DEPARTMENT OF COMPUTER SCIENCE

IT-Security Seminar SS2019

Usable Security

Vitus Jonassen Wittke – its104137

Supervised by

Prof. Dr. Gerd Beuster – FH Wedel

Handed in on July 5, 2019

Outline

Figures

1. Introduction

With connectivity steadily rising [1], criminals are gaining more and more avenues to conduct their business. Due to the many systems involved in today's applications, thwarting the criminals attempts to attack certain systems or users has become a lot more complex. This complexity is usually handled by security researchers and experts. While their findings and solutions contribute greatly to an increase in overall security, they often neglect to consider the usability of their results [2] [3]. This lack of usability can lead to systems, that are secure in theory, being misconfigured or avoided altogether and thereby decreasing or nullifying their protective capabilities.

Usable Security is a research field in which usability for security solutions is examined. Usable Security research usually focuses on the usability of security software from the perspective of the average user. While this is an important consideration, the equally important usability considerations by experts are seldom part of such research [4]. As experts are usually tasked with implementing or enforcing certain security mechanisms, increased usability in regards to their needs, can lead to increased security for the average user.

This paper will firstly cover the trade-off between security and convenience. This will be followed by an overview of the three general lessons pertaining to usability by the average user that have been figured out over the years. The next and largest chapter will cover why usability for experts should be considered more often and what advice regarding usability for system administrators and software developers has been given by researchers. Lastly, a conclusion will focus on the main points brought up in this paper.

2. Security vs. Convenience

Security and Convenience are often in opposition to one another. Many security precautions restrict the user in what he or she can actually do and are difficult to set up properly. The average user will usually sacrifice security for convenience, because he or she does not fully understand the purpose and therefore the importance of the security precautions [2].

2.1. Trade-Off

As security and convenience are both important factors for most security solutions, a trade-off between the two has to be found [5]. Because the average user prioritises convenience over security, a security mechanism stands a better chance of being accepted by the average user if it offers value to him or her [6].

This is, of course, not the case if the mechanism is enforced by the application. If the enforced security mechanism is too strict however, the whole application runs the risk of being avoided by the average user altogether. If no security mechanisms are enforced or used by the average user, security critical online actions are not feasible anymore. The same is true for security mechanisms that are strictly enforced, but too complicated or inconvenient for the average user. Both scenarios lead to the average user reverting back to old analogous ways of doing security critical actions.

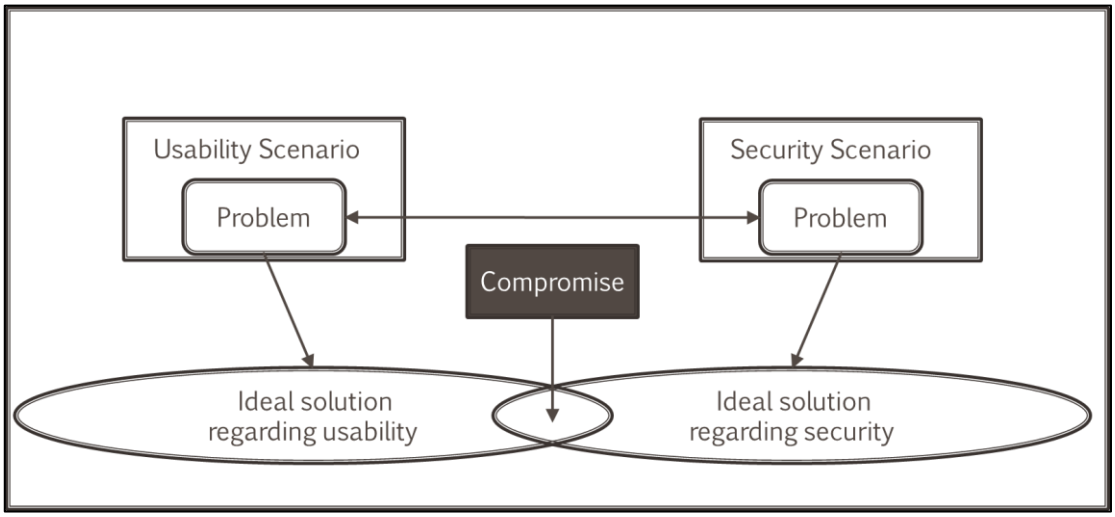

Figure 1: Trade-Off between usability and security based on [5]

Because convenience is a usability concern, usability can stand in opposition to security as well. As Figure 1 shows, the trade-off is usually a compromise between ideal solutions regarding usability and security. This compromise leads to a system that is neither ideal in regards to usability nor security, but offers enough convenience to the average user to be accepting of the security mechanisms.

2.2. Consensuses

While a simple trade-off between usability and security is acceptable in most cases, a consensus would be better. Some of such consensuses have been achieved.

