Web Security and Privacy for Novices – Part 1

A Pattern Collection and Two Meta-Patterns

Alexander G. Mirnig*, Artur Lupp*, Alexander Meschtscherjakov* and Manfred Tscheligi†
*Center for Human-Computer Interaction
University of Salzburg, Salzburg, Austria
Email: firstname.lastname@sbg.ac.at
†Center for Human-Computer Interaction &
Austrian Institute Of Technology, Salzburg & Vienna, Austria
Email: firstname.lastname@sbg.ac.at

***Abstract*—Fostering security and privacy in online interactions is important for developers, providers, and users. Since a substantial amount of content on the web today is developed by nonprofessionals (e.g., teenagers, small company owners, hobbyists, etc.), it is important to provide development guidance that is suitable for these user groups. In this paper, we present two meta-patterns intended for novice web developers. The patterns intend to address some of the most common issues pertaining to privacy and security in websites and are intended to guide the reader through the development process from a privacy- and security-centered perspective.**

Keywords–Patterns; On-line; Security; Privacy; Novice Users.

I. INTRODUCTION

From its rather humble beginnings in the nineties of the last century [1], the world wide web boasts a high number of websites today, with more and more being created and put online every year. According to Internet Live Stats [2], there were 1,766,926,408 websites online in 2017, which is an increase of 69% to 2016, where 1,045,00,534,808 had been counted. Otherwise, the average number of users per website has been consistently decreasing from 24 in 2000, to 9.1 in 2008, to 3.7 in 2015. This simultaneous increase and decrease is, in large part, due to a growing number of *inactive* websites, such as parked domains (i.e., the domain is reserved but not used) or expired sites. The latter often occurs when a website is abandoned by its owner, be it due to a lack of resources, interest, or both. And such abandoned websites are only one of many potential risks for unsuspecting visitors, as they are likely to be out of date and, therefore, more susceptible to injections or other attacks.

Whatever an individual's motivation might be in the end, it can be assumed that many individuals, who are not an experts in general web design, security, or privacy, have basic capabilities and desire to set up their own web presence. At the same time, these individuals may or may not be interested in improving their craft and eventually attain such expert knowledge. While the aforementioned teenager might decide to take the path of a professional web designer later in their life, he/she might just as well lose interest and abandon the website. The small-size company web developer might be a capable individual with a strong interest in web design, or they might just as well be the only available person to do what needs to be done (i.e., the only one possessing basic computer skills and a 10-year old web development handbook) without any further aspirations in this regard.

Available literature should reflect this difference in demographics and the reality of widespread web access - both from a user and developer standpoint. The aim should not be to replace standard literature but instead provide a resource containing the essentials for beginners, who are not necessarily students aiming at attaining an expert level eventually. What these individuals should have access to is an *essential minimum knowledge* about web design for privacy and security, in order to endanger neither themselves nor others who visit their sites or use their services.

This paper constitutes the first part of a series of three thematically connected papers, which describe patterns from the same pattern collection. The contribution of this paper is a set of two meta-patterns, which describe common security and privacy issues and how they can be addressed on a high level. Papers two and three will contain sets of patterns that address concrete issues (e.g., frequency of backups, data protection compliance, etc.). In this paper, we first outline relevant related work from the domains of web privacy, web security, and design patterns in Section II. We then describe the problem mining and pattern writing process in Section III. Section IV contains the two meta-patterns. In Section V, we discuss general aspects regarding ease of access of security-critical information for novices and conclude the paper in Section VI.

II. RELATED WORK

In the following, we provide a brief background on privacy and security in relation to novice users, together with an introduction to design patterns. Security and privacy are often considered to be inter-related, where an appropriately secure environment protects an individual's privacy acting within it. In the context of the Internet, both are connected to the information in relation to the user. Microsoft [3] defines both concepts as follows: "*Information privacy* refers to the user's ability to control when, how, and to what extent information about themselves will be collected, used, and shared with others. *Information security* refers to the ability of businesses and individuals to secure their computers from vulnerabilities and maintain the integrity of the stored information." When brought into relation with usability, *usable privacy* and *usable security* [4] refer, broadly speaking, to the ease (or difficulty) to interact with or implement a solution that fosters privacy, security, or both. This ease or difficulty is relative to an individual's level of expertise, which is why usability must be considered with the prospective user and their capabilities in mind. For the purposes of this paper, we focus on novice

 ISBN: 978-1-61208-708-5

or layman web developers.

