has shown that these indicators might be ineffective as they go unnoticed by most users [31]. We experimented with the usability of the security indicators with volunteers, who accessed and judged security of web pages. Our first implementation prototype and GUI (Fig.2) was targeted to and usability tested with technical users who are assumed to be the first adopters

978-1-4244-3806-8/09/$25.00 © 2009 IEEE

of HIP. Usability needs to be double-checked in later phases of the development with non-technical users [2].

The rest of the paper is organized as follows. Section 2 gives background on HIP and usable security. Section 3 describes the implementation of our security model for HIP. Usability evaluations are presented in Section 4, followed by results and usability improvements in Section 5. Section 6 concludes the paper with a summary and plans for our future work.

II. Background and Related Work

In this section, we compare the security mode provided by Host Identity Protocol to the familiar model of Transport Layer Security. A short overview of related work on usability of network security completes this section.

A. Host Identity Protocol

In HIP [26], IP addresses are used to route packets, but in the upper parts of the stack the addresses are replaced with Host Identifiers. These Host Identifiers form a new Internet-wide name space for hosts. In HIP, each host is directly identified with one or more public keys that each corresponds to a private key possessed by the host. Each host generates one or more public/private key pairs to provide identities for itself.

For backwards compatibility with networking APIs, applications use shorter representation of the HI. IPv4 applications use 32-bit *Local Scope Identifiers* (LSIs), and IPv6 applications use 128-bit *Host Identity Tag*s (HITs). A HIT is constructed by calculating a digest over the public key. A HIT binds the application to the public keys used for the communications, which is referred as *channel binding*.

The introduction of new end-point identifiers changes the role of IP addresses. When HIP is used, IP addresses become pure topological labels, naming locations in the Internet. One benefit of this identity/locator separation is that hosts in private address realms (behind NATs) can name each other in a unique way with HITs [21]. A second benefit is that the hosts can change their IP address without breaking transport layer connections of applications and rely on HIP to manage host mobility. Thus, the relationship between location names and identifiers becomes dynamic.

The problem of certifying the keys in Public-Key Infrastructure (PKI) or otherwise creating trust relationships between hosts has explicitly been left out from the HIP architecture, as it is expected that each system using HIP may want to take care of it in a different manner. For mere mobility and multi-homing, the systems can work without any explicit trust management, in an opportunistic manner.

HIP uses IPsec as described in [6] to provide data encryption and integrity protection for network applications. Before two network applications can communicate with each other using IPsec-protected traffic, the underlying hosts authenticate each other and negotiate encryption keys for IPsec using HIP [27].

B. Transport Layer Security

Transport Layer Security (TLS) provides security for applications at the application layer. TLS is usually supported by user-space libraries that provide an API for the application to communicate securely with a peer application.

When TLS is applied to existing legacy applications, it requires always rewriting both client and server applications. In addition, it requires the allocation of a new transport layer port at the server side because plain TCP and TLS-based TCP connections cannot use the same port. With HIP, no changes to the application are required and the same transport layer port can be used at the server side. This makes HIP easier to deploy even to binary-only legacy applications for which there is no source code available. As such, HIP can be considered as a means for extending the lifetime of legacy network applications which require security or mobility support.

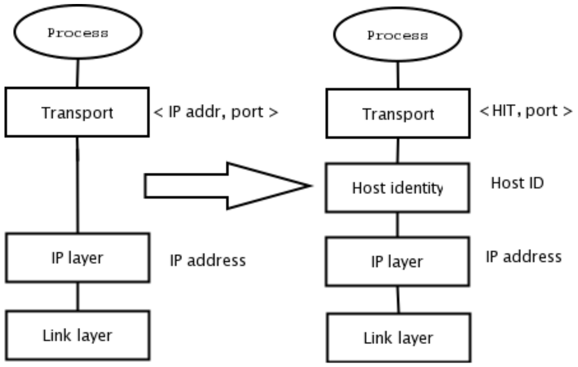


Figure 1. HIP introduces a new layer to the stack

Although there is some on-going effort to make UDP to support TLS [25], currently TLS cannot be used directly with UDP because it assumes a reliable transport protocol. This is problematic especially when VoIP calls need to be protected end-to-end., Fortunately, HIP is applicable also to UDP. HIP supports end-host mobility which is currently missing from the current TLS specification.