The use of mobile TANs in banking systems is such a consensus. The user only has to have his cell phone available to confirm transactions. As most users have their cell phone around at all times and it does not suffer from the same problems as printed TAN-Lists, it offers a great deal of usability to the user. By introducing a second authentication factor, the security of the whole system increases immensely. The success of mobile TAN-Systems is confirmed by looking at the decreased amount of successful phishing or man-in-the-middle attacks on banking sites since that system was widely adopted [7, p. 76].

Another successful consensus is the use of biometric factors for authentication purposes. The user does not have to remember a password or -code anymore and the credentials are less likely to be stolen, as they are features of the person trying to authenticate themselves [8].

3. Lessons Learned

Over several years of research in the field of Usable Security, a few lessons regarding Usable Security for the average user have been gathered [4]. The three key lessons will be the topic of this chapter.

3.1. Do not assume

As people developing security solutions are usually experts in the field, there is little understanding for what the average user of the security solution might know about the specifics involved [9]. It is also commonly expected that the average user cares as much about certain security features as the people developing them do [10]. Both of those factors lead to false assumptions about the user, that are reflected in a negative user experience of the security solutions.

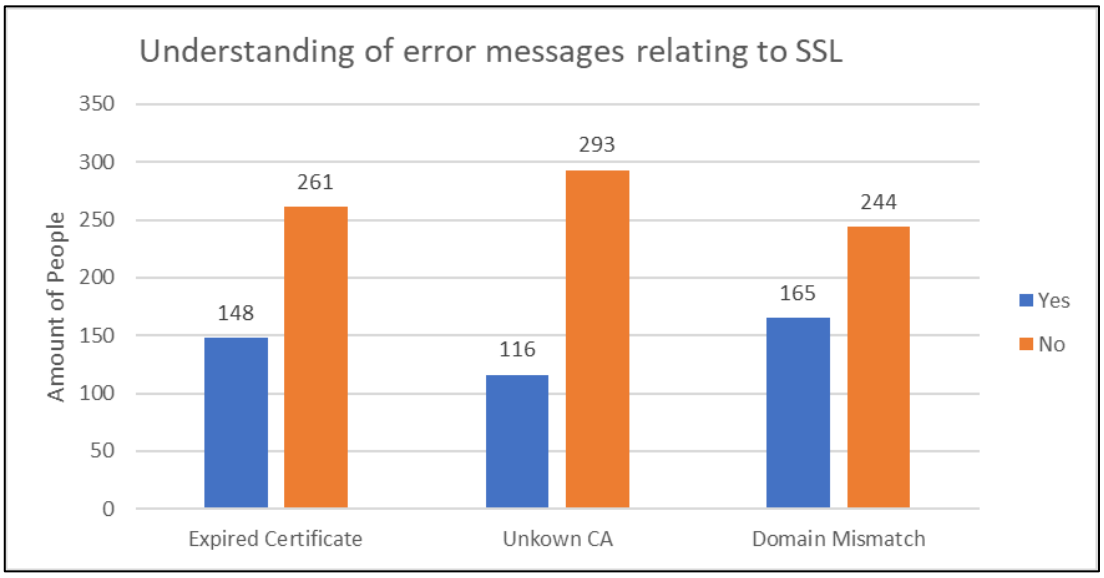

Figure 2: Understanding of error messages relating to SSL based on [11]

Figure 2 shows the results of a study evaluating how people react to certain SSL warning messages [11]. As clearly shown in the figure, most people do not even understand what the error messages mean. This is hardly surprising, as specific domain knowledge is required to know what an SSL-certificate is and what those different presented cases actually mean.

An approach to mitigate this problem is to specifically consider the entirely different perspective of the average user of the solution instead of making assumptions. This approach can yield promising results, that ultimately increase security for the average user.

3.2. Do not bother

As mentioned in the previous chapter, the average user considers security as less important than performing the task they set out to do. When security gets in their way while trying to do their main task, it is perceived as an annoyance, that has to be worked around or ignored entirely [12]. This can lead to the average users developing a habit of very insecure practices [2], such as using an only slightly altered password when asked to change their old password too often.