A. Layman knowledge about Privacy and Security

A crucial aspect of security decisions regarding home computers and websites, is the existing knowledge about computers, the internet and their security issues. LaRose et al. [5] found that more knowledge about general computer security issues correlates frequently with the intention to behave securely. Their results also show that people who agree on the statement that online safety is their personal responsibility, are more likely to protect themselves compared to those who do not agree. Generally, people familiar with common security measures are more likely to engage in security behaviors [6]. In comparison, Shillair et al. [7] did not find any correlation between expert and layman knowledge regarding security measures. Even though knowledge is a very important factor when making decisions regarding security and privacy, additional motivations are needed for people to tackle security decisions efficiently [8], [9].

One motivation could be the protection of important information, such as online banking passwords. Internet applications often provide guidance when creating a password. Users tend to reuse passwords across most of their accounts once a user needs to manage a larger number of password [10]. A countermeasure could be a password creation process provided by an application. On the one hand, this could make the password creation process easier. On the other hand, the result may be weaker passwords. Shay et al. [11] suggested that service providers should present password requirements with additional (visual) feedback to increase usability, carefully considering the representation of feedback and guidance.

B. Design Patterns in Software Engineering and Related Domains

Originally conceived by Christopher Alexander to capture solutions in Architecture [12], [13], the pattern approach was later adopted by the computer science community and adapted to capture problem solutions in software engineering [14], [15]. The pattern collection by Gamma et al. [16] (also known as the "Gang of Four", or "GoF" for short) is probably still the best known contemporary pattern collection for software engineering, and can, at the same time, be considered one of the fundamental pieces of modern pattern literature, as it lays out a basis for pattern elements and structure along with the actual patterns themselves.

Design patterns are structured documentations of solutions to reoccurring problems. Since individual problems are usually parts of larger problems, patterns are often collected in pattern collections, which are also referred to as *pattern languages*, which dates back to Alexander's original use of the term [13]. Patterns can occur on different levels of abstraction and are referred to as *high-* or *low-level* patterns [17], depending on whether they describe a high- or low-level problem. Patterns on the highest level of abstraction are also referred to as *meta-patterns*.

Patterns have been adopted by various disciplines [18], among them Human-Computer Interaction (HCI) (e.g., [19], [20]) and Interaction Design (e.g., [17], [21]). Munoz-Arteaga et al. [22] proposed a pattern-based methodology for information security feedback design, which is, like most domain-specific literature, intended for advanced users. According to Vlissides [23], one key attribute of patterns is that they can make expertise accessible to non-experts. Bach et al. [24] make use of this feature in their design patterns for data comics, where data-related information is communicated in an easy to read comic-format.

III. Pattern Generation and Structure

The Pattern generation process began with an interview-based problem mining process in order to address issues with a high degree of relevancy for online privacy and/or security. The pattern writing was conducted by an HCI expert with experienced in writing patterns and who had also been involved in the problem mining. The pattern contents are based on an internal state-of-the art containing guidelines [25]–[27], topic relevant scientific publications (primarily ACM, Springer, and IEEE), and information gained from the interview protocols.

The pattern format was adapted from Mirnig et al. 2016 [28]. This structure was chosen due to its relative simplicity, which should make it easier for novice readers to comprehend the pattern contents. It looks as follows:

- *Name*: A short and descriptive name describing the solution
- *Intent*: A short paragraph intended to allow the reader decide whether or not the solution applies to the context in question
- *Problem*: The problem statement
- *Scenario* One example of a suitable application context
- *Solution*: The solution description
- *Example*: At least one descriptive example of the solution
- *References*: To source the solution and provide access to more in-depth resources, where available
- *Keywords*: Intended to help structure the pattern collection

The finished initial versions then underwent one iteration workshop with two HCI researchers and two web developers, in which each pattern was rated, adapting the approach proposed by Wurhofer et al. [29], Krischkowsky et al. [30], and Mirnig et al. [28], [31]. Each pattern was rated individually for each of its subcategories (Name, Intent, etc.) and then discussed in plenum. The result was a collection of 16 patterns. The process, and resulting pattern structure are described in more detail in Mirnig et al. 2019 [32].

