Improving the Usability of Cybersecurity Certification and Assessment Tools through Expert Walkthroughs and Heuristic Evaluation

Deepthi Mungara, Harshini Sri Ramulu, Rachel Gonzalez Rodriguez, Ali Cem Kaynak, Mariia Bakhtina, Yasemin Acar
Paderborn University and Masaryk University

Abstract

Cybersecurity regulation landscapes are getting increasingly complex. The Cybersecurity Act (CSA) sets to develop an EU-wide cybersecurity certification framework for information and communication technology (ICT) products, services, and processes and the Cyber Resilience Act (CRA) defines mandatory cybersecurity requirements for manufacturers, importers, and distributors of products or software with a digital component. Yet, putting CSA and CRA into practice faces significant coordination challenges, and implementation acts fulfilling the regulation objectives are emerging in a slow and cumbersome way. Therefore, a reliable and flexible suite of tools is needed for continuous security assessment. From a usable security perspective, these challenges are further amplified by the gap between regulatory requirements and the practical usability of tools.

To address this gap, we conduct systematic usability research to help transform four academic research tools into tools intended for use in industry and regulatory contexts: **TLS-Scanner, SCRUTINY, ALVIE, and sec-certs**. We report on usability evaluation using expert walkthroughs and heuristic evaluation methods to assess the four early-stage tools. We identified 307 usability issues, and systematically advised tool developers on how to efficiently improve their tools' usability.

1 Introduction, Related Work, and Background

Cybersecurity certification is increasingly seen as a key mechanism for risk mitigation in complex systems, though the existing frameworks still face challenges in practical applicability [7, 14, 17]. Prior work further shows that practitioners often find certification processes inefficient highlighting structural limitations in current practices [7]. Also studies shown that usability is a critical factor in security effectiveness beyond end users. Acar et al. emphasize that security outcomes depend heavily on how tools and APIs are designed and used; empirical studies further demonstrate that poor usability leads to incorrect implementations and potential vulnerabilities [1–3]. There is a limited empirical research focusing specifically on the usability of cybersecurity certification and compliance assessment tools. Our study addresses this gap by evaluating such tools from a usable security perspective.

We conducted expert walkthroughs followed by heuristic evaluations as the appropriate methodology to assess usability problems and identify possible improvements in the reported period.

Cognitive expert walkthroughs link the interface walkthrough to a cognitive model [13]. The evaluator uses the interface to perform tasks that a typical interface user will need to accomplish. This method is referred to as "walkthrough" throughout the remainder of this document.

Heuristic evaluation is a method where a small group of usability experts inspect an interface and judge its compliance against a set of established usability principles (i.e., heuristics), identifying violations that are likely to cause problems for users [15].

Cybersecurity Certification and Assessment Toolset: We analyze four tools, detailed descriptions of the toolset can be found in the Appendix. The tools are **TLS-Scanner**, an automated analysis tool for evaluating the security behaviour of TLS clients and servers [4, 11]; **SCRUTINY**, a toolset for analysing the correctness and performance of cryptographic implementations in hardware platforms [9, 18]; **ALVIE**, an automated tool for security analysis of embedded hardware architectures, bridging the gap between formal verification and vulnerability research [5,6]; and **sec-certs**, a tool designed to increase transparency of certification documents from the Common Criteria, EUCC, and FIPS 140 certifications and associated processes [8, 10, 12]

As the tools were initially developed for academic purposes, the developers observed usability as a key barrier to broader adoption.

2 Methodology

In this study, six usability experts with senior research experience in usable security conducted the evaluation in close collaboration with the tool development teams of TLS-Scanner, SCRUTINY, ALVIE, and sec-certs.

Expert walkthroughs: We conducted expert walkthroughs for all four tools in a way that the developers of the tools led through the use of the tool, while the usability team were the evaluators. First, the tool developer described the purpose of the tool and the potential users whose tasks might align with it. Then, the developers walked through every step of the tool while fulfilling its purpose. They also showcased help and documentation, if present. The usable security research team took notes, asked clarification questions, and produced a report with concrete suggestions and solutions in collaboration with the developers during the session itself.

Heuristic evaluation: After the walkthroughs, we conducted a heuristic evaluation of all 4 tools [16]. For ALVIE, this was necessarily supported by ALVIE developers, given the tool's complexity. For SCRUTINY, the developers were asked to validate the heuristic evaluation results to confirm the correct use of the tool, given the low maturity level and the assumption of high user expert knowledge. For the other tools (sec-certs and TLS-Scanner), the heuristic evaluation was done by the usable security research team without the involvement of the tools' developers. SCRUTINY consists of 9 sub-tools, of which 8 are testable.

