# On Tools for Socio-Technical Security Analysis

Ana Ferreira*†, Rosario Giustolisi*, Jean-Louis Huynen*†, Gabriele Lenzini*
* Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
† Education Measurement and Applicative Cognitive Science (EMACS), University of Luxembourg

Many systems are hacked daily and apparently without much effort (e.g., see [1]). This happens because hackers prefer not to break security mechanisms immediately, but rather to target unguarded components first. Such components, e.g., users and human-computer ceremonies [2], are hacked by exploiting cognitive features (e.g., trust) and people's dismay with ill-designed interfaces. These user-related components are often ignored in traditional security analysis. Thus, it should not surprise that systems proved secure may fail especially when they run in different contexts from those wherein they have been proven secure.

We are interested to define a framework where to model and analyse a system's social and technical components. We describe here a variant of Bella *et al.*'s model [3]. Therein Alice and Bob are not metaphors for communicating processes, but personae linked a set of interaction layers (see Fig. 1 and its caption) that connect humans and computers and, via the network, them with other computers and users. On top of this model we define an intruder. It controls the network, as in classical Dolev-Yao [4], and also the application, the user interfaces, and the context. When using its full power the intruder can influence the components and the user behaviour, and so security depends on what happens across all layers: the analysis of security results richer, and we talk of *socio-technical security* analysis.

Studying socio-technical security compels us to revise traditional analysis techniques. Depending on the focus of the analysis, in fact, we may need different methodologies and tools. An analysis focusing more on the technical side (communicating processes, applications and interfaces) and with attackers controlling the networks and/or the interfaces, requires tools to reason about the behaviour of software components. An analysis addressing more the social side (persona and user behaviour) requires to observe and reason about users interacting with the system, so a research methodology proper of social and cognitive sciences.

In the sequel, we comment on methodologies and tools that we evaluated and selected in two experiments concerning socio-technical understanding of the security of TLS certificate validation. We have successfully applied formal methods (model checking) when considering layers "network" and "computer" (cf. Figure 1), and when assuming users behave non-deterministically (Sect I). This confirms Martina and Carlos's saying that formal methods are useful to analyse security ceremonies [5]. To study instead security with more realistic user models we had to look into research methods of human-computer interaction (HCI) and cognitive science practices; because there is no framework for security analysis in those practises, we had to define one of our own (Sect II).

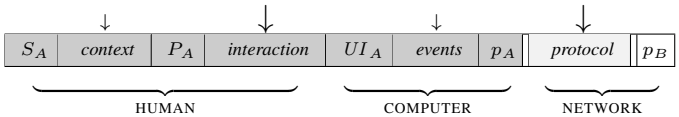


Fig. 1. The multi-layered security and threat model. Alice's side is in grey: $S_A$ is Alice's self, that contextualises in Alice's persona in a specific context, $P_A$. $UI_A$ is the user-interface she uses to interact with the networked application $p_A$. It runs a protocol with $p_B$, an element of Bob's side, here not fully shown. Arrows are possible attacks. The intruder can control context, interaction, events between application and user interface, and the network.

## I. Experiment 1 - TLS Certificate Validation

Authentication of a web server relies on TLS certificates, and it succeeds if the browser validates the server's TLS certificate. But, if the browser can not validate the certificate –e.g., because this is self-signed by the server and not by a trusted authority– authentication may depend on the user: often browsers let him to decide whether to proceed or abort the session. Thus, TLS certificate validation is a socio-technical procedure made of communicating processes (the browser engine and the server), user interfaces (the browser's window and the options offered thereon), and a persona (the user).

In this experiment we studied the security of TLS certificate validation for four of the most popular browsers: Chrome, Firefox, Internet Explorer, and Opera Mini. We modelled the browsers, their interfaces and a simple model of user that chooses, non-deterministically, among the options offered him by the browser. Browsers run different engines and ceremonies with users, so the analysis –whose focus is on the structure of the dialogue browser-user– is rich in possibilities.

We explored methodologies and tools to carry on our analysis. To model all the layers involved in TLS certificate validation, we first tried different graphic formalisms, like cognitive walk-through and flow charts, but their lacking a formal semantics precludes any formal analysis. Eventually, we chose UML activity diagrams. The contribution of UML activity diagrams is threefold. First, they fit with the layered representation of the ceremony. Second, they give immediately an easy reading of TLS sessions; in fact, a quick glance at the diagrams of the browsers under study shows clearly their different validation mechanisms. Third, despite having a semi-formal semantics it is not difficult to translate them in a formal language. A prototyped tool that translates UML activity diagrams in CSP (Communicating Sequential Processes) [6] will be available soon [7].

To carry on a formal analysis, we translated the UML diagrams in CSP#, which is a richer language based on CSP. We also modelled an intruder, here a Dolev-Yao controlling the network (the large arrow on the right in Fig. 1), and the user. Capturing the complexities of user behaviour by a formal model is a challenging open issue. As explained in the introduction, we modelled the user as a non-deterministic process: this is the weaker assumption about the user skills:

a ceremony that is secure for a non-deterministic user, is also secure for any user.

The last step of our prototype methodology consisted in defining relevant security properties. We identified four socio-technical properties that bind TLS session, validation mechanisms, and user choices. We expressed them in linear temporal logic. One property is meant to evaluate the user involvement: it assesses whether the browser always warns the user when certificate validation fails. Two properties aim to evaluate whether the mechanisms that browsers adopt to manage failed certificate validations protect user from man-in-the-middle (MIM) attacks (e.g., if they avoid that he proceeds accessing a page controlled by the intruder). The last property is about informing the user that a MIM attack might have occurred in previous TLS sessions.