IV. Patterns

In the following, we present the two meta-patterns on aspects that contributes to the general security of a website as well as how to evaluate it.

A. What contributes to the security of a website?

Intent: This Pattern lists various points contributing to the security of a website. Apart from that, it also provides methods that may be used to secure your own website.

Problem Statement: The amount of cyber-attacks on websites has increased over the last few years. The main targets are especially webshops and websites dealing with user data, for example forums or service websites.

Scenario: Setting up a website without thinking about security issues is negligent. An unprotected website is a security risk not only to yourself, but for all visitors of that site.

 ISBN: 978-1-61208-708-5

Solution: To quote Sun Tzu, ”If you know the enemy and know yourself, you need not feat the result of a hundred battles” [33]. Thus, if you know of potential dangers and common security issues in the web world, you can protect yourself more efficiently. The following list introduces some measurements that can be used to increase the security of your website.

- Access the web server only with a safe and secure computer.
  - A computer system can be considered as safe and secure, if the operating system as well as all the installed applications are up-to-date. Especially security updates play a very important role in this case.
- Dont plug in hardware from third parties to any system without checking them first.
  - External storage devices can contain malware or viruses that may infect a system. Thus, they have to be checked by anti-virus software before using them on important devices.
- Limit the amount of admin accounts.
  - Admin accounts usually have rights to access almost everything on a system. Therefore, only experts should have access to them.
  - Limit the amount of admin accounts to a bare minimum to minimize abuse.
- Use safe and secure admin passwords.
  - Studies show that longer passwords (10 characters or more) are more secure in comparison to shorter ones, even when the shorter passwords contain special characters or symbols.
- Usage of two-factor authentication.
  - If a cyber-criminal is able to get hands on an admin-password a two-factor authentication can provide an additional safety barrier.
- Set up user groups and manage the access rights.
  - Reading and writing privileges for users on a web server should be managed by administrators. This way important or system-relevant files can be protected against access and manipulation from unauthorized persons.
- Keep yourself up-to-date!
  - Inform yourself regularly about the latest web-security issues and countermeasures.
  - https://www.heise.de/security/alerts/ [ger] this website provides useful information about the latest security alerts.
- Encrypt the communication with your website (SSL (Secure Sockets Layer / TLS (Transport Layer Security)
  - An encrypted communication with a website offers data security and data integrity.
- Only allow access to your website via an encrypted connection
  - All communication with your website should only be possible using an encrypted connection.
  - If your site is accessed via http://example.com it should be automatically redirected to https://example.com.
  - Use HSTS Header to force a secured connection.
- Regular backups
  - Backups may help you to be on the safe side. Damaged or corrupted files can be resorted using a backup.
  - Some offer automatic backups for your site. It is advised to check the services of your own web host if they offer something similar.
  - For more on this topic please refer to ”When and how often should I install updates?” [34] [ger].
- Only use software or plugins from trustworthy and credible sources.
  - Software and plugins, for example for an CMS like WordPress, should only be acquired from trustworthy sources. If the source of the piece of software or plugin is unknown it should not be used.
  - A trustworthy and credible source for WordPress Plugins can be found here: https://wordpress.org/plugins/. Even when the source is trustworthy, do not forget to keep an eye on comments and ratings from other users.
- Use security plugins for your CMS.
  - Security plugins for a CMS are a good addition and offer various features that may improve the security of your CMS even more.
- Use scan tools to test your website.
  - Scan tools can be used to scan your site for known security issues, for example the implementation of certain HTTP security headers.
- Check your anti-virus software.
  - Only use anti-virus software which is known to be credible, trustworthy and updated frequently.
- XSS-Prevention.
  - Check user input before sending it to the web server to prevent injection of any code (e.g., HTML, URL or JavaScript).
  - XSS-prevention plugins are available for well-known CMS.
- Prevention of SQL- or code-injection (Figure 1.
  - For more on this topic please refer to this blog post:https://blog.varonis.de/sql-injection-verstehen-erkennen-und-verhindern/ [ger]

Figure 1. Metaphoric visualization of Code-Injection

Examples: Regular Updates - To ensure the security of a website and other systems it is necessary to pay attention to operating system and software updates. Especially security updates should not be neglected. Pattern ”When and how often should I install updates?” [34] [ger] provides more insight into this subject.