C. Managing Security in a Usable Way

The success of any application in managing security depends on its usability; this is also the case with HIP. However, usability and security are often seen as contradicting goals: what is usable cannot be secure and vice versa [9]. A typical example is an easy-to-remember password that tends to be trivial to break, whereas a strong password is hard to recall but security gets breached when users write such passwords down or share them with others [10, 29].

Overall, users can be considered as the "weakest link" in security [1, 29]. If users do not understand the security model behind the user interface, security is at risk. Furthermore, a gap between the mental model of the security experts and non-experts can lead to ineffective and poor communication of how the security works, and what the risks are [5]. Also, in computer security, user errors are in general not acceptable [12]. This leads to the fact that usable security can be described as "usability times two": in security, a single error may be too much, so the generic "trial-and-error" approach will not do. Situation where "the average man" is trying to maintain his security compares to the metaphor of an elephant visiting a store selling objects made of glass - with the lights turned off.

Further, security is usually not the users' primary but secondary goal – an enabler for trustworthy communications of money, private information of personal relationships [11]. Users are not interested or motivated in security *per se*, but rather as means to an end. Because of this, users should be burdened as little as possible with the security features [12], and usage of the security should become a natural part of the actual usage, not an unnatural, add-on extension of the security that introduces an interruption to their primary task, as only too often is the case [11]. For example, unnecessary prompts should be avoided, safe default settings provided, automating as much as possible of the security taking place.

It should also be noted that users do not often realize that they are at risk in a given situation, or what the actual risks are [31] [7]. Users may perceive the risks to be different from what they actually are. However, for the security UI to be successful, it needs to take into account the *perceived* risks or the users do not feel secure. It is a general saying in the field of usability that if the user cannot find functionality, it does not exist. For the UI this means that if the security is not *perceived* to be there, there *is* no security from the user's point-of-view.

A further problem in creating usable security is that users have learned to ignore security indicators. These include usage of padlock icons in the browser address bar and in the lower right corner, the coloring of the address bar, and an extra "s" in the protocol name to show that TLS protocol is being used. There is new work on the browser development side to create standards for web security interfaces [32][30][15]. Unfortunately, they do not work because users do not understand their significance, or the information given is too hard to follow and digest.

Users tend not to know what valid trust marks look like and how to interpret declarations of privacy and how much trust should be induced from their presence [4]. In our previous studies, it became evident that users felt trusting when an image of a visa sign was visible on a given site, falsely inhering that Visa would guarantee their transactions with the site in question. Further, if a user wants the bargain badly enough, he will give up the security [3] [22]. Current security indicators and privacy policy declarations, then, are neither sufficient nor the most usable solution to provide users with information about security [32].

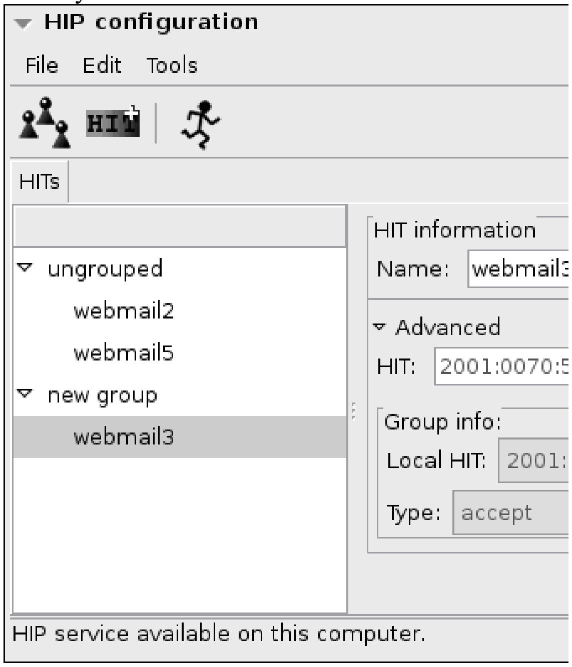
The Extended Validation Certificates [14] approach, intended to overcome some of these obstacles, introduces a new user interface for handling security. However, from usability point-of-view, this scheme seems somewhat problematic, since it includes usage of multiple colors for indicating security. Not only color coding is likely to diminish the overall accessibility, but interpretation and even perception of colors may differ according to cultural variation [8]. It is also difficult for users to identify individual colors in isolation in a reliable way, and the surrounding color scheme of the browser frame and the web page may affect how the color is perceived and how noticeable it is. Furthermore, this type of security indicators may not be noticeable enough either, as e.g. [32] have shown.