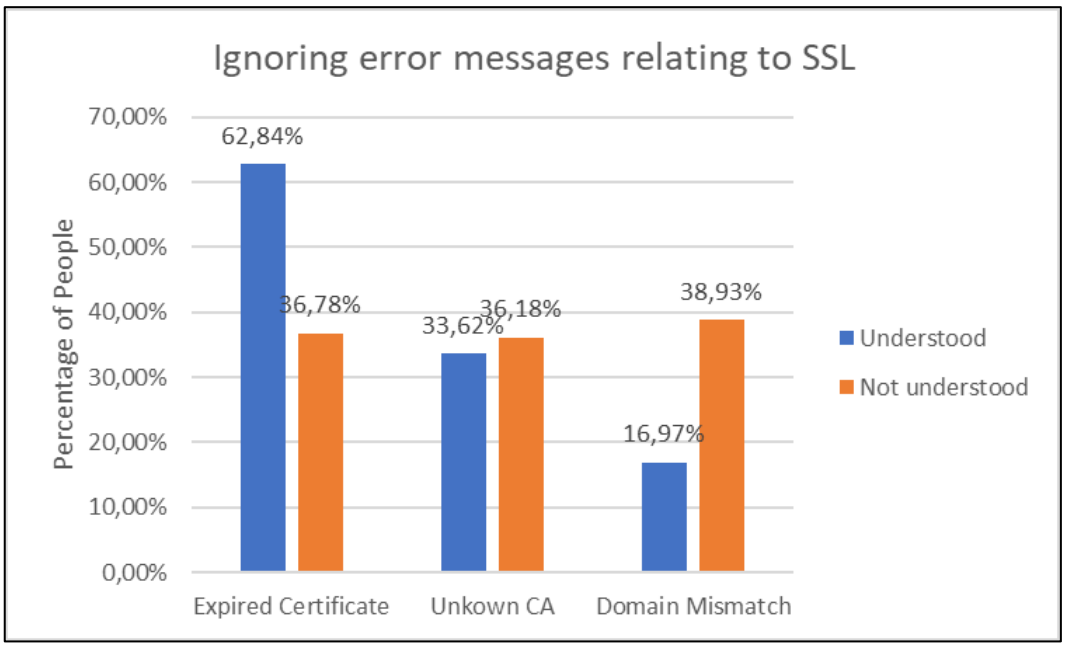

Figure 3: Ignoring error messages relating to SSL based on [11]

As shown in Figure 3, a significant number of users tended to ignore the SSL error messages. Alarmingly, many more users who actually understand what an expired SSL-certificate means, chose to ignore the warning relating to it. It is likely that those percentages could be a lot higher in the real world, as this survey was conducted as part of a study that only required the participants to answer the questions to get the gift certificate offered as an incentive. Real users would probably have a stronger motivation to actually proceed to the website, as they have something they actually want to achieve on those websites.

The most common approach to mitigate this problem is to not involve the user in as many decisions regarding security as possible [13]. This is achieved by methods like making updates automatic or using secure default configurations. Should not involving the user at all not be possible, nudging them in the right direction through opinionated design also leads to good results [14].

3.3. Do not overwhelm

Research has shown that the average user reacts negatively when confronted with an overabundance of advice or recommendations regarding security. They feel overwhelmed, as they have to first choose which advice or recommendation to listen to, before actually trying to understand and follow that advice or recommendation [15] [16]. This might make them view improving their security as too arduous and ultimately lead to them giving up on it.

Another problem is the desensitisation towards certain security issues, if the user encountered too many warning messages pertaining to that security issue which never lead to any noticeable harm for the user. Those warning messages will be seen as normal and consequently ignored [11].

The approach of mitigating this issue is to carefully consider which information should be presented to the user and which information should be kept from him or her. While this approach might sound similar to the one presented in the prior chapter, it is not about not involving the user, but about trying to present information to him or her efficiently and in line with their demands.

4. Usable Security for Experts

Research in the field of Usable Security usually only considers the perspective of the average user. The specific needs of experts are often not considered.

4.1. Importance

Experts, like software developers and system administrators are the people tasked with implementing or enforcing security mechanisms. Every mistake they make in that regard affect the users of their system negatively. It is therefore beneficial to enable them to make better decisions by improving the usability of the security mechanisms they are working with [4].

The limited research in the field of Usable Security for experts has shown that the experts considered are usually no security experts and face similar problems to the average user when dealing with security. Identifying a few lessons for specific groups of experts, similar to the lessons pertaining to the average user, is therefore an important step in improving overall security.

4.2. Security Management Systems

Most software that is used as part of security management systems is neither focused on security nor offers usable interfaces that communicate information clearly and efficiently [17]. With such software time is often spend to reduce the amount of data the system administrator has to analyse, but little effort is made to present that data efficiently on a usable interface.

It is to mention however that designing an interface that meets such criteria is harder than in many other fields, due to the nature of the system and the many different vulnerabilities that have to be kept in mind. As it is an attacker's intention to avoid detection, each new attack will be somewhat different and require different sets of information to detect it. It is therefore a challenge to provide the system administrator with all the information he or she needs without overwhelming them.