Encrypt the communication with your Website - Pattern How do I encrypt the communication with my website [35] [ger] explains how to implement a SSL/TLS Encryption into your own website.

ScanTools - After securing the website and the associated server, it is possible to test them with ScanTools for known vulnerabilities. ScanTools examine a variety of security aspects of websites and may help you to find out whether your site is lacking in terms of security. More on this topic can be found in Pattern ”How do I check the security of my website?” [36] [ger].

Keeping yourself up-to-date! - Following websites can be used to keep yourself informed about the latest web-security issues.

- https://heise.de/security/
- http://seclists.org

Prevent XSS (Cross Site Scripting) - The following pages provide the necessary knowledge on how to prevent XSS:

- Cross-Site Scripting (XSS) unterbinden [37]
- Cross Site Scripting Prevention Cheat Sheet [38]

HSTS Header on an Apache Server - The following code can be inserted into your .htaccess file to enforce an encrypted connection with your website if it is implemented:

```
# Use HTTP Strict Transport Security to force
    client to use secure connections only

Header set Strict-Transport-Security
    "max-age=3600" env=HTTP
```

For more information, please visit: How do I activate HSTS for my website? [39] [ger]

References: Studies on the subject of password security -

Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms [40]

The true cost of unusable password policies: password use in the wild [41]

Multifactor Authentication -

Multifactor Authentication [42] [ger]

Keywords: Security, Encryption, Authentication

B. How do I check the security of my website?

Intent: This Pattern shows common methods used to test the security of websites.

Problem Statement: To ensure the safety of computers users usually trust in anti-virus and malware software. Unfortunately, it is not that simple for websites. Keeping a website safe and secure is more demanding and requires more attention rather than only installing a certain piece of software or plugin. Websites can be different and diverse, so are their weaknesses and vulnerabilities.

Scenario: Before you open your website to the world, it is recommended to perform a vulnerability scan to test the security of your site.

Solution:

- Check whether your site offers encrypted communication.
  - The website should only accept and allow encrypted communication.
  - For more information on this topic, please refer to Pattern ”How do I encrypt the communication with my website?” [35] [ger].
- Run a virus and malware scan over all files on the web server.
  - Check whether your web hosting service provider offers virus protection services.
  - In case users are allowed to upload files to your web space, scan them before they are actually uploaded onto the web server.
  - To be on the safe side, avoid file upload features for users in general.
- Verify check sums of software and downloaded files you want to install onto the server.
  - Software and Plugins from trusted sources usually offer check sum values for the files you can download.
  - In case no check sum values are provided double check the credibility of the site offering the download.
- Use online ScanTool websites to have your site tested by a third party.
  - In general, ScanTool websites scan your website for known security issues and vulnerabilities (e.g., whether SQL- or Code-injection is possible or whether a n encrypted connection is mandatory to access your website).
  - Some ScanTool websites even offer explanations and solutions for certain vulnerability if they find one on your site.
- Provide minimal errors to your users.
  - Do not provide full exception details in your error messages. Keep them simple. The more information you reveal in your error messages, the easier it is to possibly exploit them.

Examples: Verifying Check Sums -

There are several types of file check sums that are used to verify downloaded files. One of the widely used and most popular is the MD5 check sum. In this example, we will check the MD5 check sum of a WordPress 4.9.7 installation package in a .zip file (see Figure 2. The MD5 check sum in this case is provided on the WordPress download page and can be accessed by clicking the md5 link 2 under the file type downloading. Clicking this ink will open a new page, showing the associated MD5 check sum for the file. In this example, the check sum for the zip file is: 075a6e7585c61e3aa2874d91d32bc336.

4.9.7	July 5, 2018	zip (md5 | sha1)	tar.gz (md5 | sha1)	IIS zip (md5 | sha1)

Figure 2. WordPress Version 4.9.7 Download Page

After downloading the zip file, use the terminal to acquire

the check sum of that file.

Terminal Command for macOS:

```
# Command
md5 wordpress-4.9.7-de_DE.zip

# Output
MD5 (wordpress-4.9.7-de_DE.zip) =
    075a6e7585c61e3aa2874d91d32bc336
```

Terminal command for Windows:

```
# Command
certUtil -hashfile wordpress-4.9.7-de_DE.zip
    MD5

# Output
MD5-Hash of wordpress-4.9.7-de_DE.zip:
    075a6e7585c61e3aa2874d91d32bc336
```

Now, compare the check sum from the terminal with the check sum provided on the website where you downloaded the file. If they match, data integrity is given and the file is safe to use.

