Will You Trust This TLS Certificate?

Perceptions of People Working in IT

Martin Ukrop
mukrop@mail.muni.cz
CRoCS, Masaryk University
Brno, Czech Republic

Lydia Kraus
lydia.kraus@fi.muni.cz
CRoCS, Masaryk University
Brno, Czech Republic

Vashek Matyas
matyas@fi.muni.cz
CRoCS, Masaryk University
Brno, Czech Republic

Heider Ahmad Mutleq Wahsheh
heider.wahsheh@unive.it
Ca' Foscari University of Venice
Venice, Italy

ABSTRACT

Flawed TLS certificates are not uncommon on the Internet. While they signal a potential issue, in most cases they have benign causes (e.g., misconfiguration or even deliberate deployment). This adds fuzziness to the decision on whether to trust a connection or not. Little is known about perceptions of flawed certificates by IT professionals, even though their decisions impact high numbers of end users. Moreover, it is unclear how much does the content of error messages and documentation influence these perceptions.

To shed light on these issues, we observed 75 attendees of an industrial IT conference investigating different certificate validation errors. We also analysed the influence of re-worded error messages and redesigned documentation. We find that people working in IT have very nuanced opinions with trust decisions being far from binary. The self-signed and the name constrained certificates seem to be over-trusted (the latter also being poorly understood). We show that even small changes in existing error messages can positively influence resource use, comprehension, and trust assessment. Our re-worded error messages and documentation can be directly adopted.

CCS CONCEPTS

• **Security and privacy** → **Usability in security and privacy**; **Authentication**; • **Human-centered computing**;

KEYWORDS

warning design, documentation, TLS certificate, usable security

1 INTRODUCTION

Nowadays, communication protected by TLS (formerly SSL) is getting more and more prevalent on the Internet (in May 2019, over 80% of page loads in Google Chrome were done over HTTPS [20]). For the TLS infrastructure to work, end entities authenticate themselves using X.509 certificates [13]. Certificate validation errors are quite common [4, 5], although an error does not necessarily imply a security incident. For example, getting a self-signed certificate may be either an attack (adversary pretending to be a trusted site) or only a misconfiguration (the local administrator unable or unwilling to obtain a certificate signed by a trusted certificate authority). Imagine the developer connecting to an identity service – if they accept the self-signed certificate presuming it is just a misconfiguration (when it is not), they will leak authentication credentials of the end users.

Much work has been conducted to understand why system administrators have difficulties in getting TLS deployment and certificate handling right. Usability shortcomings in the TLS deployment process [26] and related tools [39], as well as other factors beyond the system administrators' control [14], seem to be common root causes. Furthermore, it has been shown that invalid certificates are sometimes deployed deliberately [17].

Nevertheless, most of the previous work on TLS warnings focused on the warning perceptions of end users [6, 8, 18, 32, 38], while the perception of certificate flaws by people working in IT seems underresearched. We, therefore, focus on people interacting with software, systems and networks beyond the end user interface (henceforth referred to as *'IT professionals'*[1]). IT professionals have a stronger impact on the ecosystem: While the decision of end users to ignore a warning influences only themselves, a similar decision by an IT professional potentially puts numerous end users at risk.

Whenever IT professionals encounter a certificate validation error, we would want them to investigate and draw the right security conclusion. To gain a deeper understanding of whether they can do it, we conducted an empirical study[2] with 75 attendees of an industrial IT conference. In the test scenario, study participants were confronted with certificate validation errors in a command line interface, asked to investigate the problem and subsequently assess their trust in the certificates (methodology described in Section 2). Firstly, we look

---

[1] This includes people responsible for development, testing, deployment and administration alike. Note that we do not limit the sample to any single role since the same errors are encountered by all.
[2] Study artifacts are available at https://crocs.fi.muni.cz/papers/acsac2019.

into how IT professionals perceive TLS certificate flaws and examine the online sources they use to comprehend the problem (Section 3). Secondly, we are interested how much the content of error messages and documentation influences their perceptions (Section 4). The participants were thus divided into two groups: one with current OpenSSL error messages and documentation and another with reworded error messages and redesigned documentation. In contrast to the work of Gorski et al. [22], we adjust just the content, not the whole concept of the errors and documentation to ensure compatibility. In summary, our work yields the following contributions.

(1) *We provide detailed insights into the perception of different certificate flaws by IT professionals.* We find that their opinions are very nuanced. The trust in flawed certificates is far from binary. Moreover, the self-signed and name constrained certificates seem to be over-trusted (the latter also being poorly understood). The trust in expired certificates heavily depends on the time elapsed from the expiry date.

(2) *We propose and evaluate the influence of redesigned error messages and documentation.* We find that the redesigned certificate validation messages and documentation positively influenced resource use, comprehension, and trust assessment.

2 METHODOLOGY

To gain insights into participants' perception of certificate flaws, we presented them with a scenario-based certificate validation task. Within the task, they successively evaluated multiple server connections with flawed TLS certificates. Participants were divided into two groups with different error messages and documentation.

2.1 Procedure

The study consisted of three parts: 1) an introductory questionnaire on previous experience, 2) a certificate validation task with trustworthiness assessment and 3) an exit semi-structured interview followed by an educational debriefing. All parts of the experiment were in English, except for the interview, which could optionally be in Czech. The study had a mixed design: The participants were validating five different certificates (the within-subjects factor) across two different conditions (original or redesigned documentation; the between-subjects factor). The order of the certificates was randomized and participants were randomly assigned to one of the between-subjects conditions. A small pilot of the whole study took place before the experiment. It had shown no issues in the experiment design, but uncovered some smaller technical glitches and task ambiguities.

2.1.1 Initial Questionnaire. At the beginning, we inquired into the previous experience of the participants: the years of employment, formal education, self-reported knowledge of computer security and X.509 certificates in particular (5-point Likert scale from *poor* to *excellent* for the last two questions). To make sure the sample uses relevant tools, we asked about previous experience with OpenSSL and other certificate manipulation utilities. Afterward, they received 16 questions of the Security Behavior Intentions Scale (SeBIS) [16]. The full questionnaire can be found in Appendix A.

2.1.2 Certificate Validation Task. Participants were asked to evaluate their trust in five different server certificates (in a randomized order). The scenario stated they were implementing a patch for the conference website to support registration using external identity providers (Microsoft, Fedora Project, GitHub, Google and Facebook). It was said that they had started by writing a small program establishing a TLS connection to the providers' servers. However, since some certificates caused validation errors, the participants were asked to investigate these and evaluate the trustworthiness of the connections.

When running the given program for each identity provider in the command line interface, a short certificate validation message was displayed. Apart from that, the certificate was "downloaded" and saved next to the program binary to allow for a further (manual) inspection. The participants were not limited in their investigation – they could inspect the certificate, browse the Internet or use local sources for as long as they wanted. For convenience, the printed version of the task contained a command to view the downloaded certificate. The complete task can be found in Appendix A.

Below, we discuss the five certificate issues used in our experiment. We included three cases commonly occurring in the wild, one extension that would allow for a fine-grained control in the ecosystem (yet is still rarely used) and a control case.

- **Hostname mismatch.** The certificate that was presented by *oauth.facebook.com* had the subject of *oauth.facesbook.com* (an extra 's' in the name). Browser-based measurements show roughly 10% [4] to 18% [5] of the certificate errors due to a name mismatch.

- **Self-signed.** The certificate for *id.fedoraproject.com* was signed by its own key (it did not chain up to a trusted root). Such errors constitute roughly 1% of browser certificate errors [4], but their prevalence in TLS scans can be as high as 25% [23] or 88% [11].

- **Expired.** The certificate was issued for *login.microsoft.com* and its validity ended seven days before the experiment. Expired certificates cause from 4% [4] to 7% [5] of browser certificate errors and are present at about 18% of the sites [23].

- **Name constraints.** The chain for *accounts.google.com* contained an intermediate authority that was constrained (by the root authority) to issue certificates in the domain subtree **.api.google.com*. Thus, the violated name constraint invalidated the certificate chain. The name constraints extension was standardized back in 2002 in RFC 3280 [24] and the CA/Browser forum guidelines allow for its use on public websites since June 2012 [10]. Despite its potential to prevent certificate misissuance, only 0.001% of the unexpired certificates in May 2019 used this extension [15].

- **OK.** The certificate for *auth.github.com* had no validation issues. It was included as the control case.

All the certificates (apart from the self-signed), were chained up to a self-signed certificate for the experiment Root CA. The participants were explicitly instructed (in the task specification) to trust this root.

After each case, the participant was asked to assess the trustworthiness of the connection with such a certificate. A 7-point scale was provided. It spanned from *'Trust 6/6: I'm totally satisfied. If it were my bank's website, I would log in without worries.'* down to *'Trust 0/6: Outright untrustworthy. It is not safe to browse or to trust any information there.'* The full scale can be found in Appendix A.

2.1.3 Post-task Interview. After the certificate validation, a short semi-structured interview took place. First and foremost, the participants were asked to explain in their own words the problem with each certificate. Retrospective evaluations help gain valuable

insights into participants' reasoning, especially in the context of usability studies with advanced users. Due to the task complexity, this approach was more suitable than methods that foster immediate reflection, such as think-aloud protocols [29]. After discussing the individual certificate issues, expired certificates were talked over in detail. The full list of base questions guiding the interview can be found in Appendix A. To educate the IT community, the experimenter clarified any misconceptions or misunderstandings in the debriefing after the participants described their view.