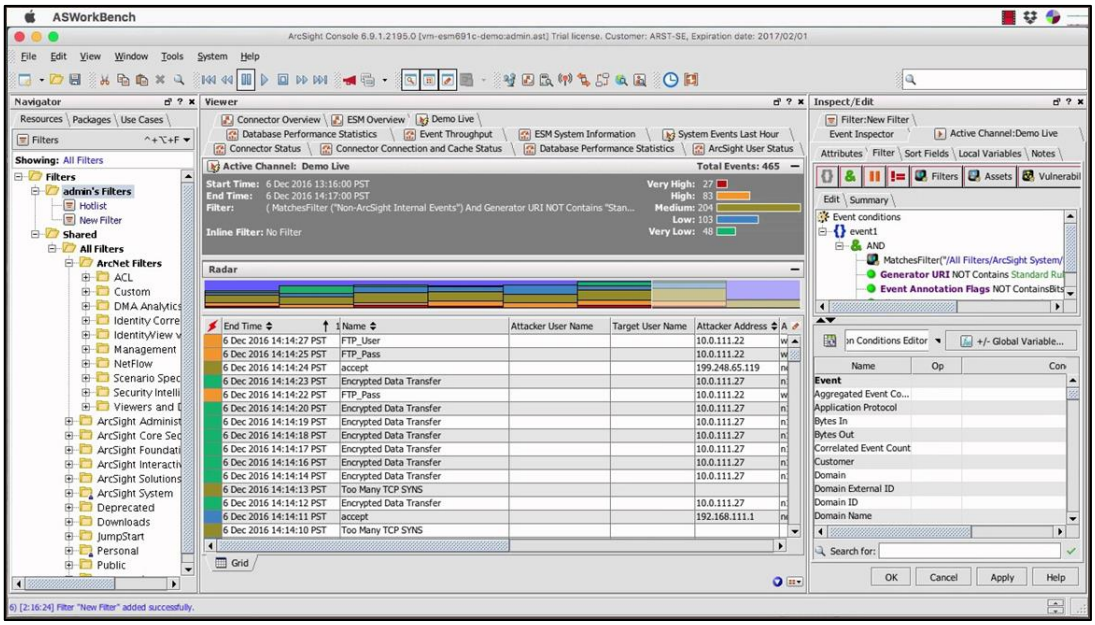

Figure 4: Screenshot of an ArcSight Console from [18]

Figure 4 shows a typical, yet slightly outdated, interface for such security software. The interface offers the user lots of information and shows a host of options to access other parts of the software all at once. This can easily overwhelm the system administrator.

Current interfaces assume the system administrator knows what to look for and where to find it. If he or she is not able to find the required information or is presented with misleading information, they might not be able to detect or analyse potential attacks. This leads to security issues for the whole system they are tasked to supervise.

A few design approaches for creating better software as part of security management systems have been outlined in the paper “Even Experts Deserve Usable Security: Design guidelines for security management systems” [17].

The first approach is that of general Usable Security. This approach suggests that users require a mental model of how the system works [19]. This model is formed by interactions with the system and its interface. To be able to form an accurate mental model of the system, the interface has to be comfortable to the user, make them aware what they have to do, how it is done and when they have successfully done it. It should also prevent the user from making dangerous errors and give enough information to the user, so that he or she is able to determine the current state of the system.

The second approach is Ecological Interface Design, which suggests to structure the interface according to a five-level abstraction hierarchy with each level becoming more detailed than the level prior [20]. The topmost level gives a general overview of the system, while the lowest level provides a representation of the actual physical layout of the system. Users can work at the level that is required for performing their tasks and switch to a higher level if they require a more general overview or a lower one if they require more detail. This hierarchical approach to presenting information to the user also helps in building an accurate mental model of the system.

Social Navigation is the third approach that should be considered, as people have a natural tendency to consult other people before making decisions for themselves [21]. In the case of system administrators this often means consulting other people through mailing lists or online forums. Instead of relying on other people to communicate their problems, solutions and approaches through other media, it would be beneficial to enable such communication through the interface of the security management software. This communication channel should include methods to directly compare events or configurations with other users of the software to improve the user's ability make comparisons. Another benefit would be the ability to suggest certain behaviours based on what the majority of users or trusted experts did.

The last approach is called Persuasive Technology and focuses on ways to influence the user such that he or she behaves in a desired way [22]. One such way of influencing the user, is to make the desired path of actions for the user to take the one that is easiest to do. This would directly translate into making the actions that result in the greatest amount of security the easiest to do for the system administrator. According to this approach the software should also suggest that it did him or her a favour, because humans tend to return favours. Other ways of influencing the user are signalling to him or her that the software is competent and to reward his or her behaviour if it is aligning what is desirable.

Another aspect, that is not directly covered by the four design approaches, but covered in the paper [17], is giving the system administrators the ability to revert the state of the system to a previous one, if one of their decisions leads to an undesired state of the system.

