

Figure 1: The pattern generation process.

Security Patterns for Webdesign: a Hierarchical Structure Approach

Alexander G. Mirnig

University of Salzburg
Salzburg, Austria
alexander.mirnig@sbg.ac.at

Alexander Meschtscherjakov

University of Salzburg
Salzburg, Austria
alexander.meschtscherjakov@sbg.ac.at

Manfred Tscheligi

University of Salzburg AIT
Salzburg Vienna, Austria
manfred.tscheligi@sbg.ac.at

Artur Lupp

University of Salzburg
Salzburg, Austria
artur.lupp@sbg.ac.at

Eleni Economidou

University of Salzburg
Salzburg, Austria
eleni.economidou@sbg.ac.at

ABSTRACT

In today's age, a wide range of individuals create their own web presence. Thanks to modern tools, creating a website is easier than ever. In order to make sure that this increased accessibility does not come at the cost of decreased security, the respective web design knowledge should become more accessible as well. We created 16 security patterns for web design based on expert knowledge. We present the solution hierarchy of these patterns and how they might be applied by non-expert users.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5971-9/19/05.

<https://doi.org/10.1145/3290607.3312789>

How do I encrypt the communication with my website?

Intent

The aim of this pattern is to provide sufficient information allowing users to request and obtain an *SSL / TLS Certificate* from a trusted Authority and to install it on their own website.

Problem Statement

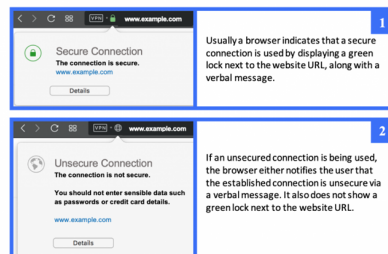
Providing an encrypted connection between the user's web browser and a website is mandatory, especially when sensitive data (e.g., names, addresses and credit card data) is communicated. Encrypted communication guarantees data integrity, provides safety is necessary for certain web applications. Even though many websites work with sensitive data, only about 40% of the websites currently online provide encrypted communication.

Scenario

To provide a safe and secure environment for the visitors of your website, it is mandatory to enable a secure connection via HTTPS on your own website.

Solution

To enable the usage of HTTPS on a website, the first step is to obtain an SSL / TLS Certificate from a Certificate Authority. This Certificate then needs then to be installed on the webserver. This is done via the admin control panel provided by the webhost. After setting up the Certificate, the `.htaccess` file needs to be adjusted to force the web browser to use a secured connection when connecting to the website. Afterwards it is necessary to check in different web browsers whether the connection to the website is forced to use HTTPS.



There are a handful of Certificate Authorities offering their services to issue an SSL / TLS Certificate. This pattern focuses on the Certificate Authority called <https://letsencrypt.org>, which provides free, automated and open certificates for websites.

Figure 2: Abbreviated example pattern (p1/3).

KEYWORDS

Design patterns; user-centered; usable security; usable privacy; novice users.

ACM Reference Format:

Alexander G. Mirmig, Artur Lupp, Alexander Meschtscherjakov, Eleni Economidou, and Manfred Tscheligi. 2019. Security Patterns for Webdesign: a Hierarchical Structure Approach. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI'19 Extended Abstracts), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3290607.3312789>

INTRODUCTION


People who create website nowadays range from teenagers who want to set up their own online forum to small or mid-sized companies who need a web presence without having the resources for a dedicated web development department. 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. The contribution of this paper is a collection of relevant issues for novice web developers and a structure to guide novice users via pattern solutions for these issues. In this paper we describe the process used to create 16 patterns (A presentation of the pattern collection is not in the focus of this paper.), including the problem mining and pattern writing process. We present the multi-level pattern structure, intended to guide the novice reader through the pattern collection. For that purpose, we introduce the concept of *pre- and post-meta-patterns* as entry and exit points for a laypersons using these patterns in a web design process. We conclude with reflections on the current state of the pattern collection and overall structure, the interplay between privacy and security, as well as the issue of enabling layperson web developers to develop responsibly.