For more information on the subject of check sums please refer to https://itsfoss.com/checksum-tools-guide-linux/.

ScanTools -

Mozilla Foundation Security Check: This security check provided by the Mozilla Foundation aims to help you to configure your site safely and securely. If you want to test your site, just type in the URL of the site you want to be scanned in here: https://observatory.mozilla.org.

SSL Labs - SSL Labs performs a deep analysis of the performs a deep analysis of the configuration of any SSL web server on the public Internet. It is pointed out by the provider, that this service is purely for information purposes and is not recommended for commercial purposes.

- https://www.ssllabs.com/ssltest/

Additional ScanTool Service Websites

- https://www.virustotal.com/de/ [ger]
- https://www.htbridge.com/websec/

Important Note: Please carefully observe the terms and conditions of the respective providers before you use their services.

References: Mozilla's security test for websites [43] [ger].

Does a web server need an anti virus software installed? [44].

Keywords: Security, Integrity, ScanTools, Malware

V. Discussion

Keeping things simple is important to keep them understandable and it is at this point where we must ask the question on how we can lower the educational access barrier to fit in with the ease of on-line content creation of today. There are many individuals out there who have access to a great number of content management systems and similar tools right at their fingertips. By using these with a "website first, security last"-mentality, they will endanger both themselves and those using their creations. Many of these creators have no aspirations to ever become experts, be it out of necessity or simple disinterest.

Restricting the access of these creators is not an attractive option and would run counter to the freedom of today's on-line world. What these individuals need are resources that reflect not only their level of expertise but also their goals and needs. If all these individuals need is an adequately secured website, then they should have the information on how to do so. This seems contrary to most sound educational intuitions, as it would essentially mean to aim for mediocrity. However, this kind of mediocrity would not compete with a superior solution but with no solution at all instead.

Using such patterns to inform one's web development will not lead to not fully secured websites. Instead, they are a resource provide an adequate minimum of security and privacy on-line. This should not be seen as to be in competition with available professional and educational literature. Rather, it supplements it. The internet of today is a much more diverse place than it was even a decade ago. In order to increase everyone's security and privacy, available information must reflect this diversity. And when some individuals aim either low or not at all as far as privacy and security are concerned, then giving them the means to at least aim low seems to be an overall improvement.

VI. Conclusion

Designing a website with privacy and security in mind from the beginning is a nontrivial task for both novices and experienced professionals. There are limits to what can be communicated in a way so that it can be understood and implemented by novices. Nonetheless, a certain minimum is as helpful as it is necessary in order to improve the online experience for everyone. In this paper, we provided two meta-patterns as guidance regarding what constitute privacy- and security-relevant aspects of a website, and how these attributes can be verified in the end. Concrete instructions on the implementation of such solutions is still necessary. Such instructions must, just like the high-level meta-patterns, be written in a way that is suitable for novices. Thus, additional lower-level patterns are necessary. In Parts 2 and 3, we provide two sets of such pattern solutions. Future work will focus on continually extending the pattern solutions to cover more issues, as well as updating existing patterns in order to keep their solutions valid and usable.

Acknowledgment

The financial support by the Internet Privatstiftung Austria (IPA) under the program "netidee" with the title "SecPatt" under grant number 2390 is gratefully acknowledged.

References

[1] "World Wide Web (W3)," http://info.cern.ch/hypertext/WWW/TheProject.html, (accessed April 10, 2019).

[2] "Total number of Websites," https://www.internetlivestats.com/total-number-of-websites, 2019 (accessed April 10, 2019).