Considering these approaches can lead to security management systems that are better in supporting the system administrator in performing their main task and thereby making him or her more efficient. This can be of great benefit to the general users of that system.

4.3. Developers

Software developers are the people tasked with implementing security mechanisms that seek to protect the user. As such their ability to properly use and understand security solutions greatly affects the average user [4]. While they regularly use security mechanisms, they are usually no experts regarding them. They suffer from many of the same problems that the average user does when it comes to the usability of security mechanisms, while not being considered as heavily in Usable Security research.

Most of the lessons given in the third chapter can be easily applied to them. They usually do not understand as much about security as security experts assume they do. Security is often not their main concern, as that role is taken by things like improving functionality. They also tend to struggle with being presented with an overabundance of advice and best practices regarding secure development.

The paper "You Are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users" [4] seeks to provide a few guidelines for improving the usability of security mechanisms for software developers.

4.3.1. Improve API Usability

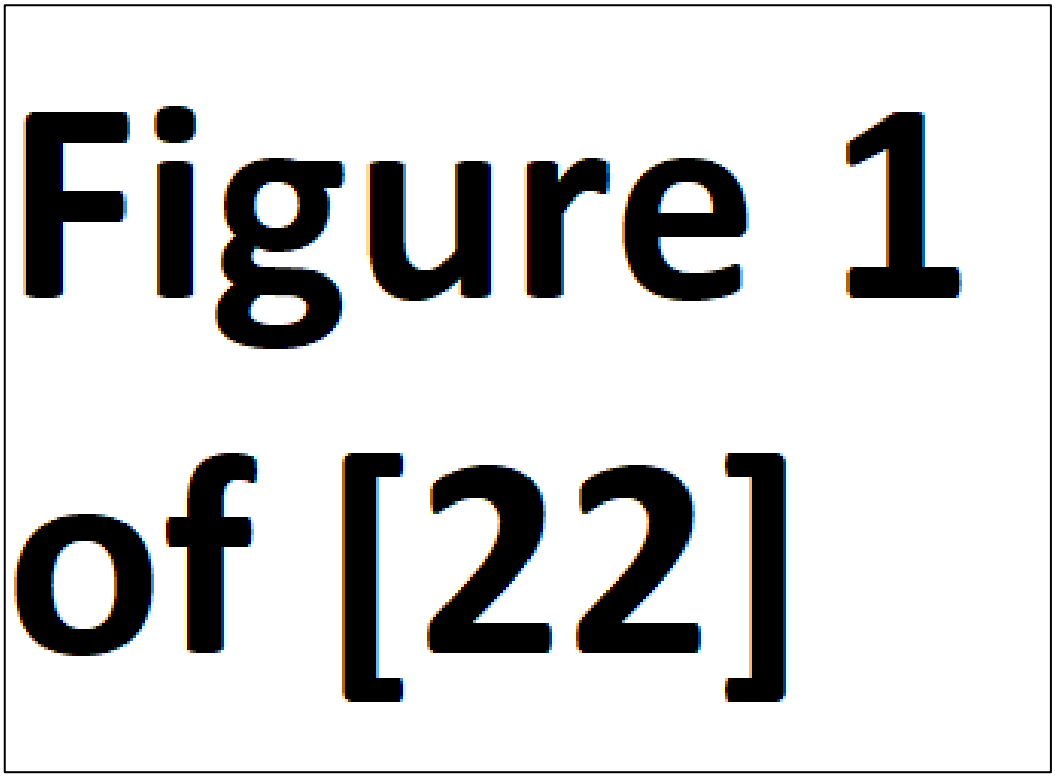

Figure 5:Number of applications that violate different encryption rules from [23]

As research has shown, developers tend to use security APIs improperly. Figure 5 shows the result of such a study, which evaluated 11,748 different Android applications for misconfigurations of cryptographic libraries [23]. It can be seen that it is more common for the applications investigated to violate one or two rules than it is to violate none. This shows that the developers of those applications weren't able to use the APIs provided by the cryptographic libraries properly.

There are already general guidelines for designing usable APIs and some considerations by security researchers regarding the usability of security APIs. These sources should be combined to create a set of guidelines on how to design usable security APIs. If this set of guidelines is followed when developing new security APIs or modifying old ones, the usability of those APIs should improve significantly.

These guidelines include recommendations like naming methods and classes of the API in a fashion a user would reasonably expect [24], including security functionality in regular APIs, so developers don't have to use specific security APIs, and choosing safe default values [25], so misconfigurations have to be done deliberately. Proper documentation and warnings, that notify the user of an insecure configuration, are also among those recommendations.

4.3.2. Secure, usable information resources

When developers search for help in using security mechanisms, they often use the quick but insecure fixes provided by the resource they turned to.