BACKGROUND

A crucial aspect of security decisions regarding home computers and websites is the existing knowledge about computers, the internet, and their security issues. In general, internet users seems to be still very naïve regarding security and privacy, despite the warnings of dangers regarding the lack of online safety practices [5, 10]. Even with guidance from experts, non-tech users lack the sense of internet security and privacy. Reeder et al. [9] asked security experts to list their top three advice to users not familiar with technologies to protect their security online. This resulted in a total of 152 unique advice and the conclusion that unless the community changes its approach of providing guidance, users will remain less secure than they otherwise could be. They state that there is a need to discuss, define and prioritize security guidance for layperson internet users. Wash et al. [11] argued, that each user should be educated individually regarding security knowledge, as the more-is-better approach might not yield the best results. One way of providing laypersons an efficient access to security and

Examples

Receive an SSL / TLS Certificate using Let's Encrypt



Visit <https://letsencrypt.org> and click on the "Get Started" button. The following page explains how to enable HTTPS on your website, depending whether you have remote Server Access via the Shell (SSH) (e.g., access to the server command line terminal) or not. In case your webhost does not provide you with Shell access, the procedure will get more complicated and further steps might be necessary. In order to receive the Certificate, Let's Encrypt needs to know, whether you are the owner of the web domain in question. If you ...

- **have access to the Shell**
 - go to <https://certbot.eff.org>
 - choose the software and system your server is using
 - the next steps are explained on the site
 - in case Certbot does not fit your needs, it is possible to use other ACME v2 compatible Client tools to receive the Certificate that are listed here:
 - <https://letsencrypt.org/docs/client-options/>
- **do not have access to the Shell, but ...**
 - the webhost supports Let's Encrypt
 - Contact your webhost via mail and ask them to handle the acquisition and installation of the Certificate via Let's Encrypt.
 - The webhost doesn't support Let's Encrypt
 - In case your webhost does not support Let's Encrypt, it is still possible to manually receive a Certificate by using your own system in combination with Certbot. However, this process requires special knowledge and experience with command line tools. As this process takes knowledge and time, it is not advised for beginners because this needs to be done each time the Certificate runs out.
 - More information for that procedure can be found here: <https://certbot.eff.org/docs/using.html#manual>
 - Apart from getting the Certificate the manual way, you could also ask your webhost whether they are interested to include Let's Encrypt support.
 - The last resort would be to change to a webhost that supports Let's Encrypt out of the box.

A detailed explanation on how to use Let's Encrypt can be found on: <https://letsencrypt.org/getting-started/>. As the installation of a Certificate requires the usage of the admin panel provided by the webhost, it is recommended to contact your host in case you have any questions.

Figure 3: Abbreviated example pattern (p2/3).

¹Link to full pattern (in German): https://www.secpatt.at/erfahrungslevel/einsteiger/pt_4/

safety guidelines can be done though design patterns (cf. to [1] for the origins of patterns). Design patterns are structured documentations of solutions to reoccurring problems. Patterns can occur on different levels of abstraction, referred to as *high-* or *low-level* patterns [3], 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*. In this paper, we intend to pursue a similar approach as [2] to provide privacy- and security-related information in a more novice-appropriate, pattern-like format.