2.2 Experimental Conditions

We deployed two experimental conditions differing in the error messages and documentation with each participant randomly assigned to one condition. The fact that they may have alternative error messages was only disclosed in the debriefing after completing the task.

The first condition, henceforth referred to as the *'original'*, used current error messages from OpenSSL 1.1.0g-fips (build from November 2nd, 2017) with the corresponding documentation (manual pages and resources available on the Internet). We decided to use OpenSSL as a base condition since it is one of the most popular crypto libraries for generating keys/certificates [30]. The original error messages contain several descriptive words and a textual error code that can be looked up in the documentation. The documentation is available in the form of manual pages and the online OpenSSL Wiki [31]. However, the descriptions in the documentation contain only little extra information compared to the actual error message.

The second, *'redesigned'*, condition had re-worded error messages with a link to the website containing our custom-written documentation. We expected the documentation might have room for improvement due to related research on poor usability of OpenSSL [39]. To ensure compatibility, we decided not to design a completely novel system of errors (compared to the work by Gorski et al. [22]).

The first draft of the new error messages and documentation was done by the principal researcher synthesizing personal experience from open-source development of certificate-validating software, existing documentation and designs by multiple students (asked to create usable documentation for the chosen certificate validation error). Subsequently, the draft underwent multiple iterations between the researcher and a usable security expert with experience from both industry and academia. Overall, the design process followed the principles of warning design proposed by Bauer et al. [7].

The error messages were re-worded to explain the cause in more detail, but to still fit into a single line (including the link to the new documentation page). We decided to keep the text codes for easy reference and reasonable compatibility. The documentation for each case has a consistent structure of four subsections: 1) summarizing the problem; 2) explaining what happened; 3) giving the security perspective (explaining the risks) and 4) proposing what steps could be taken next. The re-worded error messages and the full documentation can be seen in Appendix B.

2.3 Setting and Recruitment

The study took place in 2018 at DevConf.CZ 2018, an international community conference for developers, admins, DevOps engineers, testers and others interested in open source technologies such as Linux, middleware, virtualization or storage. The conference was organized by Red Hat Czech and had approximately 1700 attendees. We had set up a booth offering participation in research regarding the usability of developer tools. The booth featured three Linux workstations with (virtualized) Fedora 27 (a standard workstation flavor) connected to the Internet. The operating system logged executed commands and browsed websites as well as recorded the screen.

The participants were not primed to expect security tasks, but before accepting them, we made sure that they knew what TLS certificates were. We decided not to limit our sample to security professionals, as it turns out that the majority of developers work with security features from time to time [28]. The whole experiment took about 30–40 minutes per participant (pre-task questionnaire, task, post-task interview). For completing the task, we gave participants merchandise of the organizing company (i.e., no monetary compensation).

All participants were briefed about the extent of processed personal information. Participation was voluntary and the audio recording of the interview was optional. All the data was collected pseudonymously. As stated in the informed consent form, the subset of the data (excluding the screen and interview recordings) were anonymized and are published as an open dataset. The study design has been approved by our research ethics committee.

2.4 Participants

From the 78 recruited participants,[3] 3 ⚥ were omitted due to not completing all the sub-tasks, resulting in the base sample count of 75 ⚥. The participants were employed in the IT sector on average for 9.55 ± 7.06^{4} years (*median* 8). In more detail, 16% of the participants (12 ⚥) were employed for 0–2 years, 25% (19 ⚥) for 3–5 years, 16% (12 ⚥) for 6–8 years, 11% (8 ⚥) for 9–11 years and 32% (24 ⚥) for 12 or more years (yet nobody for more than 30 years).

The formal education levels were also diverse: 33% of the participants (25 ⚥) did not have a formal degree in computer science or engineering, 25% (19 ⚥) had a bachelor's degree, 38% (29 ⚥) a master's degree and 3% (2 ⚥) a Ph.D. Participants' self-reported security knowledge was *'good'* on average (*mean* 1.91 ± 1.02, *median* 2), their experience with X.509 certificates was *'fair'* on average (*mean* 1.00 ± 1.07, *median* 1). Almost all participants (91%, 68 ⚥) had used OpenSSL before, followed by Network Security Services (25%, 19 ⚥), Java Keytool (25%, 19 ⚥), GnuTLS (19%, 14 ⚥), and Windows Certutil (12%, 9 ⚥). About a third had used the Let's Encrypt Certbot before (36%, 27 ⚥).

To estimate the security behavior of our participants, we used the *Security Behaviors Intentions Scale* (SeBIS) [16]. Our sample tends to have stronger intentions towards secure behavior than the general population [16], especially for the device securement subscale (the individual subscale scores on average 0.1–1.3 points higher).

The vast majority consented to the audio recording of the interview, forming the sample size for the qualitative analyses (89%, 46 ⚥ in English, 21 ⚥ in Czech). 59% of the participants (44 ⚥) were (at random) assigned to the original condition with the remaining 41% (31 ⚥) assigned to the redesigned conditions. Checking for the differences across conditions, we see no significant differences with respect to self-reported previous experience, IT education, IT employment or used tools suggesting that the samples are comparable.

---

[3]From now on, we use the symbol ⚥ to denote the participants.

[4]The plus-minus symbol (±) is used to denote standard deviation.

To check for potential biasing effects, we asked if the participants had taken part in an experiment conducted at the same conference in the previous year [39]. Only a minority of the participants did (13%, 10 ♀). Omitting these participants did not alter the conclusions drawn from the results (we compared all the results for the whole sample and the sample omitting these participants). We, therefore, present the results of the full sample.

2.5 Data Collection and Analysis
First of all, data from all sources was matched based on timestamps and each participant had a pseudonym assigned. Timestamps were used to compute the time spent on the individual tasks. The browser history was utilized to identify used resources.

We performed a qualitative descriptive analysis of the post-task interview [35]. After transcribing the interviews, two researchers independently familiarized themselves with the data. The data was then processed using open coding [34] (the researchers independently looked for reoccurring themes and assigned codes to text passages). The analysis was open but broadly framed by the participants' comprehension of the encountered validation messages, their opinions and actions. After the open coding, the researchers discussed the created codes and consolidated a common codebook. The transcripts were then re-coded using the shared codebook. The results reported in Sections 3 and 4 are based on this coding round.

To ensure analysis reliability, a third independent coder coded half of the English and half of the Czech transcripts using the shared codebook. Interrater agreement between the original coders and the independent coder showed to be *substantial* (according to Landis and Koch [27]) for both English (Cohen's $\kappa = 0.69$, $p < 0.001$) and Czech (Cohen's $\kappa = 0.63$, $p < 0.001$). All coders used all codes, indicating that the defined codes were actually present in the interviews and that the codebook was defined well. Some deviations that appeared in the coding stemmed from the fact that the independent coder was slightly more conservative in assigning codes compared to the original coders.

Supplementary materials, including the experimental setup (the complete virtual machine) and the anonymized dataset are available at https://crocs.fi.muni.cz/papers/acsac2019.

2.6 Study Limitations
As is the case with every study, various limitations may diminish the applicability of results. First, to ensure that participants would behave as usual, we designed a realistic and appealing task (at a conference on open-source technologies, we had them "develop" a patch for the registration system). We limited neither time nor resources allowing the participants to behave as they would in reality. Realistic and well-known entities and hostnames were chosen for the certificates. Although different hostnames may have slightly different reception, we preferred this to the ecologically less valid option of solving five separate cases with the same hostname.

The second great concern is the sample bias – we recruited attendees of a single industrial conference. Nevertheless, we think our sample reflects the wider population of people working in IT well enough. Even though the conference was biased towards open-source and Linux technologies, a 2018 Developer Survey by Stack Overflow [37] estimates that almost half of the developers contribute to open source

and Linux is the most developed-for platform. Compared to the mentioned survey, our sample follows the general trends for professional experience and education but has a slightly higher mean (peaking at 3–5 years working in IT instead of 0–2 and at the master's degree instead of the bachelor's). This means the reported comprehension may be a bit higher for our sample compared to the wider population (as our sample is a bit more experienced and educated).

Thirdly, participants' behavior may have been primed towards security by the context or parts of the questionnaire. However, we were cautious not to mention security when advertising the study and tried to recruit all participants passing by regardless of their skill.

Lastly, we tried to mitigate multiple response biases. To combat the question order bias, we randomized the order of questions in the initial questionnaire and the order of evaluated certificate cases both in the task and the trust scale. Furthermore, to work against the bias of the response order, we inverted all the Likert scales in both the questionnaire and trust assessment for half of the participants. To lower the observer effect (participants behaving differently when being watched), the interviewers left the participants alone when completing the tasks (but were still available in case the participant wanted to consult something). Some participants might have been more cautious than usual due to the social desirability bias – to account for this, we see the obtained trust evaluations as a lower bound.