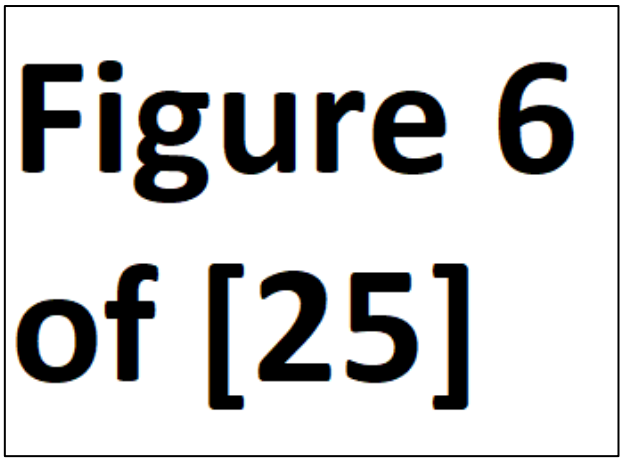

Figure 6: Participants opinion about the resources used from [26]

As Figure 6 shows, most of the participants in the study [26], rate Stack Overflow as the resource, that is easiest to use and also the most helpful, but also as less correct than the official documentation or books on the topic. This shows that the most usable resource available to the developers is significantly less correct than official resources.

To remedy this, the official documentation has to become more usable and interactive. As there is a natural tendency to consult other people before making decisions, which was briefly discussed in chapter 4.2, usable resources like Stack Overflow should be monitored more closely to be able to dissuade users from using bad advice.

4.3.3. Developer tool support

If developers could integrate tools into their development environments, that help them make more secure decisions by raising the awareness for security and providing feedback regarding security, the process of developing more secure applications could be eased significantly. While many tools exist, that make the development process faster and easier for the developer, such security tools are barely used.

4.3.4. Taking developers out of the loop

As discussed, developers are neither experts for security nor is it their main focus when developing applications. Because of that, it can be beneficial to follow the lesson given in chapter 3.2 and to involve the developers as little as possible when it comes to handling security.

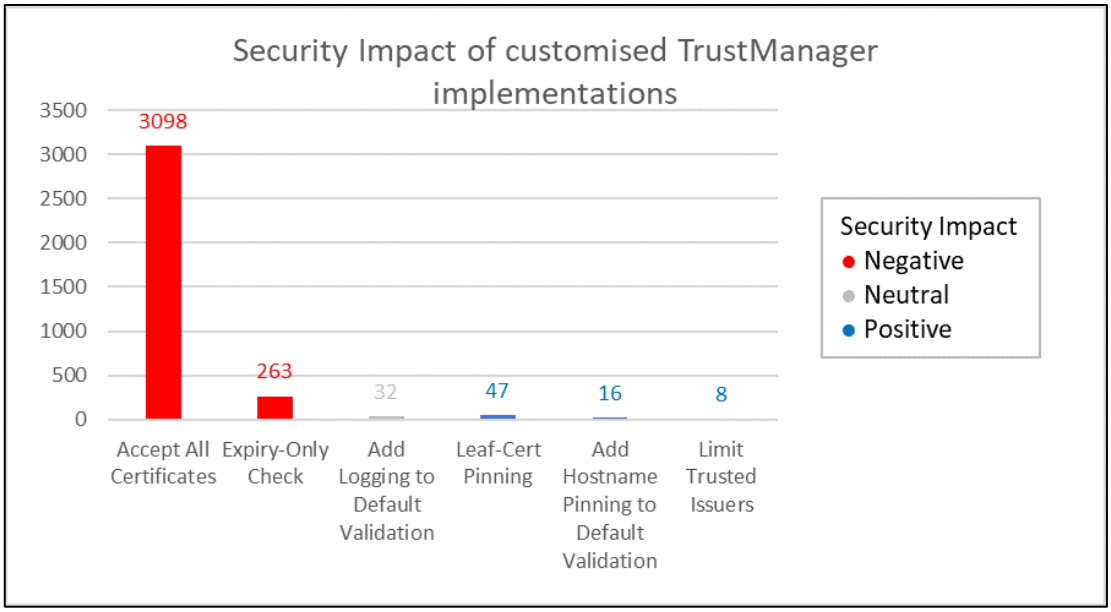

Figure 7: Security Impact of customised TrustManager implementations based on [27]

Figure 7 shows the result an evaluation that was conducted on 3,464 different Android applications, that implemented a custom TrustManager [27]. The results clearly show that a majority of developers implemented it in a way that weakened security when compared to the default implementation. Only a small minority of developers were able to improve on the default implementation.

It is therefore recommended to move security management and security-critical decisions to the OS and away from the application themselves, if it can be considered generally beneficial in that specific case. It should be evaluated which cases could benefit from such a measure and how this move can be done without restricting the developers too much.