PATTERN GENERATION METHOD

We employed a two-stage generation process adapted from Mirnig et al. 2016 [6]. Stage 1 included expert interviews (with professional web designers, a university professor, and a CERT professional), the creation of a problem statement list, a rating and clustering of this list into meta patterns, and a selection of 16 final problem statements. The second stage included pattern mining and writing, resulting in a first version of patterns including a multi-level structure, which was then discussed in a workshop, based on the results of the workshop an iteration of the patterns led to the final pattern collection including the final structure. The process is visualized in Figure 1. From the interviews, we only extracted information regarding encountered issues, not mitigation strategies or similar. This resulted in a list of **42 problem statements**. We compiled these into a separate list and had them rated regarding their priority in a professional web developer meeting with 20 participants, which ended up in a list of **16 most important issues**, each of which being the basis for one pattern. Then, the pattern writing process began. Each pattern was written by an HCI expert with experience in pattern writing. The pattern content was based on an internal state-of-the-art containing guidelines [4, 7, 8], topic relevant scientific publications (ACM, Springer, IEEE), and information gained from the interview protocols. The information thus gained was then condensed into a pattern format adapted from Mirnig et al. 2016 [6]. Each pattern consists of a **name**, the **intent**, a **problem description**, an **scenario**, the **solution**, at least one **example**, **references**, and **keywords**. 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 Mirnig et al. [6]. Each pattern was rated individually for each of its subcategories (Name, Intent, etc.) and then discussed in plenum. Figures 2-4 provides an abbreviated example pattern¹ that has been developed.

PATTERN USE AND HIERARCHICAL STRUCTURE

While the individual pattern helps to guide the reader along to a specific solution, it can only do so within its limited problem scope. An additional obstacle that novices face, however, is that of unfamiliarity with the target domain, including problem complexes and hierarchies. In other words: a problem solution is only good to the reader if they (a) know that the problem exists, and (b) aware that there is a solution. Thus, we put particular emphasis on the pattern hierarchy and provide a way that

Activate HTTPS on your website There are multiple ways to allow or force a secure communication between a web browser and your website. One way is to modify a certain page or .php file that is accessed by the URL. Just add the following code in the .php file to enable HTTPS when accessing the page:

```
// Require https
if ($_SERVER['HTTPS'] != "on") {
    $url = "https://". $_SERVER['SERVER_NAME'] .
    $_SERVER['REQUEST_URI'];
    header("Location: $url");
    exit;
}
```

If you want to force a secure connection on your whole website, you can alter the .htaccess file, which can usually be found on the root of your webserver (e.g., when using an Apache server). To force the use of HTTPS and redirect HTTP requests automatically to HTTPS, use the code below:

```
# HTTP to HTTPS redirecting

RewriteEngine On
RewriteCond %{HTTP_HOST} ^example\..com [NC]
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://www.example.com /$1 [R=301,L]
```

Instead of `example.com` please use your own URL.

References

<https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https>
<https://developers.google.com/web/progressive-web-apps/>
<https://www.w3.org/2001/tag/doc/web-https> <https://mod-rewrite-cheatsheet.com/#basics-enable-htaccess> <https://www.ssllabs.com>

Keywords

HTTPS, cryptography, encryption, SSL, TLS

Figure 4: Abbreviated example pattern (p3/3).

would guide the reader through the whole process of designing a web presence on a level that is shallow enough to be easily accessible yet still contain enough information to provide usable development knowledge. This was one of the reasons why we asked the web developers to elaborate their design process in a chronological manner, as we could then arrange the pattern solutions accordingly. In addition, thanks to the internal structuring and keyword references within the patterns, we were finally able to condense the patterns into *three levels of abstraction*.

- **Level 1** contains basic information necessary for any type of web presence: web host properties, cyber-attack detection, updates, backups, and basics on communication encryption.
- **Level 2** Provides more in-depth information regarding cyber-attack prevention and response, data storing and data protection provisions, as well as basic information on mail servers.
- Finally, **level 3** covers more specific uses by describing more in-depth information for mail server operation, commercial web shop use, and legal protection.