3 PERCEPTION OF CERTIFICATE FLAWS
In this section, we present comprehension and perceived trustworthiness of tested certificate flaws, together with the reasoning that participants provided in the exit interview. Comprehension is based on qualitative analysis of the post-task interview (67 ♀). Trust assessment and answers to the structured interview questions are available for everybody (75 ♀). The section ends with a cross-case comparison and investigation in the influence of previous knowledge.

3.1 Hostname Mismatch Case
The hostname mismatch flaw was comprehended quite well – a majority of the participants mentioned the core of the issue was the server hostname not matching the name provided in the certificate (code BadName, 50 ♀, see Table 1 for frequency, simplified definition and a representative quote on this and other codes). Many participants explicitly mentioned the extra letter 's' in the certificate name (NameCheck, 27 ♀), hinting at the fact that they looked at the certificate to investigate the issue. The prevalent opinion regarding the cause of the error was that it was an attack of some sort (Attack, 22 ♀), but a few participants explicitly mentioned it could be only a mistake or a typo (Mistake, 8 ♀).

The connection was on average assessed as *'being outright untrustworthy'* (*mean* 0.68 ± 1.08, *median* 0, see Fig. 1 for detailed information and Appendix A for the whole scale). We find the low trust in this case unsurprising as the server with mismatched second-level domain name gives almost no identity guarantees.

3.2 Self-signed Case
The self-signed case is also dominated by codes indicating comprehension: Mentioning that the certificate is self-signed (ByItself, 50 ♀), that no CA was involved in its issuance (NoCA, 28 ♀) and that literally anyone (including the participants themselves) can issue such

Table 1: Overview of the comprehension and reasoning codes occurring in at least 10% (7 ⚥) of the interviews (67 ⚥ in total, 39 ⚥ in the original (o.) and 28 ⚥ in the redesigned (r.) condition). Codes labeled with an asterisk (★) indicate case comprehension.

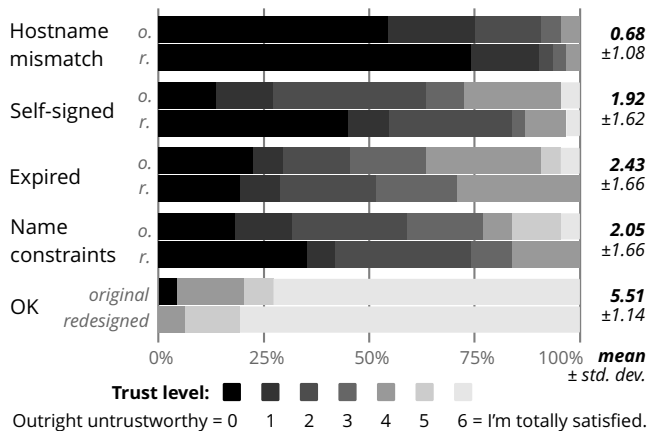| Case / Code	⚥ (o.+r.)	Code definition	Representative quote
Hostn. mismatch BadName★	50 ⚥ (29+21)	The certificate subject and server name do not match.	*"The last one server, Facebook, [the certificate] was issued for a different hostname." [P39, original]*
NameCheck	27 ⚥ (16+11)	Mentioning the exact difference in the names.	*"[...] because it is not Facebook, it is Facesbook or something like that." [P57, original]*
Attack	22 ⚥ (12+10)	Connection may be attacked.	*"It can be some phishing site or something like this." [P76, original]*
Mistake	8 ⚥ (3+5)	It can be only a mistake or server misconfiguration.	*"And in this case – it's a different domain, but I'd say it's some kind of typo or something like that." [P63, redesigned]*
Self-signed ByItself★	50 ⚥ (30+20)	The certificate is signed by itself/self-signed.	*"That it is not signed by the other authority, but it's signed by itself." [P15, original]*
NoCA★	28 ⚥ (16+12)	No CA was involved in issuing the certificate.	*"It was signed by local server for which it was generated. It was not signed by official authority." [P20, original]*
AnyoneCan★	21 ⚥ (12+9)	Anyone can create such a certificate.	*"Self-signed certificate? Anyone can create self-signed certificates." [P78, original]*
IfExpected	10 ⚥ (6+4)	It is OK if such certificate is known/expected.	*"If I knew that the certificate should be self-signed, I could consider it trustworthy." [P09, original]*
Internal	10 ⚥ (7+3)	They are used for testing or internal purposes.	*"[...] and it's usually used either by internally or for testing purposes. It shouldn't be used publicly." [P11, original]*
Attack	8 ⚥ (2+6)	Connection may be attacked.	*"[...] because that can be any hacker, [they] can phish." [P66, original]*
Expired NoLonger★	62 ⚥ (34+28)	The certificate has expired.	*"Microsoft certificate has expired, it's out of date." [P30, original]*
Mistake	27 ⚥ (15+12)	It can be only a mistake or server misconfiguration.	*"[...] it could be just forgotten and they're about to do it, they're about to renew it or something." [P10, original]*
Common	18 ⚥ (12+6)	Expired certificates are common/occur in the wild.	*"Ah, right, so, expired certificates are pretty common, so from what I can see [...]" [P01, redesigned]*
OKBefore★	14 ⚥ (4+10)	The certificate was OK in the past.	*"Yeah, the Microsoft one is expired. So it was valid in the past, and I looked at the date [...]" [P18, redesigned]*
Reputation	13 ⚥ (7+6)	Taking into account the subject of the certificate.	*"If it's like a small businesses from my local neighborhood, I would probably trust them." [P62, original]*
Attack	8 ⚥ (4+4)	Connection may be attacked.	*"[maybe] the attacker has stolen a certificate which was previously valid and has been revoked [...]" [P37, original]*
Name constraints Constraint★	25 ⚥ (12+13)	The name of the endpoint certificate is constrained.	*"I understood that there is some chain and a certain point in chain is restricting the hostname to ..." [P39, original]*
Wrong	19 ⚥ (12+7)	Giving reasoning that is wrong.	*"I find out that one of the authorities was listed as false, but the other two were fine." [P10, original]*
NotKnow	14 ⚥ (11+3)	I do not understand it.	*"I don't really understand the whole thing." [P62, original]*
Attack	10 ⚥ (3+7)	Connection may be attacked.	*"[I'd let Google know that they have a rogue admin..." [P26, redesigned]*
CAProblem★	10 ⚥ (2+8)	The intermediate CA was not allowed to issue this.	*"[...] CA has explicitly said 'I am not allowed to sign this, you should not trust this.'" [P26, redesigned]*
CAConstr★	9 ⚥ (2+7)	The constraints are set by the CA.	*"The certificate authority up the chain specifies that only domains with 'api.google.com' are valid." [P18, redesigned]*
Mistake	7 ⚥ (3+4)	It can be only a mistake or server misconfiguration.	*"It seemed like it was just an innocent misconfiguration of the kind that happens all the time." [P19, original]*
NoInfo	7 ⚥ (5+2)	Finding more information on the matter is difficult.	*"For this one I really try to find some documentation, but there was no documentation on this." [P68, original]*
OK NoIssue★	61 ⚥ (34+27)	There is no problem.	*"There wasn't a problem, it was good, OK." [P22, original]*
ExtraCheck	13 ⚥ (5+8)	Doing extra manual checks on the certificate.	*"I think it was safe, but I looked into the cert and I couldn't find anything wrong, so I would trust it..." [P13, redesigned]*
BugFree	12 ⚥ (7+5)	The program is trusted to do the verification correctly.	*"[...] everything looked fine and I thought: 'Well, if the testing tool is good, I'll trust that.'" [P77, redesigned]*

Figure 1: Comparison of trust assessment among certificate cases split by condition. Bars are normalized due to the different number of participants in each condition (44 ⚥ for the original condition, 31 ⚥ for the redesigned one).
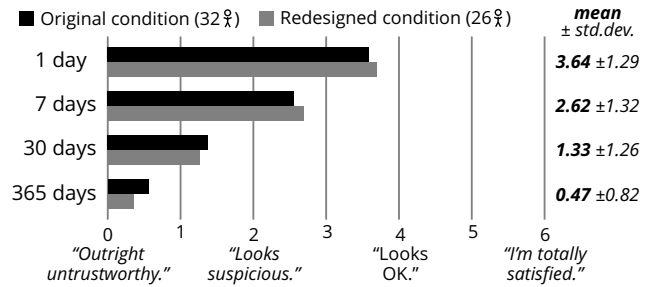

Figure 2: Mean trust in the certificates depending on how long it has been expired, compared across conditions. Only participants with different ratings for durations are included.

a certificate (ANYONECAN, 21 ⚥). Some participants mentioned that they would find self-signed certificates acceptable in places where they were told to expect them (IFEXPECTED, 10 ⚥). The self-signed certificates were multiple times said to be OK for testing or internal purposes (INTERNAL, 10 ⚥). A few participants mentioned the possibility of an attack when encountering self-signed certificates (ATTACK, 8 ⚥).

The connection with a self-signed certificate was perceived as *'looking suspicious'* (*mean* 1.92±1.62, *median* 2, see Fig. 1). The rating similar to the expired case (see Section 3.3) is surprising, given that the expired certificate had at least provided an identity guarantee in the past, whereas the self-signed certificate provides literally no identity guarantees and never did.