[3] Privacy and security on the microsoft developer network. [Online]. Available: https://msdn.microsoft.com/en-us/library/ms976532.aspx (2018)

[4] A. Amran, Z. F. Zaaba, M. M. Singh, and A. W. Marashdih, "Usable security: Revealing end-users comprehensions on security warnings," Procedia Computer Science, vol. 124, 2017, pp. 624 – 631, 4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917329666

 ISBN: 978-1-61208-708-5

[5] R. LaRose, N. J. Rifon, and R. Enbody, "Promoting personal responsibility for internet safety," Commun. ACM, vol. 51, no. 3, Mar. 2008, pp. 71–76. [Online]. Available: http://doi.acm.org/10.1145/1325555.1325569

[6] N. Kumar, K. Mohan, and R. Holowczak, "Locking the door but leaving the computer vulnerable: Factors inhibiting home users' adoption of software firewalls," Decis. Support Syst., vol. 46, no. 1, Dec. 2008, pp. 254–264. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2008.06.010

[7] R. Shillair, S. R. Cotten, H.-Y. S. Tsai, S. Alhabash, R. LaRose, and N. J. Rifon, "Online safety begins with you and me," Comput. Hum. Behav., vol. 48, no. C, Jul. 2015, pp. 199–207. [Online]. Available: http://dx.doi.org/10.1016/j.chb.2015.01.046

[8] D. Lee, R. Larose, and N. Rifon, "Keeping our network safe: A model of online protection behaviour," Behav. Inf. Technol., vol. 27, no. 5, Sep. 2008, pp. 445–454. [Online]. Available: http://dx.doi.org/10.1080/01449290600879344

[9] C. L. Anderson and R. Agarwal, "Practicing safe computing: A multimedia empirical examination of home computer user security behavioral intentions," MIS Q., vol. 34, no. 3, Sep. 2010, pp. 613–643. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017470.2017481

[10] S. Pearman, J. Thomas, P. E. Naeini, H. Habib, L. Bauer, N. Christin, L. F. Cranor, S. Egelman, and A. Forget, "Let's go in for a closer look: Observing passwords in their natural habitat," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '17. New York, NY, USA: ACM, 2017, pp. 295–310. [Online]. Available: http://doi.acm.org/10.1145/3133956.3133973

[11] R. Shay, L. Bauer, N. Christin, L. F. Cranor, A. Forget, S. Komanduri, M. L. Mazurek, W. Melicher, S. M. Segreti, and B. Ur, "A spoonful of sugar?: The impact of guidance and feedback on password-creation behavior," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 2903–2912. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702586

[12] C. Alexander, The Timeless Way of Building. New York, USA: Oxford University Press, 1979.

[13] C. Alexander, S. Ishikawa, and M. Silverstein, A Pattern Language: Towns, Buildings, Construction. New York, USA: Oxford University Press, 1997.

[14] J. O. Coplien, Software Patterns. New York, USA: SIGS Books, 1996.

[15] K. Quibeldey-Cirkel, Design Patterns in Object Oriented Software Engineering. *O*rig. title: Entwurfsmuster: Design Patterns in der objektorientierten Softwaretechnik. Berlin, Germany: Springer, 1999.

[16] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Pearson, 1994.

[17] J. O. Borchers, "A pattern approach to interaction design," AI & SOCIETY, vol. 15, no. 4, 2001, pp. 359–376.

[18] S. Köhne, A Didactical Aproach towards Blended Learning: Conception and Application of Educational Patterns. *O*rig. title: Didaktischer Ansatz für das Blended Learning: Konzeption und Anwendung von Educational Patterns. Hohenheim, Germany: University of Hohenheim, 1995.

[19] A. Dearden and J. Finlay, "Pattern languages in hci: A critical review," Human-Computer Interaction, 2006, pp. 49–102, Sheffield Hallam University. [retrieved: 02, 2016] URL: http://research.cs.vt.edu/ns/cs5724papers/dearden-patterns-hci09.pdf.

[20] A. F. Blackwell and S. Fincher, "PUX: Patterns of User Experience," Interactions, vol. 17, no. 2, 2010, pp. 27–31.

[21] M. V. Welie and G. C. V. D. Veer, "Pattern languages in interaction design: Structure and organization," in Proc. Interact '03, M. Rauterberg, Wesson, Ed(s). IOS. IOS Press, 2003, pp. 527–534.

[22] J. Muoz-Arteaga, R. M. Gonzlez, M. V. Martin, J. Vanderdonckt, and F. lvarez Rodrguez, "A methodology for designing information security feedback based on user interface patterns," Advances in Engineering Software, vol. 40, no. 12, 2009, pp. 1231 – 1241, designing, modelling and implementing interactive systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997809000519

[23] J. Vlissides, Pattern Hatching: Design Patterns Applied. Addison-Wesley, 1998.