III. IMPLEMENTATION DESIGN AND SECURITY MODEL

The user interacts with Firefox web browser and GTK-based HIP GUI. The browser contains an add-on that displays indicators when a connection is based on HITs. The GUI receives notifications on all HIP network communications from the HIP software module. The GUI prompts the user when it is needed: the user can accept or reject new HIP related network connections. Hence, the GUI acts as an end-host firewall for HIP. The user can also use the GUI to sort the server fingerprints to groups as illustrated in Figure 2. The main purpose of the groups is to distinguish between trusted and untrusted peers. Groups can also be used to apply common attributes to all of the fingerprints within the group.

The HIP module translates host names to HITs and provides the browser HIP-based connectivity by intercepting some of the networking related function calls. The module allows varying degrees of authentication for the browser by first trying the strongest authentication method available and then falling back towards weaker authentication methods if the stronger method is not available. At best, the client has obtained the public key of the server already before establishing the connection. This is visible to use both by the browser add-on and the prompt of the HIP GUI. As the second best option, the client tries to establish opportunistic security and learns the public key during the connection set up. This step is visible to the user only by the prompt. Finally, the client uses regular TCP/IP if the server is not HIP capable, which is detected through a timeout. In such a case, no HIP-based security indicators are visible to the user.

It should be noticed that TLS can be used to improve the overall level of security. Thus, the strongest level of security occurs when client uses both HIP and TLS.

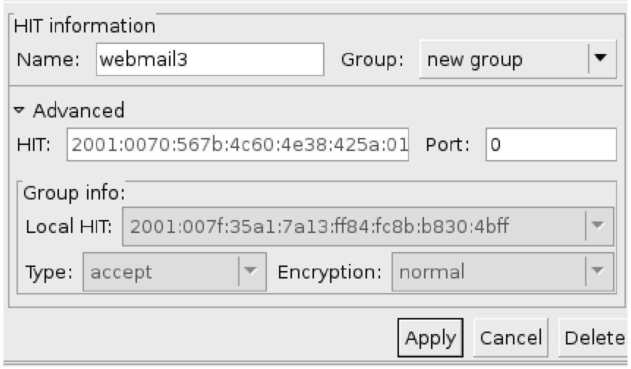


ungrouped
webmail2
webmail5
new group
webmail3

HIT information
Name: webmail3
Advanced
HIT: 2001:0070:5
Group info:
Local HIT: 2001:
Type: accept

HIP service available on this computer.

Figure 2a. HIP management GUI navigation tree



HIT information
Name: webmail3 Group: new group
Advanced
HIT: 2001:0070:567b:4c60:4e38:425a:01 Port: 0
Group info:
Local HIT: 2001:007f:35a1:7a13:ff84:fc8b:b830:4bff
Type: accept Encryption: normal

Figure 2b. A close-up on HIP management GUI

IV. USABILITY TESTING OF HIP

The test design was based on refinements made on a round of pilot tests with 10 users with a mock-up of a Finnish online auction site huuto.net, where users sell and buy personal items to peers. Based on user reactions, we changed the application to a mock-up Webmail, in order to narrow down the complexity of the choices and interactions available to better control. We also considered using an online bank as in [32] but gave up this idea since we were not able to create a mock-up design for a bank that would look convincing enough. On basis of existing literature, Webmail account design is less demanding in order to be experienced trustworthy enough to provide valid data about real-enough user reactions as described in. [13], [18], [8]. We were interested in the following questions: Would users notice the security indicators? Which security indicators would users use for judging webmail security? Would users be able to use HITs? How understandable is the concept of HIT?

According to [22], if users know the test is about security, they tend to become more caring and thoughtful about their actions, acting more responsible they normally would and even then, [3] have shown that privacy policy information tends to be noticed. Users have learned to ignore security in real life as it is often incomprehensibly expressed and tediously presented. These test effects were taken into account when designing and analyzing the test behavior and test results.

A. Test Setting

We used HIPL software branch “gui” with patch level 226 in the usability tests. The OS was Ubuntu 6.10 Linux with Linux 2.6.17.14 kernel. The user operated an IBM R51 laptop which was connected directly to another laptop hosting a number of virtual webmail servers (webmail1-5.) The servers were HIP enabled except for webmail1 and webmail4. All of the servers were running apache2 web server.