3.3 Expired Case

When participants described their perception of the problem, most of them seemed to comprehend the source of the flaw – the certificate being no longer valid (NOLONGER, 62 ⚥) or having been valid in the past but not now (OKBEFORE, 14 ⚥). The prevalence of the first code is unsurprising as it is literally the content of the displayed error message.

Several considerations seemed to play a role in opinion forming: Many thought expired certificates can be grounded in a misconfiguration, negligence or operator error on the server side, and thus the flaw is non-malicious (MISTAKE, 27 ⚥). A considerably lower number of participants mentioned that the connection might be attacked or an old certificate may be abused (ATTACK, 8 ⚥).

Next, multiple participants expressed the opinion that expired certificates are quite common (COMMON, 18 ⚥). The last notable mention comes for participants binding the trust in expired certificates to the subjects using them (REPUTATION, 13 ⚥) – seeing an expired certificate as more acceptable for local business than for Microsoft (the specific comparison was drawn because the expired certificate used in the task was issued for *login.microsoft.com*).

Participants rated the trustworthiness of the connection with the server certificate expired one week ago as rather low, close to a *'looks suspicious'* point on our scale (*mean* 2.43±1.66, *median* 3, see Fig. 1).

Given that the provided certificate does not guarantee the identity information anymore, it would have been unsurprising to us if the trustworthiness assessment went even lower.

From our own experience as programmers and IT security educators, we further suspected the trust might be influenced by the expiry duration. Subsequently, we had prepared several follow-up questions for the expired case in the exit interview. Fair enough, the majority of participants (77%, 58 ⚥) reported that the rating would be different for shorter/longer periods past the expiration date. However, only 65% of these (36 ⚥) actually checked the expiry date in the given task (an example for the not uncommon difference between self-reported intention and behavior). When subsequently asked to rate trust in the connection with the server certificate expired a day/week/month/year ago, the mean trust spanned from *'looking OK'* (*mean* 3.64±1.30, *median* 4) for a day-expired certificate to almost *'outright untrustworthy'* (*mean* 0.47±0.82, *median* 0) for the certificate expired a year ago. The detailed gradual decline of trust can be seen in Fig. 2. The results indicate significant differences in trust among the differently old expired certificates (Friedman ANOVA, $\chi^2(3) = 141.58, p < 0.001$). Almost all pairwise comparisons (except for month/year) were significant (Dunn-Bonferroni post-hoc analysis, $p < 0.005$). The effect size was the largest for the day/year comparison ($r = 0.22$), followed by day/month ($r = 0.16$) and week/year ($r = 0.15$). This confirms that expiry is not a binary feature and people's trust significantly depends on the time elapsed from the expiry date.

When asked about the relation of the expiry date and the certificate revocation, only 24% of the participants (18 ⚥) knew that expired certificates are no longer included in the certificate revocation lists (CRL). Of the 57 ⚥ who did not know this, 25 ⚥ indicated that knowing it would lower their trust rating. That suggests that insufficient knowledge of the ecosystem may cause over-trusting the certificates.

3.4 Name Constraints Case

The participants' problem descriptions in this case suggest only a limited comprehension of the flaw and its security implications. Less than half of the participants mentioned that the domain name in the endpoint certificate is constrained or that there is a blacklist/whitelist for these names (CONSTRAINT, 25 ⚥). Even fewer people said that the CA was not allowed to issue the endpoint certificate (CAPROBLEM, 10 ⚥) or that it is the CA who is constraining the name

(CAConstr, 9 ♀). More participants were able to describe the problem in the redesigned case (see details in Section 4.1).

The name constraint was often surrounded by explicit statements of not understanding the problem (NotKnow, 14 ♀) or difficulty to obtain further information (NoInfo, 7 ♀). This was expected, as name constraints are little used (and thus little known) and the error message in the original condition only says *'permitted subtree violation'* without further details. The notion of name constraints being the most complicated case is further supported by the opinion of the participants themselves – when asked in the post-task interview what task they saw as the most difficult to understand, almost all of them (90%, 61 ♀) indicated the name constraints case.

A high count of people provided wrong reasoning (Wrong, 19 ♀) for the cause of the problem. A few participants (5 ♀) considered the case to be the same as the hostname mismatch case. The others blamed other (often unrelated and flawless) parts of the certificates: the basic constraints extension (*'the CA is set to false'*), the key purpose extension (*'the purpose is wrong'*), the name format (*'there are wildcards'*, *'the third-level domain is missing'*) or other certificates in the chain (*'the chain integrity is broken'*, *'the intermediate CA certificate is missing'*). Note that misconceptions and the lack of understanding were more often mentioned in the original condition (see Section 4.1). Of the few opinions on the causes of the validation error, more participants saw it as a potential attack (Attack, 10 ♀) than a mistake or a misconfiguration (Mistake, 7 ♀).

The average trust into the connection was *'looking suspicious'* (*mean* 2.05 ± 1.66, *median* 2, see Fig. 1). We see it as surprising that the name constraints error is rated similarly to the expired case, given that its security implications are potentially more severe (the intermediate authority being corrupted). The described misunderstandings further suggest the error message does not sufficiently pinpoint the relevant issue in the certificate structure.

3.5 OK Case (Flawless)

Concerning the control case, the majority of participants stated there was nothing wrong with the certificate (NoIssue, 61 ♀). Some participants did an extra check manually (ExtraCheck, 13 ♀), though only a few of them (3 ♀) hinted this behavior was influenced by the experimental environment. Many mentioned that their opinion and trust is based on the fact that they were instructed to presume that the used program is bug-free (BugFree, 12 ♀).

Regarding the perceived trust in connections where the server is using such a certificate, participants were on average close to *'being totally satisfied'* with the certificate that validated without errors (*mean* 5.51 ± 1.14, *median* 6, see Fig. 1). We do not find this result surprising, given that the certificate was flawless.

3.6 Case Comprehension and Trust

Let us now compare and contrast the individual cases. We reason about case comprehension by analyzing frequently occurring interview codes describing the problem, see Fig. 3. Note that this measure is only approximate – (not) mentioning the particular things does not necessarily mean (not) comprehending the issue. We report only descriptive stats due to the qualitative nature of the data.

The name constraints seem to be less understood compared to the other cases: Firstly, the codes describing the problem are less
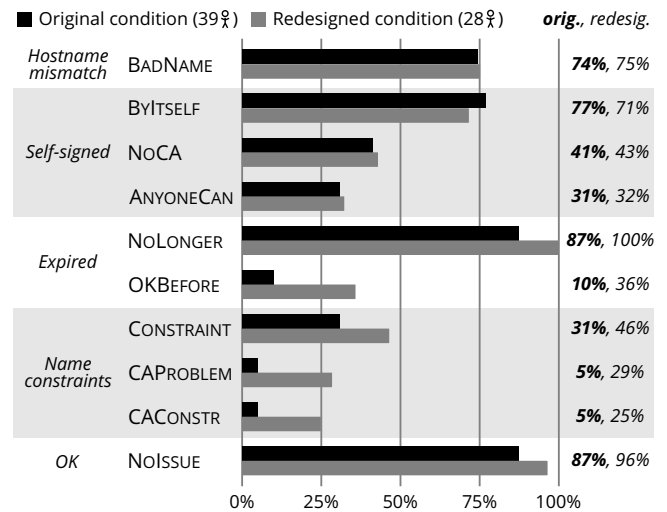

Figure 3: An overview of codes indicating case comprehension for all cases, compared across conditions. For more details on the codes, see Table 1.

frequent (only about half of the participants having at least one of these codes compared to about 80% for other cases). Secondly, there are frequent codes explicitly admitting not understanding the issue (NotKnow, NoInfo) or misunderstanding (Wrong). The comprehension seems to have been better for the redesigned condition in some cases (see Section 4.1 for details).

The trust is compared in Fig. 1. As expected, there are again significant differences among the cases (Friedman ANOVA, $\chi^2(4) = 183.5$, $p < 0.001$). Pairwise comparisons (Dunn-Bonferroni) show that the certificate in the OK case was trusted significantly more (*median* 6) and the one in the hostname mismatch case was trusted significantly less (*median* 0) than other cases ($p < 0.005$ for all comparisons). The expired, self-signed and name constraints cases did not significantly differ from each other (*median* 3 for expired, *median* 2 for self-signed and name constraints). The effect size was the largest for the comparison OK/hostname mismatch ($r = 0.26$), followed by the comparison of OK with other cases ($0.15 \leq r \leq 0.18$).

Neither the high trust in the OK case nor the low trust in the hostname mismatch case is surprising. However, we see the self-signed and name constraints cases as over-trusted (at least when compared to the expired case): The expired certificate provided full authenticity assurances before expiration, but the self-signed certificate never did (indeed anyone could have created such a certificate). The name constraints case suggests malicious activity at the authority level, which is far more severe (the intermediate CA was prohibited from issuing the certificate, yet it did). Obtained trust assessments signal potential misunderstanding of security implications in these cases.