5. Conclusion

The field of Usable Security has the tension between its two namesakes, namely Usability and Security, in its core. While some remarkable consensuses regarding those two sometimes opposite goals have been reached, there is still usually a trade-off between the two. Even those consensuses are essentially compromises, as they are less convenient and less secure than they could be, if only one of those goals would have been considered.

Even though the field of Usable Security is a relatively young one with its roughly 20 years, a few key lessons for improving Usable Security for the average user have been discovered. When those lessons are considered, the security solutions developed for the average user are more secure than they would have been without those lessons, while still being convenient for the user.

While those key lessons for average users are important, another very important group of users of security solutions has almost been neglected by Usable Security research. This group consists of people considered experts regarding IT, for example system administrators and software developers. They need usable security solutions, that enable them to enforce or develop security mechanisms that help to increase the security of their users.

Some lessons for improving the Usable Security for members of this group, have been pointed out in this paper, but many aspects of them still require additional research. This research is worthwhile because it will ultimately have a significant positive impact on the security of different applications as a whole.

6. References

[1] S. Lucero, “IoT platforms: enabling the Internet of Things,” March 2016. [Online]. Available: https://cdn.ihs.com/www/pdf/enabling-IOT.pdf. [Accessed 15 June 2019].

[2] A. Adams and M. A. Sasse, “Users are not the enemy,” *Communications of the ACM Volume 42 Issue 12,* pp. 40-46, December 1999. Available: http://discovery.ucl.ac.uk/20247/2/CACM%20FINAL.pdf. [Accessed 2 June 2019].

[3] M. F. Theofanos and S. L. Pfleeger, “Shouldn't All Security Be Usable?,” *IEEE Security & Privacy Volume 9 Issue 2,* pp. 12-17, March-April 2011. Available: https://ieeexplore.ieee.org/document/5739638. [Accessed 16 June 2019].

[4] Y. Acar, S. Fahl and M. L. Mazurek, “You are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users,” in *2016 IEEE Cybersecurity Development (SecDev)*, Boston, MA, USA, 2016. Available: https://saschafahl.de/papers/researchagenda2016.pdf. [Accessed 11 June 2019].

[5] C. Braz, A. Seffah und D. M'Raihi, „Designing a Trade-Off Between Usability and Security: A Metrics Based-Model,“ in *Human-Computer Interaction – INTERACT 2007: 11th IFIP TC 13 International Conference, Proceedings, Part II*, Rio de Janeiro, Brazil, 2007. Available: https://www.researchgate.net/publication/225680299_Designing_a_Trade-Off_Between_Usability_and_Security_A_Metrics_Based-Model. [Accessed 16 June 2019].

[6] P. Gutmann and I. Grigg, “Security Usability,” *IEEE Security and Privacy Volume 3 Issue 4,* pp. 56-58, July 2005. Available: https://ieeexplore.ieee.org/document/1492344. [Accessed 11 June 2019].

[7] E. Markatos und D. Balzarotti, The Red Book: A Roadmap for Systems Security Research, The SysSec Consortium, 2013.

[8] F. L. Podio, *Biometrics - Technologies for Highly Secure Personal Authentication,* NIST, 2001. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151516. [Accessed 15 June 2019].

[9] A. Whitten and J. D. Tygar, “Why Johnny Can’t Encrypt,” *Proceedings of the 8th USENIX Security Symposium,* pp. 169-183, August 1999. Available: https://people.eecs.berkeley.edu/~tygar/papers/Why_Johnny_Cant_Encrypt/OReilly.pdf. [Accessed 11 June 2019].

[10] A. Mathur, J. Engel, S. Sobti, V. Chang und M. Chetty, „They keep coming back like zombies,“ *SOUPS '16 Proceedings of the Twelfth USENIX Conference on Usable Privacy and Security,* pp. 43-58, June 2016. Available: https://www.usenix.org/system/files/conference/soups2016/soups2016-paper-mathur.pdf. [Accessed 12 June 2019].

[11] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri and L. F. Cranor, “Crying wolf: an empirical study of SSL warning effectiveness,” *SSYM'09 Proceedings of the 18th conference on USENIX security symposium,* pp. 399-416, August 2009. Available: https://www.usenix.org/legacy/event/sec09/tech/full_papers/sunshine.pdf. [Accessed 12 June 2019].

[12] J. Lee and L. Bauer, “Studying the Effectiveness of Security Images in Internet Banking,” *IEEE Internet Computing Volume 19 Issue 1,* pp. 54-62, January-February 2015. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.644.363&rep=rep1&type=pdf. [Accessed 11 June 2019].