The tests were conducted in a lab-type environment: a closed, silent meeting room with no outside disturbances. The usability tests with the first group were conducted at the company premises of the participants. The usability tests with the second group were conducted at the premises of the university. In the test, a moderator observed, and if necessary, guided the user through the test tasks. The test ended with an interview. Another person was taking notes. Users took the test one at a time. The test took approximately 30-45 minutes depending on the user’s eagerness to give feedback, talkativeness in the interview section, and speed of interaction in conducting the HIP management GUI test tasks.

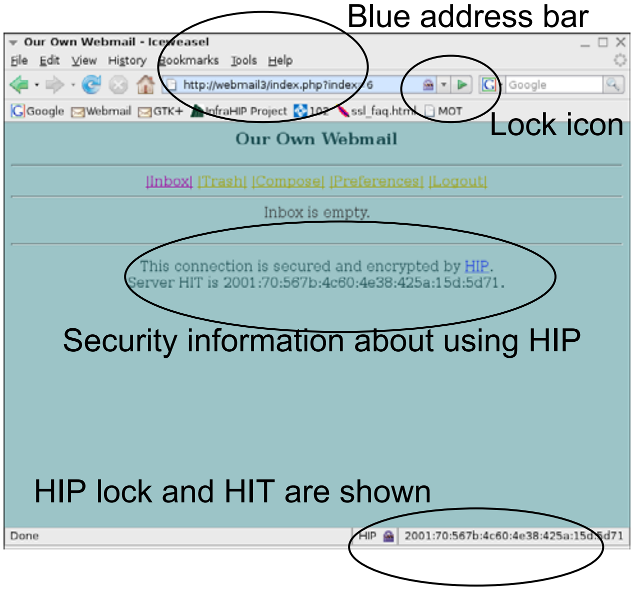
B. Test Users

We had two types of users. **Group 1** consisted of 9 users already familiar with HIP. The users were working for an international network and telecommunications vendor and their work included work on HIP. **Group 2** consisted of 6 users not familiar with HIP, but also this group was technically adept. All users were students or graduates of a technical university, aged between 18-39 years, and familiar with some type of encryption technologies other than HIP. All users were male. No user was color blind.

C. Test Procedure

Users first filled in a background questionnaire (gender, age, average computer usage). Before starting any of the tasks, the users were told shortly about the test setting: HIP was explained to be a new way to provide security in the Internet, providing kind of "fingerprints" of the services used online. Users were also told that they would be logging into email services, and then test out a new user interface to manage the fingerprints HIP created for them during logging into the email services. A talk-aloud protocol was employed: users were asked to tell what they were thinking as they proceeded through the test tasks.

The users would first log into the five Webmail accounts, one by one. The test proceeded from the least secure account login procedure to the most secure. The security indicators were introduced gradually in an incremental fashion. The reason for such a set-up was our hypothesis that users would realize that the security indicators were missing only after their gradual introduction.


Blue address bar
Our Own Webmail - Iceweasel
File Edit View History Bookmarks Tools Help
http://webmail3/index.php?index=6
Google
Google Webmail GTK+ InfraHIP Project 102 ssl_faq.html MOT
Lock icon
|Inbox| |Trash| |Compose| |Preferences| |Logout|
Security information about using HIP
HIP lock and HIT are shown
Done
HIP
2001:70:567b:4c60:4e38:425a:15d:5d71

Our Own Webmail

Inbox is empty.

This connection is secured and encrypted by HIP.
Server HIT is 2001:70:567b:4c60:4e38:425a:15d:5d71.

Figure 3. Webmail 3 site with blue address bar,icons and text that all indicate a HIP secured connection.

This also proved to be the case. The five webmail accounts showed the following security indicators:

Webmail 1 & 4: Insecure connection. There are no security indicators visible, except for the statement in plaintext in the middle of the page saying "This connection is insecure. Please enable HIP".

Webmail 2: The connection is secured with opportunistic HIP. Traditional security indicators are, however, still missing, and the security is stated only in plaintext in the middle of the page saying "This connection is secured and encrypted by HIP". Below the statement, the HIT for the connection is shown.

Webmail 3: HIP module finds the HIT of the server before contacting it and now there are more security indicators visible in addition to the text on the web page: the address bar has turned blue, and there is a picture of padlock both in the address bar and in the lower right corner, with text "HIP" and also the HIT for the site is shown as visualized in Figure 3.

Webmail 5: Both TLS and HIP are used, the address bar has turned yellow, and the same security indicators as in previous case are present. Web server forwarded traffic from HTTP port to HTTPS.