3.7 Task Times and Used Resources

We have measured the time spent on each certificate case from the moment of seeing the validation message for the first time till the message of the following case was invoked. The differences among cases are significant (Friedman ANOVA, $\chi^2(4) = 48.81, p < 0.001$). Post-hoc
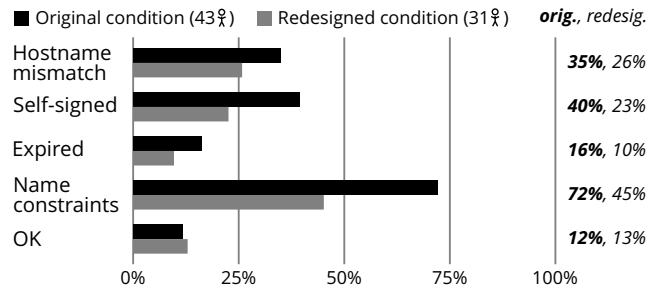
Figure 4: Overview of online browsing during task completion, split and normalized by condition, compared across cases. One participant with connection problems is omitted.

analysis (Dunn-Bonferroni) shows a significantly higher mean time for the name constraints case (*mean* 4.9 *min.*, *median* 4.5 *min.*) when compared to other cases ($p \leq 0.001$ for all comparisons). All other cases had similar means of about 2 minutes (OK 2.1 *min.*, expired 1.75 *min.*, self-signed 2.3 *min.*, hostname mismatch 2.6 *min.*). For detailed time distribution, see Appendix A. The standard deviations are comparable to means, indicating the timing data might be influenced by other participant-specific factors that we did not measure (experience did not seem to have an influence, see Section 3.8).

Looking at the resources used, about three-quarters of the participants (73%, 55 ♀) browsed the Internet. One of the participants experienced technical problems with the Internet connection – we thus only report data for 74 participants. The basic overview of the online resource use per case can be seen in Fig. 4. The differences among the cases are significant (Cochran's Q test, $\chi^2(4) = 68.85$, $p < 0.001$) with the most participants browsing for information about the name constraints case (61% of all participants, 45 ♀), less about self-signed and hostname mismatch cases (32%, 24 ♀ and 31%, 23 ♀) and the least about the expired and OK cases (14%, 10 ♀ and 12%, 9 ♀). The post-hoc analysis (Dunn-Bonferroni) shows significant pairwise comparisons for all combinations with the name constraints case ($p \leq 0.001$) and also among three other pairs ($p \leq 0.05$).

In summary, the comparison of timings and resource use confirms obstacles in the comprehension of the name constraints case, which is in alignment with results from previous sections.

3.8 Influence of the Previous Knowledge

Taking the self-reported previous experience into account, we find no significant systematic influence on trust assessment or task times (testing for the number of years in IT, formal education in computer science, knowledge of security/certificates or previous usage of other tools using a diversity of parametric and non-parametric tests). This result is in contrast to the 2017 study by Acar et al. [3] that reports a significant difference in both functionality and security of programming tasks, depending on self-reported years of experience.

4 INFLUENCE OF REDESIGNED ERRORS

In this section, we examine whether error comprehension by IT professionals can be influenced by a content change in the error messages and documentation (without a major redesign). The section reiterates over the presented results again, focusing on the observed

differences between experimental conditions – 44 ♀ for the original error messages and documentation versus 31 ♀ for the redesigned condition (39 ♀/28 ♀ for interview codes).

4.1 Case Comprehension and Trust

Let us look again at the most common interview codes. For the comprehension codes (see Fig. 3): In the name constraints and expired cases, the comprehension codes occur notably more frequently in the redesigned condition (CONSTRAINT 31% vs. 46% in the redesigned condition, CAPROBLEM 5%/29%, CACONSTR 5%/25%, NOLONGER 87%/100%, OKBEFORE 10%/36%). The redesigned condition further features a decrease in the codes that describe incomprehension (WRONG 31%/25%, NOTKNOW 28%/11%, NOINFO 13%/7%, absolute numbers in Table 1).

Looking at other codes: In the OK case, more people did manual certificate checks in the redesigned condition (EXTRACHECK 15%/29%), probably due to the new documentation suggesting to the user to perform extra checks. In the redesigned condition, there was an increase in believing that the self-signed and name constraints error is a consequence of an attack (ATTACK, 5%/21% for self-signed, 8%/25% for name constraints). Other than the few mentioned differences, the occurring codes are consistent between the conditions.

Let us now look back at the trust assessment overview in Fig. 1. We compared the differences between conditions using a Mann-Whitney-U test for each case. The results indicate that, in the redesigned condition, the trust was significantly lower for the self-signed case ($U = 433.00$; $p = 0.006$, $z = -2.77$, $r = -0.32$). The mean trust decreased from 2.32 ± 1.54 (*median* 2) to 1.35 ± 1.58 (*median* 1). There was a similar trend for hostname mismatch and name constraints, but the differences are not statistically significant.

All in all, the redesigned documentation seems to be no worse than the original one. It seems to significantly increase the comprehension of the flaw in the name constraints case and better conveys the possible attack vectors in the OK, self-signed and name constraints cases. As we considered the self-signed case to be over-trusted, we see the influence of the redesigned documentation on the trust ratings as helpful, shifting the perception of IT professionals in the desired direction.

4.2 Task Times and Used Resources

The overview of the task times across cases was given in Section 3.7. For all cases, there was a general trend towards lower tasks times for participants in the redesigned condition (see Fig. 5 in Appendix A). In addition, there were notably fewer people with very high times (over 8 minutes) in the name constraints case (20%, 9 ♀ for the original condition, 6% 2 ♀ for the redesigned). Nevertheless, the differences were not statistically significant (Mann-Whitney-U test for each case).

As for the resources used, roughly the same proportion of participants browsed the Internet in both conditions (79%, 34 ♀ for the original condition, 68%, 21 ♀ for the redesigned). Inspecting the differences by case (Fig. 4), there was a significant decrease in the name constraints case for the redesigned condition (72% to 45%, Fisher's exact test, $p = 0.029$). There was also a difference in the online search behavior – significantly fewer people used online search in the redesigned condition (only 62% of those who browsed online, while it was 100% in the original condition, $\chi^2(1) = 10.72$, $p = 0.001$).

Both these results support the hypothesis that the redesigned error messages and documentation positively influence resource use. Participants in the redesigned condition had to spend less effort to achieve similar trust assessment and comprehension for the expired, hostname mismatch and OK case. Furthermore, they even achieved better results for the self-signed case (trust assessment) and the name constraints case (comprehension).

As detailed in Section 2.2, there was a link to the new documentation website *x509errors.cz* embedded directly in the re-worded validation messages. The link in the message was followed by 71% of the participants in the redesigned condition who browsed online (15 ♀). It is important to mention that all the participants browsing online who did not use online search (8 ♀) visited only the linked documentation page. People opening *x509errors.cz* usually did so for multiple cases (*mean* 4.2 ± 1.27, *median* 5). Everybody consulting the page at least once did so for the name constraints case. Incorporating a documentation link directly into the error message thus turned out to be highly influential – the page was visited by the majority of browsing participants, often in multiple cases.

5 RELATED WORK

In this section, we focus on the interaction of IT professionals with TLS certificates and the perception of TLS warnings by end users. Furthermore, we briefly discuss the general influence of documentation and error messages on system security.

5.1 IT Professionals and TLS Certificates

In general, the environment of X.509 certificates and TLS infrastructure is rather complicated, allowing for a wide variety of things that can go wrong [12]. As shown by Krombholz et al. [26], TLS seems to be complicated to set up and configure even for IT professionals. In a conducted usability study on the deployment of TLS on Apache servers, even people with network security knowledge struggled to configure the server correctly. When being asked about the usability shortcomings of the deployment process, participants noted the unavailability of best practice tutorials, misleading terminology and error messages, as well as a weak default configuration.

Apart from the server configuration, there are deficiencies in programming APIs as well. Georgiev et al. [19] showed that options, parameters, and return values of widely used TLS implementations can be easily misunderstood by developers, often leading to certificate validation in non-browser software done incorrectly or not at all.

Ukrop and Matyas present another usability study [39], in which participants were to generate and validate certificates with the OpenSSL command line interface. Almost half of the participants in the experiment assumed that they had succeeded in creating a valid self-signed certificate, although they had not. Only a fraction (19%) of participants were able to correctly validate the provided certificates, suggesting the usability of OpenSSL is also far from optimal.

Usability issues, as described above, are a major obstacle in getting TLS configurations and certificate handling right for developers, testers and administrators alike. However, security issues are not only the result of poor usability – they may be caused by institutional or organizational factors out of control of the administrators [14]. In the context of TLS certificates, Fahl et al. [17] found that almost two-thirds of the non-validating certificates were deployed deliberately.

One reason included the websites were never supposed to be publicly accessible (this manifested itself also in our study, see Section 3.2). Another reason for deploying self-signed certificates was to save money (also mentioned in our interviews, but only marginally).

However, even being able to deploy TLS correctly does not tell us much about IT professionals' perception of certificate flaws they encounter when connecting to other servers. In this respect, our work complements the previous research mentioned above.

5.2 End Users and TLS Certificates

In contrast to IT professionals, the perceptions of end users with respect to TLS warnings have been thoroughly investigated. Studies indicate that a considerable portion of end users does not understand the causes of certificate warnings. Sunshine et al. [38] surveyed 400 browser users in 2008 to investigate reactions and understanding of TLS warnings (expired, unknown issuer, domain name mismatch). They found that the majority of users did not understand the warnings. Interestingly, those users who understood the warning chose more often to adhere to name mismatch warnings but considered warnings for expired certificates as low risk (a similar trend is also present in our study in trust assessment, see Section 3.6).