[13] L. F. Cranor, “A framework for reasoning about the human in the loop,” *UPSEC'08 Proceedings of the 1st Conference on Usability, Psychology, and Security,* pp. 1-15, April 2008. Available: https://www.usenix.org/legacy/event/upsec08/tech/full_papers/cranor/cranor.pdf. [Accessed 19 June 2019].

[14] A. P. Felt, A. Ainslie, R. W. Reeder, S. Consolvo, S. Thyagaraja, A. Bettes, H. Harris and J. Grimes, “Improving SSL Warnings: Comprehension and Adherence,” *CHI '15 Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems,* pp. 2893-2902, April 2015. Available: https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43265.pdf. [Accessed 15 June 2019].

[15] C. Herley, “More Is Not the Answer,” *IEEE Security & Privacy Volume 12 Issue 1,* pp. 14-19, November 2013. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MoreIsNotTheAnswer.pdf. [Accessed 12 June 2019].

[16] E. M. Redmiles, A. R. Malone and M. L. Mazurek, “I Think They're Trying to Tell Me Something: Advice Sources and Selection for Digital Security,” *2016 IEEE Symposium on Security and Privacy (SP),* pp. 272-288, May 2016. Available: https://drum.lib.umd.edu/bitstream/handle/1903/17328/CS-TR-5048.pdf?sequence=1&isAllowed=y. [Accessed 13 June 2019].

[17] S. Chiasson, R. Biddle and A. Somayaji, “Even Experts Deserve Usable Security: Design guidelines for security management systems,” *SOUPS Workshop on Usable IT Security Management (USM),* pp. 1-4, 2007. Available: https://pdfs.semanticscholar.org/1069/93f734c41155873d7a9873755a3413d4ba1a.pdf. [Accessed 13 June 2019].

[18] P. Brettle, „ArcSight Console training - Part 1,“ 20 December 2016. [Online]. Available: https://www.youtube.com/watch?v=N7J0EwdbKF0. [Accessed 13 June 2019].

[19] D. Norman, The design of everyday things: Revised and expanded edition, Basic books, 2013. Available: http://www.nixdell.com/classes/HCI-and-Design-Spring-2017/The-Design-of-Everyday-Things-Revised-and-Expanded-Edition.pdf. [Accessed 15 June 2019].

[20] K. Vicente and J. Rasmussen, “Ecological interface design: theoretical foundations,” *IEEE Transactions on Systems, Man, and Cybernetics Volume 22 Issue 4,* pp. 589-606, July-August 1992. Available: http://www.realtechsupport.org/UB/I2C/Ecologicalinterfacedesign_1992.pdf. [Accessed 15 June 2019].

[21] A. Dieberger, P. Dourish, K. Höök, P. Resnick and A. Wexelblat, “Social navigation: techniques for building more usable systems,” *Interactions Volume 7 Issue 6,* pp. 36-45, December 2000. Available: https://www.researchgate.net/publication/234815601_Social_navigation_Techniques_for_building_more_usable_systems. [Accessed 14 June 2019].

[22] B. Fogg and G. E. Fogg, Persuasive Technology: Using Computers to Change What We Think and Do, Morgan Kaufmann, 2003.

[23] M. Egele, D. Brumley, Y. Fratantonio and C. Kruegel, “An empirical study of cryptographic misuse in android applications,” *CCS '13 Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security,* pp. 73-84, November 2013. Available: https://sites.cs.ucsb.edu/~chris/research/doc/ccs13_cryptolint.pdf. [Accessed 15 June 2019].

[24] B. A. Myers and J. Stylos, “Improving API usability,” *Communications of the ACM Volume 59 Issue 6,* pp. 62-69, June 2016. Available: https://www.cs.cmu.edu/~NatProg/papers/API_Usability_Article_submitted.pdf. [Accessed 15 June 2019].

[25] M. Green und M. Smith, „Developers are Not the Enemy!: The Need for Usable Security APIs,“ *IEEE Security & Privacy Volume 14 Issue 5,* pp. 40-46, September-October 2016. Available: http://mattsmith.de/pdfs/DevelopersAreNotTheEnemy.pdf. [Accessed 11 June 2019].

[26] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek and C. Stransky, “You Get Where You're Looking for: The Impact of Information Sources on Code Security,” *2016 IEEE Symposium on Security and Privacy,* pp. 289-305, May 2016. Available: https://www.cs.umd.edu/class/fall2017/cmsc818O/papers/get-where-look.pdf. [Accessed 11 June 2019].

[27] S. Fahl, M. Harbach, H. Perl, M. Koetter and M. Smith, “Rethinking SSL development in an appified world,” *CCS '13 Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security,* pp. 49-60, November 2013. Available: http://cm.1-s.es/Cryptome-update-14-0812/2014/08/rethinking-ssl.pdf. [Accessed 15 June 2019].