We created random usernames and passwords to be used during the test, instead of asking users to use their own username and password because users have shown reluctance in using this type of personal, private information in test settings in previous tests by us [19] and others [32]. The Webmail addresses and the username and password, of type “username” and “passwd” were presented to the users on paper slips one by one. After logging in, the users were asked to rate the experienced security of the Webmail in question on a scale from 1 to 5, where 1 was considered “insecure” and 5 “secure” We were unsure if users would be willing to use the scale and if they would only use some ratings, but, in fact, it turned out that they used the full scale from 1 to 5, also stating the reasons behind their judgments.

After the Webmail log-ins, the users were asked to complete several tasks with the HIP GUI. The test tasks can be found in **Table 1.** The tasks were described in natural language, e.g. for Task 8 “Can you change the name the group you created earlier?” The word “security” was not mentioned in task description to users in order to avoid bias. Care was taken to see that each user at least tried to accomplish all tasks at some point of the test. With the prototype, not all possible functionalities were available but, they were shown on the GUI to give users some idea of all properties of the security management that GUI could allow for.

Table 1. The usability test tasks (from moderator’s perspective). Tasks 1-6 describe the security level and the order in which users logged into the Webmail accounts. Tasks 7-9 are HIP GUI related test tasks.

Task no	**Task content**
1	Log into insecure webmail1
2	Log into webmail2 with opp. HIP
3	Log into webmail3 with normal HIP
4	Repeat task no 1
5	Repeat task 3, no prompt this time
6	Log into webmail5 with TLS and HIP
7a	Find new fingerprint
7b	Create a new group and rename it
7c	Move a fingerprint
8	Rename a group
9	Delete a fingerprint

All of the test sessions ended with a brief interview, where the user could provide feedback in a free fashion. Users were also asked about their real life usage of security after testing the Webmail accounts and the HIP GUI. The questions in the interview part included self-report on:

- what kind of encounters they had had with security;
- what kind of encounters their friends had had with security;
- if were they conducting online transactions, and if so, what were the payment methods they were using;
- if they had had any problems with security before; and if so, what kind of problems;
- if they were interested in security in general, and if so, how would this relative interest/disinterest manifest itself in their behavior.

D. *Analysis of the Tests*

Figures 4 and 5 show how users evaluated the security of the Webmail accounts and how they succeeded in the tasks related to the HIP GUI management. Figure 4 shows mean and standard deviation of security grades from users in the test: Overall, users evaluated the security level of the HIP protected communications roughly twice as secure as unprotected communications. The case with TLS (and HIP) was rated more secure than HIP communications but the difference was insignificant. Group 1 doubted the security indicators on the web page more than Group 2.

Most users reported awareness of security indicators on websites and claimed they were actively following them. They claimed to be actively following and were familiar with 1) pictures of locks in the browser, 2) changing color of address bar and 3) the 'S' in the HTTPS string, and 4) certificate announcements, all associated with the SSL protocol usage which was trusted by all users in our study. However, in our study, it became evident that in practice this was not really so, as many users did not report that the security indicators were missing from the first Webmail accounts they were shown during the test.

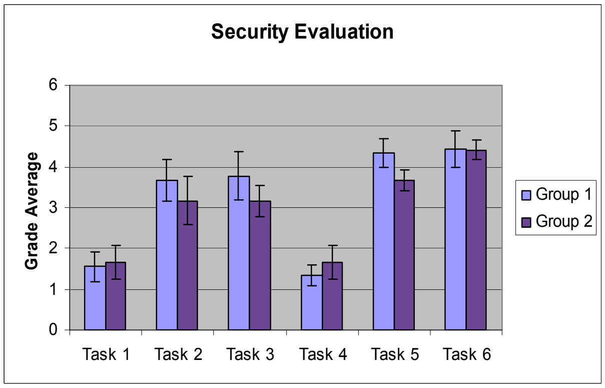


Figure 4. Perceived level of security with the Webmail sites according to user evaluations.