As the tools in the project are of different maturity levels, the application of usability methods has been slightly adjusted to accommodate each tool's specifics.

3 Results

Based on the walkthrough and heuristic evaluation reports, we identified several recurring usability issues across the CCAT tools. While most of the CCAT tools (namely TLS-Scanner, ALVIE, and most SCRUTINY sub-tools) are command-line interface (CLI) tools with a low maturity level, the analysis revealed common issue patterns :

Documentation and guidance: The evaluation revealed several common usability issues related to documentation and guidance. Users often needed to consult the associated research paper to interpret tool outputs, suggesting that the auto-generated wiki alone was insufficient as a standalone source of guidance. Also, only a limited range of use cases was documented, with features typically demonstrated through narrow or single scenarios. CLI options and configuration parameters were also inconsistently documented outside of the command-line help function. Several options were only discoverable via CLI help and were not clearly described in the main documentation.

Handling errors: The evaluation identified several recurring usability issues related to error handling and reliability of feedback. In multiple cases, error states did not provide actionable information for resolving issues. Instead, users were presented with messages or outputs that did not clearly indicate how to correct the underlying problem. Timeouts and server disconnections further affected reliability, as results could still be interpreted as valid even when incomplete or failed, with no explicit distinction between successful and unsuccessful executions. Also, the tools were sensitive to input formatting. Minor deviations such as typos, lowercase input, or trailing punctuation could lead to incorrect or unexpected results.

Information presentation and visual consistency: The evaluation identified several inconsistencies in terminology and interface elements across the tools. Navigation labels and destination pages were sometimes mismatched, and similar actions were labeled differently in different parts of the interface. Also, some key content sections were visually underemphasized with low-contrast colors that made it blend into surrounding content. In some cases, the color usage was inconsistent across results. Colors were used to encode different types of information, but their meaning was not always clear or consistently applied.

Ease of use for the first-time user: The evaluation showed several usability issues affecting first-time users, particularly related to visibility and navigation clarity. Interactive elements were often visually ambiguous, and the lack of clear distinction between clickable and non-clickable elements reduced interface usability. In addition, the tools did not provide accessible help, FAQ, or explanatory documentation, increasing the learning burden for new users. The search experience also showed limitations, as users could not modify or refine their search once submitted.

4 Recommendations

Based on the identified usability issues, the following recommendations are proposed for improving the tool suite. The documentation should be extended to wiki and tutorials that supports task completion. Well-documented use cases should be added to illustrate realistic applications. CLI commands should include clearer explanations of parameters, inputs, and expected behavior. Error handling should provide clearer input specifications and actionable feedback for invalid inputs. Error messages should explain failures in a way that supports interpretation of result reliability. Errors should also be prioritized by severity to improve clarity in reporting. Terminology and interface elements should be made consistent across tools. Visual design should clearly distinguish interactive from non-interactive elements. The meaning of color usage should be explicitly defined and applied consistently. Clickable elements and navigation behavior should be made more visually explicit. A dedicated help or FAQ section should be provided to support on-boarding. Search functionality should be improved to support easier information discovery and refinement.

References

[1] Yasemin Acar, Michael Backes, Sascha Fahl, Simson Garfinkel, Doowon Kim, Michelle L Mazurek, and Christian Stransky. Comparing the usability of cryptographic apis. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 154–171. IEEE, 2017.

[2] Yasemin Acar, Michael Backes, Sascha Fahl, Simson Garfinkel, Doowon Kim, Michelle L. Mazurek, and Christian Stransky. Comparing the usability of cryptographic apis. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 154–171, 2017.

[3] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. You are not your developer, either: A research agenda for usable security and privacy research beyond end users. *2016 IEEE Cybersecurity Development (SecDev)*, pages 3–8, 2016.

[4] Fabian Bäumer, Marcus Brinkmann, Nurullah Erinola, Sven Hebrok, Nico Heitmann, Felix Lange, Marcel Maehren, Robert Merget, Niklas Niere, Maximilian Radoy, et al. Tls-attacker: a dynamic framework for analyzing tls implementations. *Proceedings of Cybersecurity Artifacts Competition and Impact Award (ACSAC 2024)*, 2024.

