instability.

References

1. Ілляшенко С. М. Економічна безпека підприємства. — Суми: Університетська книга, 2016;
2. Економічна безпека : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Н. П. Карачина, А. В. Вітюк. – Вінниця : ВНТУ, 2023. – 112 с;
3. Козаченко Г. В., Пономарьов В. П., Ляшенко О. М. Економічна безпека підприємства: сутність та механізм забезпечення. — Київ: Лібра, 2003;
4. Бланк І. О. Управління фінансовою безпекою підприємства. — Київ: Ніка-Центр, 2013 ;
5. Діденко Є. О. Економічна безпека підприємства: теорія і практика. — Київ: Центр навчальної літератури, 2015.
6. Литвиненко, І. М. Системи внутрішнього контролю та захист активів підприємства. Київ, 2021.
7. Сидоренко, Л. І. Корпоративна безпека та ризик-менеджмент. Київ, 2018.
8. Кравченко, Н. І. Ризик-менеджмент та корпоративна безпека: сучасний підхід. Харків, 2022.
9. Іваненко, О. В., Петренко, С. М. Економічна безпека підприємств: збірник наукових праць. Київ, 2022.
10. Романенко, Ю. В. Аудит та оцінка економічної безпеки підприємства. Київ, 2019.
11. Ковальчук, В. П. Сучасні методи забезпечення економічної безпеки підприємств. Харків, 2021.
12. Шевченко, В. П. Інформаційна безпека та захист даних підприємств. Львів, 2020.

LINUX DRIVER SECURITY ISSUES AND COMMON DEVELOPER MISTAKES

Pavliuk I., student,

Novik I.O., candidate of economic sciences, associate professor,

National technical university "Kharkiv polytechnic institute"

Linux driver security it's not the most discussed problem in systems programming, but it exists. Since drivers operate in kernel space and have direct access to hardware resources, any failures in their implementation can have serious consequences, ranging from incorrect device operations to critical security vulnerabilities. This isn't to say that the problem applies only to Linux, but to all operating systems in general. However, we'll be discussing Linux specifically because,

for example, a significant number of standalone systems, such as servers, operate on Linux-like systems, and errors in the drivers on these systems can lead to the entire server system crashing.

This is my opinion, not only because of the technical difficulty of developing drivers but also because most of them have been developed for specific devices, sometimes under great time pressure and with inadequate documentation. It is this that makes their development process more prone to human error and incomplete testing.

Driver Security Issues

The main difficulty is that drivers operate at the most privileged level of the system, lacking the protection mechanisms common to user applications. Any developer error can cause a kernel crash or open the door to exploitation.

The most common issues pertain to memory management, input processing, and correct behavior under multithreading conditions. Even slight logical mistakes or omission of checks may result in vulnerabilities that could be exploited by an attacker to acquire privileges. Yet another complication is introduced by the fact that some drivers are supported by a small number of developers, so individual bugs can go unnoticed for a very long time.

Common Developer Mistakes

1. Failure to check pointers between arrays.

One of the most common problems is a lack of trust in our input data. Vendors sometimes assume that data loggers are correct and the buffer size is always valid. This creates a barrier to reverse-casting, use-after-free, and other attacks.

2. Attribute races.

High data flow and interrupts complicate driver development. Bugs in variable spinlocks, mutexes, or atomic operations lead to incorrect driver behavior and degrade performance.

3. Use of deprecated or broken kernel APIs.

Some functions were implemented in Linux during the time when security was not considered that important, like unchecked kmalloc or insecure versions of copy_to_user/copy_from_user. Still out of habit or due to lack of experience, some

developers use these functions.

4. Incorrect resource allocation.

Kernel instability is also often caused by memory errors, resource leaks, and frequent processing stalls.

5. Logistical tradeoffs and revision count.

The code is technically correct, but the number of revisions introduced by marginal minds leads to logical inconsistencies.

Most problems come not only from the complexity of driver development but also from the lack of a secure coding culture in low-level programming. Furthermore, many problems have emerged at the edges, while young engineers continue to copy decade-old solutions, unaware of modern security measures.

Situation can be improved through secure programming, automated code analysis, and specialized knowledge of kernel principles. It's also important to update documentation and applications, even if they reflect the style and structure of early distributor projects. Linux driver security is no longer a pressing issue and is constantly evolving.

Developer errors are due both to the technical complexity of the order and to inattention to detail. The implementation of practical, secure coding, modern static analysis tools will reduce the number of leaks and improve the stability of Linux-addition/supporting systems.

https://drive.google.com/drive/folders/1kcDRQlfZqAu68dyQTG9udj5FOyMa04NF?usp=drive_link

CHALLENGES IN THE FIELD OF LABOR REMUNERATION IN UKRAINE AND THEIR RESOLUTION

Rohach A. O., student,

supervised by Polyakova T. L., PhD, Associate Professor

National Technical University «Kharkiv Polytechnic Institute»

Wages are a crucial economic indicator that affects not only the standards of population’s living, but also the overall stability of the state. In context of the war that has been going on in Ukraine since 2022, the issue of wages has become especially