A study by Felt et al. [18] investigated whether users can understand the threats they might be facing when seeing TLS warnings. The most users misidentified the threat source (more than 50% thought it was malware) and less than 20% understood what data was actually at risk (most overestimated the risk).

Reeder et al. [32] conducted a study on browser warnings to find out why users do not adhere to warnings. The most common reasons to ignore a TLS warning was connecting to a known trusted website, e.g., company-internal or their own (both opinions also arose in our study, see Section 3.2). Whereas warning adherence rates increased over time for major browsers, comprehension rates remain low and misconceptions are still a problem [18].

In summary, research shows that end users seem to lack the understanding of certificate warnings and their security implications. In most cases, IT professionals could set the appropriate decision already during system development. However, to do so, it is crucial that they understand the errors and security consequences themselves – which is why we decided to focus on IT professionals.

5.3 Documentation and Error Messages

Our attempts to improve the understanding of security consequences of certificate validation errors were based on re-worded error messages and redesigned documentation. Bralo-Lillo et al. [9] found that when end-user security dialogues get appropriately adjusted, significantly fewer people ignore clues indicating they may be at risk.

The importance of usable documentation is discussed by Georgiev et al. [19], stressing the importance of clean and consistent error reporting. Error handling is also seen as vital by Gorski and Lo Iacono [21]. In further work [22], they prototyped an API-integrated security advice system successfully nudging developers towards secure decisions.

Moreover, several works emphasized the crucial role of documentation for security API usability. Robillard argues [33] that documentation and code examples are an essential information source for developers to learn how to use an API. Concerning documentation in

the security context, Acar et al. [2] have shown that IT professionals resort to online forums and tutorials on the Internet, often helping them get the code functional but not necessarily secure. Therefore, providing easily accessible documentation with security information and examples of secure code is of crucial importance [1, 2].

6 DISCUSSION AND CONCLUSIONS

We conducted a study with 75 people working in IT to understand how they perceive flawed TLS certificates. Even though similar studies had been conducted for end users before, this is the first such study for IT professionals. We further investigated how we can influence comprehension and resource use by re-wording the error messages and redesigning documentation.

6.1 Trust Decisions Are Not Binary

From a security point of view, the decision to trust a certificate deployed by someone else is binary: if the certificate does not validate, it should not be trusted. Our study results show that the trust decisions of IT professionals are not necessarily binary and depend on the particular flaw and its context.

For example, for a majority of participants, the trust in expired certificates heavily depends on the time elapsed from the expiry date, indicating the expiration time may be used as a proxy to determine whether it was a misconfiguration or an attack. This was also evident in the qualitative results, where many participants mentioned that expired certificates are likely a mistake and rather common. This perception may be influenced by the actual reality of the Internet: 25% of websites with expired certificates are accessed a maximum of seven days after the expiration date, indicating that the problem is often fixed within one week [5]. Moreover, several participants suggested that the reputation of the certificate subject also plays a role.

Besides the fuzzy trust in the expired certificate, IT professionals seem to overly trust the self-signed and the name constrained certificates: both were rated similarly to the expired case even though the security implications are rather worse. Furthermore, we have to keep in mind that this may be only a lower bound on the trust in the broader population due to a social desirability bias (see Section 2.6). The trust evaluation of the name constraints case is especially worrying, as this case was also poorly understood. For the self-signed case, the context was important: expecting a self-signed certificate on a (known) server or using it for internal or testing purposes is seen as a less severe problem.

6.2 Security Implications

The security implications of trusting flawed certificates are different in different cases. For example, it may be acceptable for the system administrator to trust a flawed certificate deployed deliberately on their own server for testing purposes (the potential security consequences concern only themselves). However, for IT professionals developing applications used by (potentially) millions of end users, the situation is different as their decision impacts all the end users.

Delegating the decision to the end users is generally a bad idea since they tend to make uninformed decisions due to the lack of understanding (see Section 5.2). Moreover, given that certificate flaws are common, but mostly benign, end users could further lose their trust in the security of the TLS ecosystem [25, 36]. In most cases, IT professionals should, therefore, make security decisions during the development, testing and deployment process as they are capable of better-informed decisions.

Even though IT professionals have a better potential to make appropriate security decisions than end users, our study shows some still fairly trust invalid certificates: 21% of the participants (16 ♀) say the self-signed certificates *'looks OK'* or better; for name constraints certificate it is 20% (15 ♀). Thus, certificates with errors are still trusted by some of the IT professionals working with them.

While our work presents valuable insights into IT professionals' perceptions of flawed certificates, further research is needed to investigate the coping strategies people in IT use when encountering such certificates in the wild.

6.3 Error and Documentation Design Matters

We show that even a simple content redesign of the error messages and documentation matters: It positively influenced the comprehension of the errors and trust assessment of the certificates (see Sections 4.1 and 4.2). Both over-trusted cases (self-signed and name constraints) were rated lower and the comprehension in the name constraints case improved considerably (expressed by more frequent comprehension codes, less frequent incomprehension codes, lower task times and less Internet browsing).

The majority of participants with the opportunity (71%, 15 ♀) followed the link provided in the re-worded error message, pointing to an excellent and cost-effective design opportunity to lead users to a trusted unified documentation source.

6.4 Future Work

While we obtained new insights into participants' perceptions of flawed TLS certificates, more work is needed to investigate what influences these perceptions and what coping behavior they cause.

Nonetheless, we should design systems embracing the complexity of trust decisions IT professionals make instead of forcing them to choose a binary option. We should pay more attention to error message and error documentation design as it turns out that even simple content changes can have significant effects. The name constraints extension seems to be very poorly understood, suggesting its wider deployment might be problematic without extra educational efforts. We plan to propose a few simple patches to OpenSSL (and possibly other libraries), re-wording name constraints error message and the accompanying documentation. More significant changes, such as error messages linking to good documentation or rewriting the documentation to clearly state the security implications, may also be beneficial but would require a discussion with the developer community.

In conclusion, the TLS certificate ecosystem proves to be complicated enough to produce a wide variety of attitudes, opinions and misconceptions even among IT professionals. To improve the situation, designers need to knowingly strive for good developer usability.

REFERENCES

[1] Yasemin Acar, Michael Backes, Sascha Fahl, Simson Garfinkel, Doowon Kim, Michelle Mazurek, and Christian Stransky. 2017. Comparing the usability of cryptographic APIs. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, Piscataway, NJ, USA, 154–171. https://doi.org/10.1109/sp.2017.52

[2] Yasemin Acar, Michael Backes, Sascha Fahl, Doowon Kim, Michelle Mazurek, and Christian Stransky. 2016. You get where you're looking for: The impact of information sources on code security. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (S&P)*. IEEE, Piscataway, NJ, USA, 289–305. https://doi.org/10.1109/sp.2016.25

[3] Yasemin Acar, Christian Stransky, Dominik Wermke, Michelle Mazurek, and Sascha Fahl. 2017. Security Developer Studies with GitHub Users: Exploring a Convenience Sample. In *Proceedings of the 13th USENIX Conference on Usable Privacy and Security (SOUPS)*. USENIX Association, Berkeley, CA, USA, 81–95.

[4] Mustafa Acer, Emily Stark, Adrienne Felt, Sascha Fahl, Radhika Bhargava, Bhanu Dev, Matt Braithwaite, Ryan Sleevi, and Parisa Tabriz. 2017. Where the Wild Warnings Are: Root Causes of Chrome HTTPS Certificate Errors. In *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*. ACM Press, New York, NY, USA, 1407–1420. https://doi.org/10.1145/3133956.3134007

[5] Devdatta Akhawe, Bernhard Amann, Matthias Vallentin, and Robin Sommer. 2013. Here's my cert, so trust me, maybe?: Understanding TLS errors on the web. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*. ACM Press, New York, NY, USA, 59–70. https://doi.org/10.1145/2488388.2488395

[6] Devdatta Akhawe and Adrienne Felt. 2013. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *Proceedings of the 22nd USENIX Security Symposium*. USENIX Association, Berkeley, CA, USA, 257–272.

[7] Lujo Bauer, Cristian Bravo-Lillo, Lorrie Cranor, and Elli Fragkaki. 2013. *Warning design guidelines*. Technical Report CMU-CyLab-13-002. CyLab, Carnegie Mellon University. Retrieved 2019-06-01 from https://www.cylab.cmu.edu/_files/pdfs/tech_reports/CMUCyLab13002.pdf

[8] Robert Biddle, Paul van Oorschot, Andrew Patrick, Jennifer Sobey, and Tara Whalen. 2009. Browser Interfaces and Extended Validation SSL Certificates: An Empirical Study. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security (CCSW)*. ACM Press, New York, NY, USA, 19–30. https://doi.org/10.1145/1655008.1655012

[9] Cristian Bravo-Lillo, Saranga Komanduri, Lorrie Faith Cranor, Robert W. Reeder, Manya Sleeper, Julie Downs, and Stuart Schechter. 2013. Your Attention Please: Designing Security-decision UIs to Make Genuine Risks Harder to Ignore. In *Proceedings of the 9th USENIX Conference on Usable Privacy and Security (SOUPS)*. USENIX Association, Berkeley, CA, USA, 1–12. https://doi.org/10.1145/2501604.2501610

[10] CA/Browser Forum 2018. *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates*. Retrieved 2019-05-01 from https://cabforum.org/baseline-requirements-documents/

[11] Taejoong Chung, Yabing Liu, David Choffnes, Dave Levin, Bruce MacDowell Maggs, Alan Mislove, and Christo Wilson. 2016. Measuring and Applying Invalid SSL Certificates: The Silent Majority. In *Proceedings of the 16th ACM Internet Measurement Conference (IMC)*. ACM Press, New York, NY, USA, 527–541. https://doi.org/10.1145/2987443.2987454

[12] Jeremy Clark and Paul van Oorschot. 2013. SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy (S&P)*. IEEE, Piscataway, NJ, USA, 511–525. https://doi.org/10.1109/sp.2013.41

[13] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. 2008. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. RFC Editor.