In addition to the patterns within these three levels, there are two *meta-patterns*. As mentioned previously, a meta-pattern is a pattern on a higher abstraction level, which usually describes a meta level problem consisting of other lower-level problems, or a solution space consisting of other lower-level solutions. In order to incorporate this logically into the structure, we introduced two meta-levels from a chronological – one in the beginning and one in the end. The pattern on the **pre-meta-level** (Pre-ML) provides a high-level list of factors contributing to the security of a web presence. It serves as an orientation to the reader and helps them comprehend the problem space. It also serves as a guidance towards the individual lower-level solutions and where they can be found. The **post-meta-level** (Post-ML) comes right at the end and consists of information on how the security of a website can be verified. It provides a means to the reader to verify whether their solutions were implemented correctly. Both meta-level patterns aid the reader in properly navigating the information presented to them within the lower-level patterns. The post-meta-level is particularly vital for a novice developer, as they cannot be expected to get everything right on their first attempt and with self-study being their only resource. The post-meta-level can be seen as the counterpart to the meta-level. In this way, both levels also serve a chronological function and mark the end- and starting-points for the reader.

DISCUSSION

The contribution of this paper are not the individual patterns themselves or the overall mining and iteration process but rather the collection of important problem questions for laypersons as well as the hierarchical structure, which allows them to use patterns in a new way. The traditional way is to search for a suitable pattern to a certain problem. We suggest a different approach, where a clear structure of hierarchical pattern levels guides the reader from pre- to post-metalevel, thereby providing a clear path for the uninformed reader (see Figure 5).

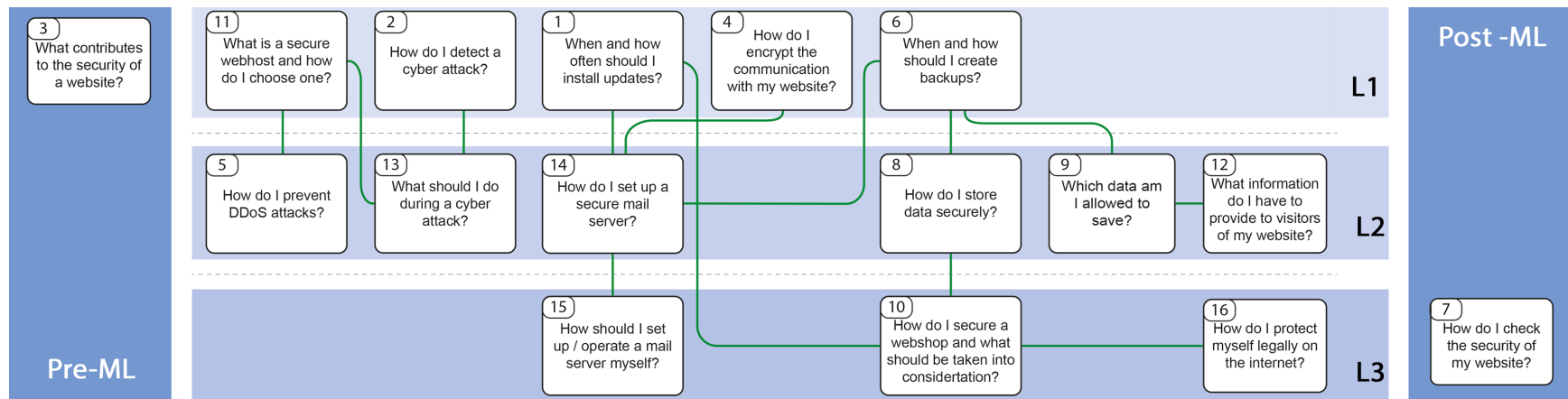


Figure 5: Hierarchical structure of the pattern solutions. Patterns on the pre-meta-level (Pre-ML) and post-meta-level (Post-ML) signify the beginning and end.

Completeness vs. Conciseness. The pattern collection is intended as a tool to help novices design their web presence with security and privacy in mind. The solutions that such novices create will always be worse than expert-created ones. Since the patterns are targeted at a non-expert audience, the extent of this information has to be limited to a certain minimum. It is also for this reason, that many of the patterns contain high-level knowledge rather than low-level instructions, since novices' problems often start at a very initial or conceptual level. A definitive list of the most common issues would require a more thorough analysis of incident statistics world-wide, together with broader input on how these problems are rated, in order to account for regional differences and other potentially relevant factors. Our patterns provide sourced information on problem solutions, along with step-by-step instructions and implementation examples, all formatted to make reading easier without losing the essentials of the presented content. Validating the patterns will require longitudinal study along to rate the developed result regarding privacy and security.