Most users were indulging in online transactions in real life on a regular basis. However, users were not very trusting towards the online service providers they were unfamiliar with. Users reported using several means to protect their assets online, such as a) using only sites they knew well, or b) only making payments via their bank's online services. Some reported also c) having multiple credit cards: one for offline and one for online purchases, with very limited credit limit on the latter in order to minimize the risk. For some of the youngest users in our tests, d) using someone else's card (parents') was one additional way to overcome personal security risks in online situations

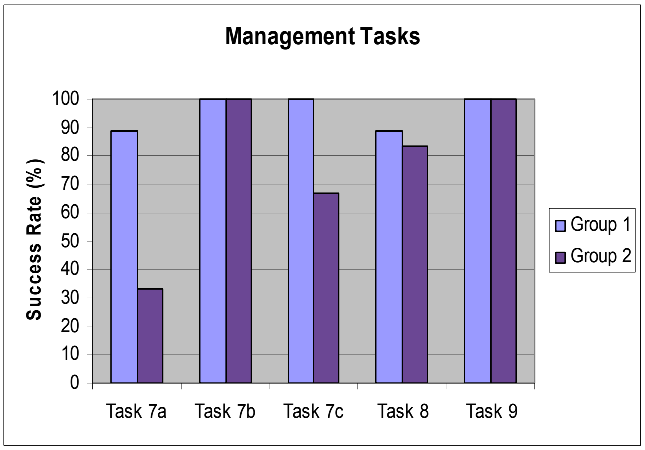


Figure 5. The success rate for management tasks with HIP GUI.

.

Familiarity with the security indicators came up as one of the key ingredients in promoting willingness to feel secure. Users wanted to see something similar to the current SSL implementation also with HIP. This is a natural outcome, as routine ways to deal with new interactions are very often preferred by users - 'old habits die hard'. Yet, even on basis of habitual use patterns, users failed to notice there was a color difference in the address bar; for HIP it was blue, whereas for SSL and HIP-SSL combination it was yellow. This is a remarkable result from the perspective of the EV, since it is to a great extent based on changing the coloring of the address bar to inform the user about the security of the situation. Without educational efforts the users may not notice such indicators, or will misinterpret them. Our users were happy as long as the address bar was colored, *regardless of the color*.

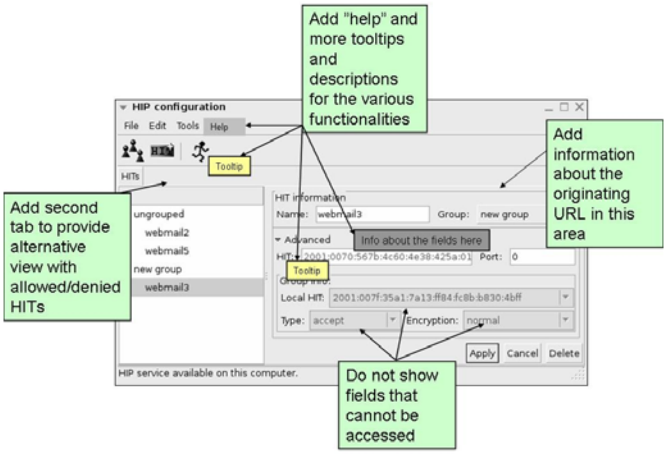
There was a clear difference between the two user groups. The group already experienced with HIP was more actively searching for the security indicators, whereas the other group only rarely noticed these indicators at all. Further, even group experienced with HIP sometimes failed to notice missing security indicators until logging in a webmail which had more security indicators. Only then would they realize that these indicators were not present in the previous webmail accounts they had logged into and considered secure.

There was also a clear gap between what users were actually noticing during the test about security indicators, and what they claimed to be searching for in online situations in real life. So, once again, there was a clear difference in what users claim to do with what they actually do which is typical in usability studies. This is why it is so important to observe users in action and not only rely on their reports of their own actions [28].

Overall, even if users seemed to have learned to look for indicators of security at least to some extent and were involved in online transactions, their attitudes, sources of information and amount of interest in security were surprisingly underdeveloped. The users claimed they would not mind if someone would gain access to their personal e- mails, since "they didn't have anything to hide" – a claim users often abandon once the privacy gets breached. Security was seen as a burden, and users were not really interested in it – they did not follow security news, and did not express worry about bad things happening to them.

V. DISCUSSION

The test setting was not very realistic: a laboratory environment with no disturbance; users were not using their own usernames and passwords. They also knew the test was about security. However, the usability tests revealed a number of areas of improvement in the UI (Figure 6).



ungrouped
webmail2
webmail5
new group
webmail3

HIT information
Name: webmail3
Group: new group
Advanced
HIT: 2001:0070:567b:4c60:4e38:425a:01
Port: 0
Group info:
Local HIT: 2001:007f:35a1:7a13:ff84:fc8b:b830:4bff
Type: accept
Encryption: normal

HIP service available on this computer.

