**Лабораторная работа № 1**

**ФИО**

Свистунов А.И. (группа 6111)

**Topic**

Malware; Obfuscation; Binary Codes

**Описание предметной области**

Данный топик охватывает исследования в области вредоносного программного обеспечения (ПО), в частности на основе бинарного представления и применения метода запутывания кода для затруднения его анализа. Вредоносное ПО (malware) - это приложения или код, которые препятствуют нормальному использованию конечных устройств. Запутывание кода (obfuscation) - это процесс, в результате которого исходный код или исполняемый код программы приводится к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции. Бинарное представление данных (binary codes) - это способ представления информации в виде кода, в котором каждый разряд принимает одно из двух возможных значений, обычно обозначаемых цифрами 0 и 1. В контексте вредоносного ПО, бинарное представление данных часто используется для создания и распространения вредоносных программ.

**Недостаток**

Существующие методы обработки бинарных кодов, методы обфускации кода и методы обнаружения вредоносного ПО имеют свои ограничения. Предлагаются определенные методы, которые, хоть и улучшают безопасность и эффективность программного кода, но все равно имеют свои недостатки. Существующие методы профилирования последовательностей кода и лексической обфускации не исключают возможность обратной инженерии, метод маскировки данных для обеспечения безопасности программы снижает производительность из-за дополнительных операций с данными.

**Идея**

Необходимо разработать более совершенные и эффективные методы. Подход, объединяющий преимущества различных методов, должен минимизировать их недостатки, обеспечивая высокий уровень безопасности и эффективности программного кода. Современные методы должны исключать возможность обратной инженерии и риски снижения производительности программы из-за дополнительных операций с данными.

**Краткий текст обзора**

Двоичные коды играют важную роль в информационных технологиях и компьютерных науках, представляя данные компактно и эффективно с использованием всего двух возможных значений для каждого разряда. Изучение и развитие двоичных кодов имеют важное значение для современной информационной технологии [1], [2], [3]. Pengzhi предложил метод профилирования последовательностей кода [1], который упрощает анализ. Mahinthan предложил метод обнаружения уязвимых путей и ошибок производительности в бинарном коде [2]. К сожалению, существующие методы имеют сложности при обработке бинарных кодов [3], [4]. Мы предлагаем использовать техники машинного обучения для анализа бинарного кода, чтобы устранить недостатки [5]. Алгоритм будет извлекать уникальные признаки из анализа бинарного кода на этапе обучения, что позволит обнаруживать уязвимости и находить ошибки производительности.

Обфускация кода в программировании повышает безопасность приложений, делая его сложным для анализа и модификации [6], [7], [8]. Исследования в этой области важны для обеспечения безопасности программного обеспечения и защиты информации. Banescu предложил метод лексической обфускации, который улучшает безопасность программы [7], но не исключает возможность обратной инженерии. Bhaskari предложил метод затруднения чтения данных для увеличения сложности анализа программы [8], но возможно снижающий производительность из-за дополнительных операций с данными [5]. В статье авторы предложили различные методы маскировки кода, которые улучшают безопасность программы, но могут повлиять на производительность [5], [7], [8]. Мы предлагаем эффективный алгоритм, который основывается на улучшении методов запутывания кода и сохраняет структуру программы, делая ее сложной для обратной инженерии, тем самым помогая упростить процесс защиты программного кода.

Исследования в области вредоносного ПО важны для обеспечения безопасности информационных систем, так как угрозы постоянно возрастают [9], [10], [11]. Защита от таких угроз основана на использовании методов запутывания кода [6], [7], чтобы усложнить его анализ и обнаружение. Mell предложил метод использования статического анализа для обнаружения вредоносных программ без их запуска, что позволяет проводить анализ без риска [11]. Однако, он не всегда способен полностью понять функциональность вредоносного ПО. Fernandez предложил метод предоставления информации о структуре и функциональности программы [12], но этот метод также имеет ограничения которые подробно описал Sanders [4]. Мы предлагаем комбинированный подход статического анализа, который может более точно определить потенциальные угрозы и действия вредоносного ПО, используя различные техники анализа [9], [10]. Этот подход поможет улучшить обнаружение угроз и защитить информационные системы.

**Ресурсы**

[1] P. Xu, Z. Mai, Y. Lin, Z. Guo, и V. S. Sheng, «A Survey on Binary Code Vulnerability Mining Technology», *J. Inf. Hiding Priv. Prot.*, т. 3, вып. 4, сс. 165–179, 2021, doi: 10.32604/jihpp.2021.027280.

[2] M. Chandramohan, «Scalable analysis for malware and vulnerability detection in binaries», Nanyang Technological University, 2018. doi: 10.32657/10220/46626.

[3] V. Vouvoutsis, F. Casino, и C. Patsakis, «On the Effectiveness of Binary Emulation in Malware Classification». arXiv, 2022 г. Просмотрено: 5 март 2024 г. [Онлайн]. Доступно на: http://arxiv.org/abs/2204.04084

[4] J. Saxe и H. Sanders, *Malware data science: attack detection and attribution*. San Francisco, CA: No Starch Press, 2018.

[5] H. Xue, S. Sun, G. Venkataramani, и T. Lan, «Machine Learning-Based Analysis of Program Binaries: A Comprehensive Study», *IEEE Access*, т. 7, сс. 65889–65912, 2019, doi: 10.1109/ACCESS.2019.2917668.

[6] K. A. Roundy и B. P. Miller, «Binary-code obfuscations in prevalent packer tools», *ACM Comput. Surv.*, т. 46, вып. 1, сс. 1–32, 2013, doi: 10.1145/2522968.2522972.

[7] S. Banescu, C. Collberg, V. Ganesh, Z. Newsham, и A. Pretschner, «Code obfuscation against symbolic execution attacks», в *Proceedings of the 32nd Annual Conference on Computer Security Applications*, Los Angeles California USA: ACM, 2016, сс. 189–200. doi: 10.1145/2991079.2991114.

[8] C. K. Behera и D. L. Bhaskari, «Different Obfuscation Techniques for Code Protection», *Procedia Comput. Sci.*, т. 70, сс. 757–763, 2015, doi: 10.1016/j.procs.2015.10.114.

[9] M. Egele, T. Scholte, E. Kirda, и C. Kruegel, «A survey on automated dynamic malware-analysis techniques and tools», *ACM Comput. Surv.*, т. 44, вып. 2, сс. 1–42, 2012, doi: 10.1145/2089125.2089126.

[10] B. Kolosnjaji *и др.*, «Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables», в *2018 26th European Signal Processing Conference (EUSIPCO)*, Rome: IEEE, 2018, сс. 533–537. doi: 10.23919/EUSIPCO.2018.8553214.

[11] P. Mell, K. A. Kent, и J. Nusbaum, «Guide to malware incident prevention and handling», National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-83, 2005. doi: 10.6028/NIST.SP.800-83.

[12] D. Fraunholz, C. Lipps, M. Zimmermann, S. Duque Antón, J. K. M. Mueller, и H. D. Schotten, «Deception in Information Security: Legal Considerations in the Context of German and European Law», в *Foundations and Practice of Security*, т. 10723, A. Imine, J. M. Fernandez, J.-Y. Marion, L. Logrippo, и J. Garcia-Alfaro, Ред., в Lecture Notes in Computer Science, vol. 10723. , Cham: Springer International Publishing, 2018, сс. 259–274. doi: 10.1007/978-3-319-75650-9\_17.