**Implementing Disk Encryption and
Integrity Checks on Linux to Prevent Hard Drive Corruption in Client System**

Damian D. Richardson

Old Dominion University Norfolk, VA

Course number: CYSE 270

Doctor Shideh Mehr

December 3rd, 2024

 **Abstract**

The proposed solution demonstrates how these combined approaches enhance data security and system reliability, particularly in environments prone to physical damage, power failures, or malicious activities. By leveraging open-source tools, this research provides a scalable and cost-effective framework for securing Linux client systems against hard drive corruption and data breaches. This study explores the implementation of disk encryption and integrity checks on Linux-based systems to prevent hard drive corruption and protect sensitive information. Disk encryption secures data at rest by encoding it, making unauthorized access virtually impossible, while integrity checks monitor and verify the consistency of stored data, detecting and mitigating corruption early.

**Securing Linux Systems with Disk Encryption and Integrity Checks**

Linux has long been a trusted operating system for its flexibility, stability, and robust open-source community. Unfortunately, Linux is not immune to threats like hard drive corruption and unauthorized data access. This paper takes a practical and approachable look at leveraging disk encryption and integrity checks to safeguard Linux client systems, ensuring data remains secure and reliable. We will also measure the success of this research by looking at CPU usage and file access latency.

***Background***

. Disk encryption tools like LUKS (Linux Unified Key Setup) and dm-crypt can protect data at rest by encrypting it so only authorized users can access it. Meanwhile, integrity checks using tools like fsck (File System Consistency Check) and IMA (Integrity Measurement Architecture) ensure data stays consistent and secure. Linux systems already offer these tools, but the question is: how can we combine them in a way that's practical, efficient, and accessible to everyday users and administrators?

***Objectives***

This research aims to enhance data security by implementing disk encryption, ensuring sensitive data remains confidential even in cases of theft or unauthorized access, evaluating system performance to balance security measures with efficiency, focusing on minimal impact on CPU usage and file access latency, and develop a scalable and cost-effective framework using open-source tools, making it accessible for personal devices and enterprise-level systems alike.

***Methods***

We Achieved these goals using Ubuntu on Oracle's VMWare to bring them to life. To create a secure and reliable Linux-based system, we implemented an approach that combined encryption, integrity checks, and system hardening techniques. For disk encryption, we used LUKS/dm-crypt with AES-256. We also configured the drives to lock automatically during idle periods or when the system shuts down, reducing potential exposure.

For integrity checks, we focused on the filesystem and real-time validation. Journaling filesystems like ext4 were selected for their ability to recover quickly from power outages and unexpected shutdowns. We automated regular fsck scans to detect and fix any filesystem inconsistencies, ensuring data integrity even under stress. Integrity Measurement Architecture (IMA) was used to validate critical system files in real-time, with secure hash storage using the system's Trusted Platform Module (TPM). This approach allowed for confident detect tampering or corruption systems before it could compromise the hard-drives.

System-level hardening played a crucial role in the overall security framework. We partitioned the disks, separated system files, user data, and temporary files to minimize cross-contamination risks in case of corruption. For power loss scenarios, write-back caching was disabled, prioritizing data integrity over speed. Permissions were strictly managed using chmod and chown. Finally, we automated encrypted backups using tools like duplicity. Simultaneously, integrity checks were implemented using fsck for filesystem consistency and IMA for real-time file validation.

Common data corruption scenarios were simulated to test how well these tools worked together. The results were analyzed to measure system performance, focusing on metrics like CPU usage, file access latency, and recovery success rates.

***Results***

Disk encryption kept sensitive data secure, while integrity checks caught and resolved corruption before it could cause lasting damage. CPU usage saw an increase of 7% (33% usage to 35.35%), and file access latency rose by just 4% (20% to 20.8%) which is manageable We can create a secure and reliable Linux environment that addresses real-world risks by integrating disk encryption and integrity checks.

**Conclusion**

By integrating disk encryption and integrity checks, we can create a secure and reliable Linux environment that addresses real-world risks. The combined approach not only protects against data corruption and unauthorized access but does so in a way that’s efficient, cost-effective, and easy to implement.