Add "help" and more tooltips and descriptions for the various functionalities

Add information about the originating URL in this area

Add second tab to provide alternative view with allowed/denied HITs

Do not show fields that cannot be accessed

Figure 6. HIP GUI improvement items.

The UI was clearly too technical: users experienced it difficult to understand, "aimed for technical administrators", and most non-HIP users reported they wished they would prefer not manually handling the HITs at all. Further, **t**he traditional security indicators were not efficient: users didn't notice the changing color of the address bar (blue for HIP, yellow for SSL). The *absence* of the security indicators went unnoticed, too. Only the Firefox add-on displayed security indicators when HIP was used in "normal" mode. For the opportunistic mode, the add-on displayed IP addresses instead of HITs. This was the only way for the user to distinguish between the opportunistic and normal modes. However, users didn't notice the change, which means that the users did not notice that they were using leap-of-faith security. A user could have compared the HIT of the server displayed on the web page and noticed that it does not match to IP address displayed by the add-on, but he didn't. The lesson learned is that the lowered security used in opportunistic mode should be informed to the user at least in the prompt.

Overall, the UI concepts were difficult. Users were confused with the concepts of HIT and HIP and explicitly expressed a craving for more information. Even when they were able to learn that fingerprint and HIT were synonyms, the concept of a fingerprint or HIT itself was experienced to be difficult. Especially, differentiating between HITs of the local and peer host was very hard. Further, usage of grouping needs retouching: Users could imagine, when prompted, some possible uses for groups, such as grouping the HITs according to the service or context with which they would be used. However, they didn't at this point at least realize that the groups could be used for indicating which HITs were allowed and which were not. Better visualization of the allowed/denied dimension is probably needed for enhanced usability.

Users were looking for a help menu, and also wanted to have more tooltips and explanatory texts present in the UI. This is indicated that the UI was somewhat immature and technical. For the same reason, UI was seen as "administrative GUI". Further, users were frustrated of being shown fields that they could not access. Some users reported it made them realize how little they actually knew of the technology behind the GUI. In the next version, such fields must be either enabled or not shown to the users.

The familiarity aspect was important: Users liked the HIT announcement to the extent that it reminded them of other types of certificates they were familiar with. Users also expressed an explicit wish for the procedure to be similar to SSL.

Currently, in the navigation panel, there is only one view available for the HITs. However, there is probably need for more, alternating views to the same data. Users may want to organize the HITs according to contents and/or services they are related to, or according to when the HITs are in fact allowed or denied, to enhance personalization.

Further usability improvements include creating suitable icons for the HITs and adding keyboard shortcuts for advanced users in order to support multiple interaction methods.

VI. CONCLUSIONS

On basis of tests it is obvious that a lot of work still needs to be done for the HIP GUI to be truly usable. However, users were able to manage the created HITs with the prototype to the extent that they were able to create and remove groups, and have some idea how they could be used in practice. The identified usability improvements are straightforward to implement and would probably enhance the user-friendliness of the GUI to a great extent – something to be evaluated with another round of usability tests. The differences with the two user groups were relatively small, which may be indication that it is possible to please most users with just one GUI, instead of having several versions for various users.

HIP is based on low-layer IPsec mechanisms which may not be always visible especially to legacy network applications. In such a case, as complete automation may not be the best way to go as users tend to crave for visual confirmation and feedback for security taking place, *prompting* can be used to assure the user that the underlying communications are in fact secured. Alternatively, the client or server software can be modified to show security indicators to the user in a way that is likely to get noticed. We experimented with both of these approaches in this paper.

Our work has further corroborated that the current security indicators do not work. Existing research has shown that users are interested in security only as a secondary goal, as means to an end [11] and do not understand security information when it is provided for them [32]. Still, users may want to know more about security if it is easily available and provided in a way that is understandable [1].

References

[1] Adams, A., Sasse, M.A., Users are not the Enemy, CACM 1999.

[2] Cao, X. and Iverson, L. 2006. Intentional access management: making access control usable for end-users. Proc. of SOUPS '06, vol. 149. ACM Press, New York, NY, 20-31

[3] Grossklags, J., Thaw, D., Perzanowski, A., Mulligan, D., and Konstan, J. (2006), User Choices and Regret: Understanding Users' Decision Process about Consensually acquired Spyware, I/S: A Journal of Law and Policy for the Information Society, Vol. 2, No 2.

[4] Tsai, J., Egelman, S., Shipman, R., Pu, K-C.D., Cranor, L., Acquisti, A (2006), Symbols of Privacy. Poster Abstract.