We verified the properties with the PAT (Process Analysis Toolkit) model checker [8]. The most interesting results regard Firefox. PAT reports a trace showing that Firefox does not warn the user when a certificate validation fails. This is due to the drawbacks of storing server certificates permanently, which Firefox allows its users to do. Moreover, it is worth noting that no browser keeps records of past warnings, exposing users to vulnerabilities when they bootstrap with MIM. This finding suggests a novel, more secure, strategy for browsers.

## II. Experiment 2 - Human Behavioural Model

This second experiment is about on-going work. Building a more realistic user behaviour requires tools and research methods commonly employed in experimental psychology like surveys, diaries, focus-groups, interviews or non-interfering observations in a laboratory setting. Those tools allow us to build behavioural patterns by studying quantitative and qualitative aspects of the human behaviour in a scenario with socio-technical attacks. We adopt our multi-layered security

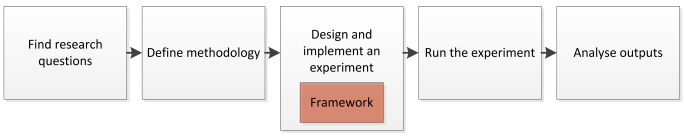


Fig. 2. HCI Research Methods, with Socio-Technical Analysis framework.

and threat model and we assume that socio-technical attacks may strike as indicated in Fig. 1. To study how users behave therein, we use the traditional HCI's hypothetical-deductive research process that we have re-adapted to our needs (see Fig. 2). The process starts with us stating hypothesis (research questions) about an attack, defence(s), user, application, and context. One problem we met is that there is no work-flow helping us to design and implement laboratory experiments that stress hypothesis of this kind: we have to design one. Starting from the multi-layered security and threat model, we identified key components in User, Application and Context (basic components) to which we added Attack and Defence. All those elements but Defence are mapped to elements of our multi-layered model in Fig. 1. Attack and defence may compete: the former pushing users to an insecure behaviour, the latter, if present, to a secure behaviour. What secure/insecure behaviour mean, is defined in the methodology.

Every component has a state and acts with input/output actions according to a Behaviour Control Process (BCP). The state, like the BCP, can be very complex and we may not be able to formalize it fully. For instance, a browser has its code as BCP (as seen in Section I) but we do not have such things for the user. However, we can inquire properties of that behaviour by observing it during the laboratory experiment. For instance if the user reads some information on a website we know that he inputs something, but we do not know if he has learnt something until we observe his following actions.

We now apply this framework to the TLS certification verification. We know that users already ignore 60% of Interstitial Warnings (IW) in Google Chrome [9] and this rate may increase if an Attack changes the user's "state" just before he makes a choice. An Attack controlling the user interface, for example a browser-in-the middle (the big arrow on the left in Fig. 1), can add a Fake IW before the genuine warning to misled the user in interpreting self-signed certificate as SELF-signed certificate, that is "certificated signed by SELF", where SELF is the name of a certification authority yet unknown by the user's browser, but introducing itself as trusted in the text of the IW. This introduces a polysemy on the word "self" that may lead the user to misinterpret the meaning of the word (called equivocation fallacy). To test whether people's knowledge of TLS certificates is robust enough to resist such misleading inputs we need to instantiate our framework (i.e., to decide how to launch the attack, what and how to observe, what to ask users afterwards, etc.)

Further experiments are needed to observe the patterns that lead users to fail or to resist the attack. This is future work that we plan to do in the EMACS usability laboratory.

## III. Conclusion and Future Work

We presented a framework where to analyse socio-technical security of systems, and we commented on methodologies that, in our experience, are apt for this analysis. However, we need to test our framework with more use scenarios.

We thank G. Bella, and V. Koenig for their help with the experiments herein described.

## References

[1] CIRCL, "A Perspective of Computer Incident In Luxembourg," Computer Incident Response Centre Luxembourg, Tech. Rep., 2011.
[2] C. Ellison, "Ceremony Design and Analysis," Cryptology ePrint Archive, Report 2007/399, Tech. Rep., 2007.
[3] G. Bella and L. Coles-Kemp, "Layered Analysis of Security Ceremonies," in *Proc. of the 27th IFIP Int. Conf. on Security and Privacy, 4-6 June 2012, Crete, Greece*, 2012.
[4] D. Dolev and A. Yao, "On the security of public-key protocols," *IEEE Transaction on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
[5] J. E. Martina and M. C. Carlos, "Why Should We Analyse Security Ceremonies?" in *Proc. of CryptoForma, May 25, 2010, Paris, France*, 2010.
[6] A. R. Hoare, *Communicating Sequential Processes*. Prentice Hall International, 1985.
[7] I. Abdelhalim, S. Schneider, and H. Treharne, "An integrated framework for checking the behaviour of fUML models using CSP," *International Journal on Software Tools for Technology Transfer*, 2012.
[8] J. Sun, Y. Liu, J. S. Dong, and J. Pang, "PAT: Towards Flexible Verification under Fairness," in *Proc. of CAV'09*, ser. LNCS, vol. 5643. Springer, 2009, pp. 709–714.
[9] A. Langley, "Living with https," July 2012. [Online]. Available: http://www.imperialviolet.org/2012/07/19/hope9talk.html