Privacy vs. Security. As can be seen in the final pattern level structure, most issues are related to security as a whole, with only few explicit references to privacy related issues. Since both concepts are related and privacy is often seen as a part of security, we explicitly asked for input regarding both during the interviews. But even then, the discussion almost always quickly went back to security in general. This is, in part, because the interviewees always related privacy-fostering measures to a

certain level of security as a necessary precondition, which seems reasonable. But it also seems as if there was less of an awareness regarding privacy issues, which could be related to their consequences.

CONCLUSIONS

Starting from the position that today's internet is populated by a wide variety of individuals with an equally varying levels of expertise, we investigated the issue of web design and development knowledge that is accessible to individuals without expert knowledge. Opting for a design pattern approach, we compiled a list of the 16 most important issues when developing a web presence. The quality of the patterns themselves and their suitability for novices was not within the scope of this paper and is subject to future work, together with an extension of the pattern basis. The main contribution of this paper consisted in a pattern structure that guides the reader through the design process in a level-by-level manner. Thus, the reader is guided not only through the individual problems but also through the problem space itself, which fosters accessibility of the presented information.

ACKNOWLEDGMENTS

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

REFERENCES

- [1] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. 1997. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, USA.
- [2] Benjamin Bach, Zehong Wang, Matteo Farinella, Dave Murray-Rust, and Nathalie Henry Riche. 2018. Design Patterns for Data Comics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 38, 12 pages. <https://doi.org/10.1145/3173574.3173612>
- [3] Jan O. Borchers. 2001. A Pattern Approach to Interaction Design. *AI & SOCIETY* 15, 4 (2001), 359–376.
- [4] GoogleForEd 2018. Privacy and Security on Google For Education. Retrieved Sep 19, 2018 from https://edu.google.com/k-12-solutions/privacy-security/?modal_active=none
- [5] Julian Jang-Jaccard and Surya Nepal. 2014. A survey of emerging threats in cybersecurity. *J. Comput. System Sci.* 80, 5 (2014), 973 – 993. <https://doi.org/10.1016/j.jcss.2014.02.005> Special Issue on Dependable and Secure Computing.
- [6] Alexander Mirnig, Tim Kaiser, Artur Lupp, Nicole Perterer, Alexander Meschtscherjakov, Thomas Grah, and Manfred Tscheligi. 2016. Automotive User Experience Design Patterns: An Approach and Pattern Examples. *International Journal On Advances in Intelligent Systems* 9 (2016), 275–286.
- [7] MSGuide 2018. Microsoft Security Guide. Retrieved Sep 19, 2018 from <https://technet.microsoft.com/en-us/library/bb794718.aspx>
- [8] OECD 2013. Privacy Guidelines. Retrieved Sep 19, 2018 from <http://www.oecd.org/sti/ieconomy/privacy-guidelines.htm>
- [9] Robert W. Reeder, Adrienne Porter Felt, Sunny Consolvo, Nathan Malkin, Christopher Thompson, and Serge Egelman. 2018. An Experience Sampling Study of User Reactions to Browser Warnings in the Field. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 512, 13 pages. <https://doi.org/10.1145/3173574.3174086>
- [10] S. Shyam Sundar and Sampada S. Marathe. 2010. Personalization versus Customization: The Importance of Agency, Privacy, and Power Usage. *Human Communication Research* 36, 3 (Jun 2010), 298–322. <https://doi.org/10.1111/j.1468-2958.2010.01377.x>
- [11] Rick Wash and Emilee J Rader. 2015. Too Much Knowledge? Security Beliefs and Protective Behaviors Among United States Internet Users.. In *SOUPS*. 309–325.