[5] Liu, D., Asgharpour, F., Camp, L.J, Risk Communication in Computer Security Using Mental Models. Proc. of USEC07 & Financial Cryptography 2007, LNCS.

[6] P. Jokela, R. Moscowitz, Nikander, P. Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP), RFC5202, April 2008.

[7] Camp, L.J, Trust & Risk in Internet Commerce, MIT Press, (Cambridge, MA) (2000).

[8] Cyr, D., Head, M. and Ivanov, A. (2006). Design Aesthetics Leading to M-loyalty in Mobile Commerce. Information and Management.

[9] DeWitt, A. J. and Kuljis, J. 2006. Is usable security an oxymoron? Interactions 13, 3 (May. 2006), 41-44.

[10] Dhamija, R., Perrig, A., Deja Vu: A User Study. Using Images for Authentication. Proc. of the 9th USENIX Security Symposium, August 2000.

[11] Yee, K-P., Guidelines and Strategies for Secure Interaction Design, in Cranor, L.F & Garfinkel, S (Eds.): Security and Usability: Designing secure systems that people can use. O'Reilly Books (2005) 247-274.

[12] Whitten, A, Tygar, J.D (1999), Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. Proc. of the 8th USENIX Security Symposium, August 1999.

[13] Ecommerce Trust Study, Cheskin Research and Studio Archetype/Sapient. 31. http://www.cheskin.com (1999)

[14] Franco, R., Better Website Identification and Extended Validation Certificates in IE7 and Other Browsers. http://blogs.msdn.com/ie/archive/2005/11/21/495507.aspx, Nov. 21, 2005.

[15] Hartman, S., IETF Internet-Draft: Requirements for Web Authentication Resistant to Phishing. http://www.ietf.org/internet-drafts/draft-hartman-webauth-phishing-09.txt, August 2008.

[16] Henderson, T., Nikander, P., Komu M, RFC5338: Using HIP with Legacy Applications, Sep 2008.

[17] Touch, J., Black, D., Wang, Y-S., RFC5378: Problem and Applicability Statement or Better Than Nothing Security (BTNS), Nov 2008

[18] Karvonen, K., The Beauty of Simplicity. Proc. of the ACM Conference on Universal Usability (CUU 2000), November 16-17, 2000, Washington DC, USA

[19] Karvonen, K: Creating Trust, Proc. of the Fourth Nordic Workshop on Secure IT Systems (NordSec'99), November 1-2, 1999.

[20] Komu, M, Tarkoma, S, Kangasharju, J., Gurtov, A., Applying a Cryptographic Namespace to Applications, Proc. of the first ACM workshop on Dynamic Interconnection of Networks (DIN 2005)

[21] Komu et al, Basic HIP Extensions for the Traversal of Network Address Translators, Oct, 2008, Internet draft, work in progress

[22] Kuo, C., Perrig, A., Walker, J., Designing an evaluation method for security user interfaces: lessons from studying secure wireless network configuration. Interactions, 13(3):28-31, ACM Press, 2006.

[23] A. Gurtov, Host Identity Protocol (HIP): Towards the Secure Mobile Internet, ISBN 978-0-470-99790-1, Wiley and Sons, June 2008.

[24] Modadugu, N., Rescorla, E., The Design and Implementation of Datagram TLS, Proc. of NDSS 2004, February 2004

[25] Moskowitz, R., Nikander, P., Host Identity Protocol (HIP) Architecture, RFC 4423, May 2006.

[26] Moskowitz, R., Nikander, P. Jokela, P., Henderson, T., Host Identity Protocol, RFC 5201, April 2008.

[27] Nielsen, J, Usability Engineering, Academic Press, Boston 1993,

[28] Riegelsberger, J., Sasse, M.A., & McCarthy, J., The Researcher's Dilemma: Evaluating Trust in Computer Mediated Communications. International Journal of Human Computer Studies, Vol. 58, (2003) 759-781

[29] Staikos, G., Web Browser Developers Work Together on Security. http://dot.kde.org/1132619164/, Nov. 2005.

[30] Salam, A.F, Rao, H.R., and Pegels, C.C.: Consumer-Perceived Risk in E-Commerce Transactions. Comm. of The ACM, December 2003/Vol. 46, No. 12, (2003) 325-331

[31] Schechter, S., Dhamija, R., Ozment, A., Fischer, I., The Emperor's New Security Indicators, Proc. of the IEEE Symposium on Security and Privacy, May 2000.7