[24] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. Henry Riche, "Design patterns for data comics," in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ser. CHI '18. New York, NY, USA: ACM, 2018, pp. 38:1–38:12. [Online]. Available: http://doi.acm.org/10.1145/3173574.3173612

[25] Oecd privacy guidelines. [Online]. Available: http://www.oecd.org/sti/ieconomy/privacy-guidelines.htm (2013)

[26] Privacy and security on google for education. [Online]. Available: https://edu.google.com/k-12-solutions/privacy-security/?modal_active=none (2018)

[27] Microsoft security guide. [Online]. Available: https://technet.microsoft.com/en-us/library/bb794718.aspx (2018)

[28] A. Mirnig, T. Kaiser, A. Lupp, N. Perterer, A. Meschtscherjakov, T. Grah, and M. Tscheligi, "Automotive user experience design patterns: An approach and pattern examples," International Journal On Advances in Intelligent Systems, vol. 9, 2016, pp. 275–286.

[29] D. Wurhofer, M. Obrist, E. Beck, and M. Tscheligi, "A quality criteria framework for pattern validation," International Journal On Advances in Software, vol. 3, no. 1 and 2, 2010, pp. 252–264.

[30] A. Krischkowsky, D. Wurhofer, N. Perterer, and M. Tscheligi, "Developing patterns step-by-step: A pattern generation guidance for hci researchers," in PATTERNS 2013, The Fifth International Conferences on Pervasive Patterns and Applications. IARIA, 2013, pp. 66–72. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=patterns_2013_3_30_70053

[31] A. G. Mirnig, A. Meschtscherjakov, N. Perterer, A. Krischkowsky, D. Wurhofer, E. Beck, A. Laminger, and M. Tscheligi, "User experience patterns from scientific and industry knowledge: An inclusive pattern approach," International Journal On Advances in Life Sciences, vol. 7, no. 3 and 4, 2015, pp. 200–215. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=patterns_2015_2_30_70011.

[32] A. G. Mirnig, A. Lupp, A. Meschtscherjakov, E. Economidou, and M. Tscheligi, "Security patterns for webdesign: a hierarchical structure approach," in CHI Conference on Human Factors in Computing Systems Extended Abstracts, ser. CHI'19 Extended Abstracts. New York, NY, USA: ACM, 2019. [Online]. Available: http://doi.acm.org/10.1145/3290607.3312789

[33] S. B. Griffith, Sun Tzu: The art of war. Oxford University Press London, 1963, vol. 39.

[34] SecPatt, "When and how often should I install updates? [ger]," https://www.secpatt.at/patterns/pt_1/, 2018 (accessed April 10, 2019).

[35] SecPatt, "How do I encrypt the communication with my website? [ger]," https://www.secpatt.at/patterns/pt_4/, 2018 (accessed April 10, 2019).

[36] SecPatt, "How do I check the security of my website? [ger]," https://www.secpatt.at/patterns/pt_6/, 2018 (accessed April 10, 2019).

[37] "Prevent Cross-Site Scripting (XSS) [ger]," https://www.php-kurs.com/cross-site-scripting-xss-unterbinden.htm, (accessed April 10, 2019).

[38] "Cross Site Scripting Prevention Cheat Sheet," https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.md, (accessed April 10, 2019).

[39] "How do I activate HSTS for my website?" https://www.cyon.ch/support/a/wie-aktiviere-ich-http-strict-transport-security-hsts-fur-meine-website, (accessed April 10, 2019).

[40] "Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms." https://ieeexplore.ieee.org/abstract/document/6234434/?part=1, 2012 (accessed April 10, 2019).

[41] "The true cost of unusable password policies." https://dl.acm.org/citation.cfm?id=1753384, 2010 (accessed April 10, 2019).

[42] "Multifactor Authentication," https://www.onlinesicherheit.gv.at/praevention/konten_und_passwoerter/mehrfaktor-authentifizierung/249584.html, 2017 (accessed April 10, 2019).

[43] "Mozilla introduces free security test for websites," https://www.heise.de/ix/meldung/Mozilla-bringt-kostenlosen-Sicherheitstest-fuer-Websites-3306197.html, 2016 (accessed April 10, 2019).

[44] "Does a web server need an anti virus software installed?" https://security.stackexchange.com/questions/245/does-a-webserver-need-an-antivirus-software-installed, 2010 (accessed April 10, 2019).

 ISBN: 978-1-61208-708-5