[14] Constanze Dietrich, Katharina Krombholz, Kevin Borgolte, and Tobias Fiebig. 2018. Investigating System Operators' Perspective on Security Misconfigurations. In *Proceedings of the 25th ACM Conference on Computer and Communications Security (CCS)*. ACM Press, New York, NY, USA, 272–1289. https://doi.org/10.1145/3243734.3243794

[15] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS)*. ACM Press, New York, NY, USA, 542–553. https://doi.org/10.1145/2810103.2813703

[16] Serge Egelman and Eyal Peer. 2015. Scaling the security wall: Developing a security behavior intentions scale (SeBIS). In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, New York, NY, USA, 2873–2882. https://doi.org/10.1145/2702123.2702249

[17] Sascha Fahl, Yasemin Acar, Henning Perl, and Matthew Smith. 2014. Why Eve and Mallory (also) love webmasters: A study on the root causes of SSL misconfigurations. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (CCS)*. ACM Press, New York, NY, USA, 507–512. https://doi.org/10.1145/2590296.2590341

[18] Adrienne Felt, Alex Ainslie, Robert Reeder, Sunny Consolvo, Somas Thyagaraja, Alan Bettes, Helen Harris, and Jeff Grimes. 2015. Improving SSL Warnings: Comprehension and Adherence. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, New York, NY, USA, 2893–2902. https://doi.org/10.1145/2702123.2702442

[19] Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov. 2012. The most dangerous code in the world: Validating SSL certificates in non-browser software. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM Press, New York, NY, USA, 38–49. https://doi.org/10.1145/2382196.2382204

[20] Google 2019. *Transparency report: HTTPS encryption on the web*. Retrieved 2019-05-01 from https://transparencyreport.google.com/https

[21] Peter Gorski and Luigi Lo Iacono. 2016. Towards the Usability Evaluation of Security APIs. In *Proceedings of the 10th International Symposium on Human Aspects of Information Security and Assurance (HAISA)*. University of Plymouth, Plymouth, UK, 252–265.

[22] Peter Leo Gorski, Luigi Lo Iacono, Dominik Wermke, Christian Stransky, Sebastian Möller, Yasemin Acar, and Sascha Fahl. 2018. Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic API Misuse. In *Proceedings of the 14th USENIX Conference on Usable Privacy and Security (SOUPS)*. USENIX Association, Berkeley, CA, USA, 265–281.

[23] Ralph Holz, Lothar Braun, Nils Kammenhuber, and Georg Carle. 2011. The SSL landscape: A thorough analysis of the X.509 PKI using active and passive measurements. In *Proceedings of the 11th ACM Internet Measurement Conference (IMC)*. ACM Press, New York, NY, USA, 427–444. https://doi.org/10.1145/2068816.2068856

[24] R. Housley, W. Polk, W. Ford, and D. Solo. 2002. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 3280. RFC Editor.

[25] Katharina Krombholz, Karoline Busse, Katharina Pfeffer, Matthew Smith, and Emanuel von Zezschwitz. 2019. "If HTTPS Were Secure, I Wouldn't Need 2FA" – End User and Administrator Mental Models of HTTPS. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (S&P)*. IEEE, Piscataway, NJ, USA, 252–265.

[26] Katharina Krombholz, Wilfried Mayer, Martin Schmiedecker, and Edgar Weippl. 2017. I have no idea what I'm doing – on the usability of deploying HTTPS. In *Proceedings of the 26th USENIX Security Symposium*, Vol. 17. USENIX Association, Berkeley, CA, USA, 1339–1356.

[27] J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33, 1 (1977), 159–174. https://doi.org/10.2307/2529310

[28] Luigi Lo Iacono and Peter Leo Gorski. 2017. I Do and I Understand. Not Yet True for Security APIs. So Sad. In *Proceedings of the 2nd European Workshop on Usable Security (EuroUSEC)*, Vol. 17. Internet Society, Reston, VA, USA. https://doi.org/10.14722/eurousec.2017.23015

[29] Alena Naiakshina, Anastasia Danilova, Christian Tiefenau, Marco Herzog, Sergej Dechand, and Matthew Smith. 2017. Why Do Developers Get Password Storage Wrong? A Qualitative Usability Study. In *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*. ACM Press, New York, NY, USA, 311–328. https://doi.org/10.1145/3133956.3134082

[30] Matus Nemec, Dusan Klinec, Petr Svenda, Peter Sekan, and Vashek Matyas. 2017. Measuring popularity of cryptographic libraries in Internet-wide scans. In *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC)*. ACM Press, New York, NY, USA, 162–175. https://doi.org/10.1145/3134600.3134612

[31] OpenSSL 2019. *OpenSSL Wiki*. Retrieved 2019-05-01 from https://wiki.openssl.org

[32] Robert Reeder, Adrienne 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 36rd Annual ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, New York, NY, USA, 512. https://doi.org/10.1145/3173574.3174086

[33] Martin Robillard. 2009. What makes APIs hard to learn? The answers of developers. *IEEE software* 26, 6 (2009), 27–34. https://doi.org/10.1109/ms.2009.116

[34] Johnny Saldaña. 2015. *The coding manual for qualitative researchers* (3rd ed.). SAGE Publishing, Thousand Oaks, CA, USA.

[35] Margarete Sandelowski. 2000. Whatever happened to qualitative description? *Research in nursing & health* 23, 4 (2000), 334–340.

[36] Angela Sasse. 2015. Scaring and Bullying People into Security Won't Work. *IEEE Security & Privacy* 13, 3 (2015), 80–83. https://doi.org/10.1109/msp.2015.65

[37] Stack Overflow 2018. *Stack Overflow Developer survey*. Retrieved 2019-05-01 from https://insights.stackoverflow.com/survey/2018/

[38] Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Cranor. 2009. Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In *Proceedings of the 18th USENIX Security Symposium*. USENIX Association, Berkeley, CA, USA, 399–416.

[39] Martin Ukrop and Vashek Matyas. 2018. Why Johnny the Developer Can't Work with Public Key Certificates: An Experimental Study of OpenSSL Usability. In *Topics in Cryptology – CT-RSA 2018: The Cryptographers' Track at the RSA Conference 2018*. Springer International Publishing, New York, NY, USA, 45–64. https://doi.org/10.1007/978-3-319-76953-0_3
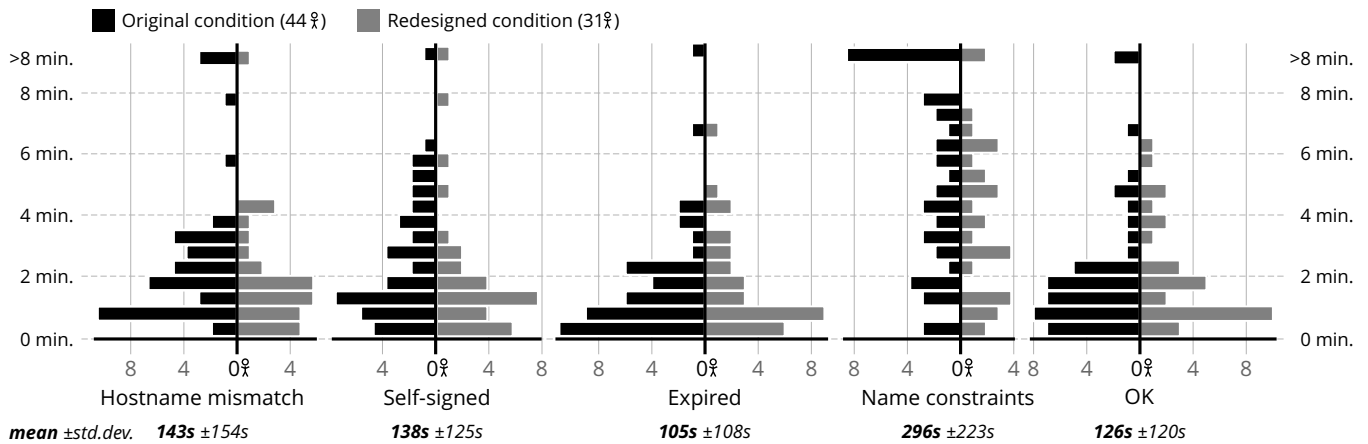
Figure 5: Histogram comparison of time spent on each certificate case, split by experimental condition. The vertical axis has task time divided into 30-second intervals and the horizontal axis shows the number of participants with time in the particular interval (all times over 8 minutes are merged for simplicity). Note that the number of participants is different for the two conditions.

