

## 1. Introduction

- Popular fields: Plugin development in modern software ecosystems (Overview of plugin development and its significance in modern software ecosystems)
- Introduction to security challenges in plugin development
- Thesis statement and outline (Description of the research objectives and goals of the thesis)

## 2. Background

- Overview of programming languages and their use in plugin development and how plugin developed in the history
- Discussion of the concepts and principles of access control in software systems.
- The consequences of access control vulnerabilities in plugins.
- Description of the IntelliJ platform and its architecture.
- Description of the VScode platform and its architecture

## 3. Related Work

- Survey of existing research on security issues in plugin development
- Introduction to language-based security mechanisms and their relevance to plugin development.
- Review of literature on language-based security mechanisms and their effectiveness in preventing access control vulnerabilities in software.
- Analysis of the strengths and limitations of prior research
- Discussion of current practices and tools for evaluating security in plugin development.

## 4. Evaluating Language-Based Security Mechanisms for Plugin Development

- Overview of the approach used to evaluate language-based security mechanisms for plugin development.
- Description of the test scenarios used to evaluate the effectiveness of language-based security mechanisms.
- Development of IntelliJ Plugins to Test Access Control Vulnerabilities
  - Description of the approach used to develop the test plugins.
  - Discussion of the design and implementation of the test plugins
- Evaluation of the Test Plugins
  - Analysis of the results of the tests performed using the test plugins.
  - Discussion of the implications of the test results for improving plugin security.
- Case studies evaluating language-based security in plugins developed using popular IDEs such as IntelliJ and VScode
- Discussion of the implications of the study for the security of plugin development
- Evaluation of the effectiveness of different programming languages and language-based security mechanisms in plugin development using case studies and real-world examples.
- Evaluation of the effectiveness of language-based security mechanisms in preventing

access control vulnerabilities in plugins

- Analysis of the experimental results and findings

## 5. Capability-Based Security for Plugin Development: A Comparative Study

- Introduction to capability-based module systems like Wyvern.
- Overview of the study design and methodology
- Comparison of the effectiveness of different programming languages and security mechanisms
- Analysis of Capability-Based Module Systems in Addressing Access Control Vulnerabilities
- Analysis of which access control vulnerabilities identified in previous sections can be addressed by capability-based module systems.
- Discussion of the potential benefits and drawbacks of using capability-based module systems in plugin development

## 6. Concluding Remarks

### 6.1 Conclusion

- Summary of the key findings and contributions of the research
- Discussion of the limitations of the study and future research directions

### 6.2 Future Work

- Discussion of future research directions in language-based security for plugin development
- Recommendations for improving the security of plugins using language-based security mechanisms



1. Rodrigo. Santos, Cludia. Werner, Olavo. Barbosa and Carina. Alves, "Software Ecosystems: Trends and Impacts on Software Engineering," 2012 26th Brazilian Symposium on Software Engineering, Natal, Brazil, 2012, pp. 206-210, doi: 10.1109/SBES.2012.24.
2. Rupp, S. & Daum, V.. (2006). Design and implementation aspects for plugin-base software frameworks. 5. 425-432.
3. Konstantinos Manikas, Klaus Marius Hansen, Software ecosystems – A systematic literature review, Journal of Systems and Software, Volume 86, Issue 5, 2013, Pages 1294-1306, ISSN 0164-1212, <https://doi.org/10.1016/j.jss.2012.12.026>.
4. Inamdar, Danish & Gupta, Shyam. (2020). A Survey on Web Application Security. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 223-228. 10.32628/CSEIT206543.
5. Li, X., & Xue, Y. (2011). A survey on web application security. Nashville, TN USA, 25(5), 1-14.
6. Fonseca, José & Vieira, Marco. (2013). A Survey on Secure Software Development Lifecycles. Software Development Techniques for Constructive Information Systems Design. 1. 57-73. 10.4018/978-1-4666-3679-8.ch003.
7. Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): a survey. Security and communication networks, 9(18), 5803-5833.
8. Tosi, Christian & Zaroni, Marco & Maggioni, Stefano. (2009). A Design Pattern Detection Plugin for Eclipse.
9. Huerta M, Caballero-Hernández JA, Fernández-Ruiz MA. Comparative Study of Moodle Plugins to Facilitate the Adoption of Computer-Based Assessments. Applied Sciences. 2022; 12(18):8996. <https://doi.org/10.3390/app12188996>
10. Y. Wang et al., "An Empirical Study of Usages, Updates and Risks of Third-Party Libraries in Java Projects," 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME), Adelaide, SA, Australia, 2020, pp. 35-45, doi: 10.1109/ICSME46990.2020.00014.
11. Singh, Dr. Tejinder. (2015). JAVA WEB DESIGN FRAMEWORKS: REVIEW OF JAVA FRAMEWORKS FOR WEB APPLICATIONS. IJARSE. 592-595.
12. Gerber, Adam & Craig, Clifton. (2015). Gradle. 10.1007/978-1-4302-6602-0\_13.
13. Cardoso, M., De Castro, A. V., & Rocha, Á. (2018, June). Integration of virtual programming lab in a process of teaching programming EduScrum based. In 2018 13th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

14. Ciccarelli, E. (1978). An introduction to the emacs editor.
15. Levy, J. (1996, July). A Tk Netscape Plugin. In Tcl/Tk Workshop.
16. Hausladen, J., Pohn, B., & Horauer, M. (2014, September). A cloud-based integrated development environment for embedded systems. In 2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA) (pp. 1-5). IEEE.
17. JetBrains. (2021). IntelliJ IDEA. Retrieved from <https://www.jetbrains.com/idea/>
18. Jackson, W., & Jackson, W. (2017). An Introduction to the Android Studio Integrated Development Environment. Android Apps for Absolute Beginners: Covering Android 7, 33-57.
19. Microsoft. (2021). Visual Studio Code. Retrieved from <https://code.visualstudio.com/>
20. Prakash, M. G., Ponmalar, A., Deeba, S., Akilandeswari, A., Rasool, S. B. M., & Lavanya, P. (2022, December). VSCODE-Code With Voice Using Natural Language Processing (NLP). In 2022 International Conference on Computer, Power and Communications (ICCCP) (pp. 571-574). IEEE.
21. Sindre, Guttorm & Opdahl, Andreas. (2005). Eliciting security requirements with misuse cases. Requirements Engineering. 10. 34-44. 10.1007/s00766-004-0194-4.
22. Shostack, A. (2014). Threat modeling: Designing for security. John Wiley & Sons.
23. Venson, E., Guo, X., Yan, Z., & Boehm, B. (2019, August). Costing secure software development: A systematic mapping study. In Proceedings of the 14th International Conference on Availability, Reliability and Security (pp. 1-11).
24. Gressin, S. (2017). The equifax data breach: What to do. Federal Trade Commission, 8.