[5] Matteo Busi. Alvie tool. https://github.com/unive-alvie/alvie, 2024. Accessed: 2026-05-15.

[6] Matteo Busi, Riccardo Focardi, and Flaminia Luccio. Bridging the gap: Automated analysis of sancus. In *2024 IEEE 37th Computer Security Foundations Symposium (CSF)*, pages 233–248. IEEE, 2024.

[7] Gabriel Ferreira. Software certification in practice: How are standards being applied? In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 100–102. IEEE, 2017.

[8] Centre for Research on Cryptography and Security (CRoCS). About sec-certs. https://sec-certs.org/, 2025. Accessed: 2026-05-15.

[9] Centre for Research on Cryptography and Security (CRoCS). Scrutiny. https://github.com/crocs-muni/scrutiny/, 2025. Accessed: 2026-05-15.

[10] Centre for Research on Cryptography and Security (CRoCS). Sec-certs. https://github.com/crocs-muni/sec-certs, 2025. Accessed: 2026-05-15.

[11] System Security group. Tls-scanner. https://github.com/tls-attacker/TLS-scanner, 2025. Accessed: 2026-05-15.

[12] Adam Janovsky, Łukasz Chmielewski, Petr Svenda, Jan Jancar, and Vashek Matyas. Revisiting the analysis of references among common criteria certified products. *Computers & Security*, 152:104362, 2025.

[13] Thomas Mahatody, Mouldi Sagar, and Christophe Kolski. State of the art on the cognitive walkthrough method, its variants and evolutions. *Intl. Journal of Human–Computer Interaction*, 26(8):741–785, 2010.

[14] Sara N. Matheu, José L. Hernández-Ramos, Antonio F. Skarmeta, and Gianmarco Baldini. A survey of cybersecurity certification for the internet of things. *ACM Comput. Surv.*, 53(6), December 2020.

[15] Jakob Nielsen. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 373–380, 1992.

[16] Jakob Nielsen. Finding usability problems through heuristic evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, page 373–380, New York, NY, USA, 1992. Association for Computing Machinery.

[17] Jason R. C. Nurse, Sadie Creese, Michael Goldsmith, and Koen Lamberts. Guidelines for usable cybersecurity: Past and present. In *2011 Third International Workshop on Cyberspace Safety and Security (CSS)*, pages 21–26, 2011.

[18] Petr Svenda, Rudolf Kvasnovskỳ, Imrich Nagy, and Antonin Dufka. Jcalgtest: Robust identification metadata for certified smartcards. In *SECRYPT*, pages 597–604, 2022.

A Tool Descriptions

TLS-Scanner is an automated analysis tool for evaluating the security behaviour of TLS clients and servers [4, 11] . It is based on TLS-Attacker, a framework introduced in 2016 to enable systematic testing of TLS libraries through precisely defined TLS protocol flows. While TLS-Attacker executes a single, developer-specified TLS flow, TLS-Scanner extends this approach by automatically executing and evaluating multiple TLS flows to analyse implementation behaviour in a comprehensive and repeatable manner.

SCRUTINY is a toolset for analysing the correctness and performance of cryptographic implementations in hardware platforms such as smartcards and Trusted Platform Modules (TPMs), Hardware Security Modules (HSMs), and software cryptographic libraries [9, 18]. These implementations are often deployed as black-box systems without source code or detailed vendor documentation. SCRUTINY performs automated probing of cryptographic implementations to collect

data on supported algorithms, performance behaviour, key generation, digital signatures, and random number generation, followed by statistical analysis to detect anomalous patterns and visualization to support expert assessment.

ALVIE is an automated tool for security analysis of embedded hardware architectures, bridging the gap between formal verification and vulnerability research [5, 6]. It was developed to evaluate Sancus, an embedded security architecture that remains susceptible to implementation bugs. ALVIE constructs formal models of hardware systems using black-box active automata learning and applies model checking to detect potential side-channels, enabling formal verification of security properties on real implementations.

sec-certs is a tool designed to increase transparency of certification documents from the Common Criteria, EUCC, and FIPS 140 certifications and associated processes [8, 10, 12]. It collects, processes, and structures publicly available documents from security certifications artifacts into machine-readable data. It helps to link the certified products to publicly listed vulnerabilities (CVEs), expose unavailable data, show inconsistent or hidden relationships between certified products, and errors in existing data. The tool can be accessed via API or a web interface at sec-certs.org, and provides cleaned, structured data for further analysis and verification.