A TASK DETAILS AND TIMING

This appendix contains the full formulations of the pre-task questionnaire, the task and its context (as given to the participant) and questions guiding the post-task semi-structured interview. Apart from that, detailed timing information for all the participants are summarized in Fig. 5.

A.1 Pre-task Questionnaire

To better put your opinions in context, we need to know what previous experience you have.

(1) How many years have you been employed in the IT sector? (enter '0' if not employed) *[number answer field]*
(2) Do you have a university degree in computer science? *[No formal education; Bachelor degree; Master degree; Postgraduate degree]*
(3) How would you describe your knowledge of computer security in general? *[Poor; Fair; Good; Very good; Excellent]*
(4) How would you describe your knowledge of X.509 certificates? *[Poor; Fair; Good; Very good; Excellent]*
(5) Which of these tools/libraries have you ever used? Tick all that apply. *[OpenSSL, GnuTLS, Network security services (NSS); Java Keytool; Windows Certutil; Let's Encrypt Certbot]*
(6) Have you participated in our last yearâĂŹs experiment here at DevConf? *[Yes; No]*

These questions were followed by the full Security Behavior Intentions Scale [16] with randomized question order.

A.2 Experiment Task

CONTEXT
- Imagine you want to improve DevConf registration system by allowing login with third-party accounts (GitHub, Fedora Project, Google, Microsoft, Facebook). You decide to contribute by coding it yourself and then submitting a pull request.
- Imagine, you have already created a small program testConnection.c testing the connection and validating server certificates.

However, some server certificates seem to be problematic. Now, you should check them out.

YOUR TASK
Use the testing program `testConnection.c` to validate the certificates of the servers. That is, for each server:
(1) Run the testing program using
   `./testConnection <server_name>`
(2) If the certificate is not valid, try to understand the problem and risks (read the documentation, tutorials, ...).
(3) Decide how much do you trust the server having this certificate? (Fill in the provided form.)

DETAILS
- No coding or bug fixing is required from you, the program `testConnection.c` does the verification correctly (showing you all occurred errors).
- You (ultimately) trust the *DevConf Root CA* certificate authority.
- You can view the downloaded certificates using
   `certtool -i --infile <chain.pem>`
- When trying to understand the problem, you can use any resources you want (manual pages, tutorials, Google, ...).
- Server order does not matter, but investigate the next server only after finishing the previous – we measure individual times.
- The testing program `testConnection.c` is located in `/Documents`, start your task by running it in the terminal with no arguments (displays help).

A.3 Trust Evaluation Scale

How much do you trust the server having this certificate?
- Trust 6 / 6 (I'm totally satisfied. If it were my bank's website, I would log in without worries.)
- Trust 5 / 6
- Trust 4 / 6 (Looks OK. I would log in with my library account, but not with my bank account.)
- Trust 3 / 6

Table 2: Certificate validation messages in the original and redesigned conditions.

Case	Original error message	Redesigned error message
Hostname mismatch	hostname mismatch (X509_V_ERR_HOSTNAME _MISMATCH)	The server hostname does not match the certificate subject name. (X509_ERR_HOSTNAME _MISMATCH, see http://x509errors.cz)
Self-signed	self signed certificate (X509_V_ERR_ DEPTH_ZERO_SELF_SIGNED_CERT)	The certificate is self-signed and not found in the trust store. (X509_ERR_SELF_SIGNED, see http://x509errors.cz)
Expired	certificate has expired (X509_V_ERR_CERT_HAS_EXPIRED)	The certificate has expired or is not yet valid. (X509_ERR_EXPIRED, see http://x509errors.cz)
Name constraints	permitted subtree violation (X509_V_ERR_PERMITTED_VIOLATION)	The subject name violates constraints set by CA. (X509_ERR_NAME_CONSTRAINTS_VIOLATION, see http://x509errors.cz)
OK	ok (X509_V_OK)	All performed checks passed. (X509_OK, see http://x509errors.cz)

- Trust 2 / 6 (Looks suspicious. I will read the page, but I will not fill in any information.)
- Trust 1 / 6
- Trust 0 / 6 (Outright untrustworthy. It is not safe to browse or to trust any information there.)

A.4　Post-task Interview

Now please describe in your own words what was the problem with the certificates case by case. (Fedora project, Facebook, GitHub, Google, Microsoft)

Note: This open-ended question usually took over half of the interview time. The interviewers prompted the participant to elaborate further as necessary. Afterward, the following closed questions were posed if not answered spontaneously during the open-ended discussion.

- Did you check for how long is the certificate expired? (Yes, No)
- Does it make a difference for you how long are they expired? (Yes, No)
- If yes, how would you rate the certificate expired... (yesterday, a week ago, a month ago, a year ago)
- Do you know what it means to revoke the certificate?
- Did you know that expired certificates are no longer included in Certificate Revocation Lists? (Yes, No)
- If not, does this change your trust rating?
- Which one was the most difficult to understand? Why? (Self-signed, name mismatch, OK, Expired, Name constraints)
- Can you name any examples of exceptionally well or exceptionally poorly done error messages you encountered in your life?
- Which of the following parts should it have? (Error number, Text code, Human-readable description)
- Imagine you can have unlimited time and resources to change anything regarding error reporting and the developers of the world would comply. What would you want?

B　REDESIGNED DOCUMENTATION

This appendix contains the full wording of the original and redesigned error messages (see Table 2) and the redesigned documentation that was available during the experiment at x509errors.cz. A link to this website is the part of the redesigned error messages.

B.1　Hostname Mismatch Case

The server hostname does not match the certificate subject name. (X509_ERR_HOSTNAME_MISMATCH)

EXPLANATION: The domain name provided by the server you are connecting to does not match the subject name of the certificate.

SECURITY PERSPECTIVE: Your communication will be encrypted, but you communicate with a different (maybe malicious) server from the one listed in the certificate. However, it can also be caused by malicious attackers pretending to be the server you are connecting to.

NEXT STEPS: See the Common Name (CN) or the Subject Alternative Name extension (SAN) in the certificate and compare the value with the domain name of the server. In case of web servers, the error can be caused by improper redirect configuration between valid web aliases (e.g., the version of the site without the "www" in the domain name).

B.2　Self-signed Case

The certificate is self-signed and not found in the trust store. (X509_ERR_SELF_SIGNED)

EXPLANATION: A self-signed certificate is signed by the same entity whose identity it certifies. In technical terms, a self-signed certificate is one signed with its own private key. This is common for root keys (usually pre-installed on your system).

SECURITY PERSPECTIVE: No certificate authority issued the certificate. Anyone can create a certificate with these parameters. It means a really low level of security and trust. Since this self-signed certificate is not "managed" by a CA, there is no possible revocation.
It can be trusted only if you control every machine between you and the server or you check that the keys in the certificate are what you expect them to be. However, it can also be caused by malicious attackers pretending to be the server you are connecting to.

NEXT STEPS: If you can, try to validate the certificate in a different way âĂŞ compare the fingerprint with what is expected, compare it with the certificate obtained on previous visits, etc.

B.3　Expired Case

The certificate has expired or is not yet valid. (X509_ERR_EXPIRED)

EXPLANATION: Most of the certificates have a defined period of validity (the "not before" and "not after" validity fields). The expiration date is specified by the issuing certificate authority, usually for 1–2 years (there could be cases with shorter or longer validity).

SECURITY PERSPECTIVE: Your communication will be encrypted, but the identity of the server stated in the certificate is now no longer guaranteed by the trusted certificate authority.

However, the certificate revocation status is no longer published. That is, the certificate might have been revoked long ago, but it will no longer be included in the CRL. That means you cannot know if the certificate is already revoked or not.

NEXT STEPS: In some cases, the error may be caused by the misconfiguration on the client side – make sure your local time settings are correct. If you are responsible for the certificate, contact the company that issued it to learn how to renew it. If not, inform the appropriate administrator.

B.4   Name Constraints Case

The subject name violates constraints set by CA. (X509_ERR_NAME_CONSTRAINTS_VIOLATION)

EXPLANATION: By default, all certificate authorities can issue certificates for any subjects. One way to constrain this risk of issuing a malicious certificate is to limit CAs to issue only for certain names, using the "name constraints" extension for X.509 certificates. The occurred error says that the authority in the chain was prohibited from issuing the certificate for that particular subject.

SECURITY PERSPECTIVE: The certificate authority was not allowed to issue the certificate it issued. This suggests suspicious activity at the authority level.

NEXT STEPS: See the Name Constraints extension in the certificate authority and compare the allowed/restricted name ranges with the Common Name (CN) or the Subject Alternative Name extension (SAN) in the issued certificate. Report the certificate to the issuing authority.

B.5   OK Case

All performed check passed. (X509_OK)

EXPLANATION: No errors occurred during the performed validation tasks. Make sure that everything you wanted to check was checked.

SECURITY PERSPECTIVE: Everything seems OK.

NEXT STEPS: Check the hostname, the revocation status (by means of CRL or OCSP) check having appropriate verification flags.