УДК: 519.85 **Гаврилів В.В.**

МЕТОДИ АНАЛІЗУ ВРАЗЛИВОСТЕЙ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

У статті обґрунтовано основні методи аналізу можливих шляхів атаки на апаратне забезпечення через інтегровані програмні системи, та розглядаються механізми організації захисту, використовуючи технології перевірки. моделей.

Ключові слова: рівні проектування програмного забезпечення, рівні проектування апаратного забезпечення, інсерційне моделювання, агенти та середовища.

The article substantiates the main methods of analyzing possible ways of attacking hardware through integrated software systems, and considers the mechanisms of protection organization using model checking technologies.

Key words: software development levels, hardware development levels, insertion modeling, agents and environments.

Атаки на апаратне забезпечення, такі як атаки на бічні канали живлення, ставлять під загрозу безпеку вбудованих систем за низьку вартість. Захист вбудованих систем від таких атак передбачає захист апаратних блоків, програмного забезпечення та інтеграцію апаратних та програмних комплексів.

Оскільки розробка апаратного та програмного забезпечення зазвичай є незалежними процесами, вони потребують додаткових механізмів та апаратних рішень для забезпечення достатнього рівня безпеки саме на рівні програмно-апаратної взаємодії. Процеси розробки апаратного та програмного забезпечення мають свої власні складності та особливості. Розробник апаратного забезпечення або розробник програмного забезпечення реалізує апаратне чи програмне забезпечення мовами високого рівня, які пізніше оптимізується до кінцевого продукту за допомогою автоматизованих засобів.

Ці автоматизовані інструменти в основному спрямовані на оптимізацію архітектури як апаратної частини так і програмного застосунка для покращення продуктивності і може виникнути ситуація за якої важливий вставлений контрзахід безпеки буде проігноровано. Крім того, важливо оцінити вразливість системи перед її розгортанням. Методи оцінки кінцевого застосунка під час проектування дозволяють визначити, чи є проект вразливим до атаки побічного каналу, а також завдяки спільно розробленим програмно-апаратним механізмам захисту дозволяють переконатися, що впроваджені засоби захисту залишаються на кінцевому продукті.

Додатково розглядаються засоби перевірки валідності оцінки часу проектування порівняно з фізичними вимірюваннями шляхом розробки та виготовлення спеціального чіпа.

У якості головного інструмента для аналізу можливих шляхів проведення атаки на апаратно-програмну архітектуру в дипломному дослідженні використовується система інсерційного моделювання, яке спрямовано на побудову моделей та вивчення взаємодії агентів та середовищ у складних розподілених багатоагентних системах.

У якості прикладу програмно-апаратної атаки розглядається механізм проведення атаки на програмне забезпечення, записане на чіп методом переписування двійкового коду з використанням застосунків-дизассемблерів.

Побудова інсерційної моделі для зазначеного типу атаки базується на сприятливій властивості відповідного рівня ієрархії - а саме на тому, що точно відомо, яка частина програмного коду відповідає якій частині машинного коду. Ця властивість застосовується для побудови ітераційного процесу, який, використовуючи моделювання ефектів несправності може локально застосовувати контрзаходи лише до тих частин, яким вони потрібні.

В роботі на основі цих процесів описується побудова відповідної інсерційної моделі, яка дозволяє проведення комплексного аналізу можливих вразливостей, з урахуванням як особливостей забезпечення захисту з апаратної сторони програмно-апаратного комплексу так і з програмної. Модель системи може включати також і модель зовнішнього середовища для цієї системи. У загальному вигляді середовище представляє собою усі можливі стани програмно-апаратного комплексу, а зловмисник у якості агента використовує відповідну функцію занурення, яка може відображати можливі варіанти атаки на середовище.

ЛІТЕРАТУРА:

1. C.A.R. Hoare. Communicating Sequential Processes. Prentice Hall, New York, 1985.
2. L. Aceto, Wan Fokking, and C. Verhoef. Structural operational semantics. In J. A. Bergstra, A. Ponce, and S. A. Smolka, editors, Hand book of Process Algebra, pages 197– 292. North-Holland, 2001.
3. Letichevskyand D. Gilbert. A Model for Interaction of Agentsand Environments. In D. Bert, C. Choppy, P. Moses, editors. Recent Trendsin Algebraic Development Techniques. Lecture Notesin Computer Science 1827, Springer, 1999.
4. Abubakr Abdulgadir, William Diehl, andJens-PeterKaps. Anopen-sourceplatformforevaluationofhardwareimplementationsoflightweightauthenticatedciphers. In David Andrews, Ren´e Cumplido, Claudia Feregrino, and Marco Platzner, editors, 2019 International Conferenceon Re Con Figurable Computing and FPGAs, Re Con Fig 2019, Cancun, Mexico, December 9-11, 2019, pages 1–5. IEEE, 2019.
5. Kapil Anand, Matthew Smithson, Aparna Kotha, Khaled Elwazeer, and Rajeev Barua. Decompilation to compiler highirin a binaryrewriter. University of Maryland, Tech. Rep, 2010.

Науковий керівник доктор фізико-математичних наук, професор Песчаненко В.С.