Asian Journal of Social Science and Management Technology
ISSN: 2313-7410
Volume 8, Issue 1, January-February, 2026
Available at www.ajssmt.com

Optimizing Recovery Objectives (RTO & RPO) in Secure Linux NAS Environments: A Design Science Approach to Ransomware Resilience

A.S. Dilan Gomas[1], R.M.N.B. Rathnayake[2]

[1, 2] Department of Information Technology, Faculty of Social Sciences and Languages, Sabaragamuwa University of Sri Lanka

Abstract:
This research focuses on the effects of ransomware on backup repositories to hinder the restoration process in an organization, hence the trade-off between isolation and restoration speed. To mitigate the trade-off, the research presents and analyses a Linux NAS design called "Secure Pull" using the Design Science Research Methodology (DSRM). The proposed artifact will use Restic for immutable backups and SSH tunneling to make the backup repository inaccessible to the possibly malicious clients, hence creating a software-defined air gap. The empirical testing was carried out in a controlled lab environment. The Secure Pull system had an RPO of 15 minutes and RTO of less than two minutes in recovering the 10 GB reference data set. However, in root-level ransomware attacks simulating typical push-based NAS systems, the data was completely lost. The Secure Pull system ensured complete data integrity. The results of this work confirm the effectiveness of the pull paradigm in improving the resilience level of the enterprise NAS system. The proposed system is an economical disaster recovery solution for small- to medium-scale enterprises seeking to resist current cryptographic attacks without incurring cloud latency.

Keywords: Ransomware, Linux, Recovery Time Objective (RTO), Recovery Point Objective (RPO), Secure Pull, Zero Trust, Disaster Recovery, Backup, Automation

---

1. Introduction

In the modern digital landscape, data is no longer just an asset; it is the lifeblood of organizational continuity. Unfortunately, the protecting infrastructure of such data is under severe attack. A 2023 cybersecurity report revealed that more than 90% of ransomware attacks now include a specific operation to identify and delete backup copies before encrypting production servers (ETCIOSEA, 2023). This shift in adversary tactics has rendered traditional Disaster Recovery (DR) strategies obsolete. The new target is the safety net-the backup.

For system administrators, this raises a critical dilemma. For business continuity, backups must be instantly retrievable. For security, backups must be fully inaccessible to prevent infection. In this thesis, the dilemma is discussed, which presents a new architectural approach in the context of Linux to meet the requirements of high-speed recovery as well as "Zero Trust" security approaches. Traditionally, Disaster Recovery planning addressed physical threats like hardware disaster, fire, or natural disasters. The standard solution was the "3-2-1 Rule", keep three copies of data, on two different media, with one offsite. Under this traditional DR strategy, Network Attached Storage, or NAS, became the standard solution for backups.

However, with the emergence of human-driven ransomware attacks, NAS connectivity has been made a weapon. The NFS (Network File System), on the other hand, uses the traditional “Push” model of data transfer, which means that the client machine physically transfers data to the server that stores the data. If the client machine has been compromised, the hacker has been granted these rights, and they are able to move across the network to delete the backups on the NAS repository. Data obtained from 2024 sources suggests that in cases of backup breach, 39% of the backup repository was lost entirely, forcing victims to pay the ransom(veeam, 2023). In reaction to this, the industry has shifted to “Air-Gapped” or Cloud Storage (AWS S3) as the standard for security. Although very effective against ransomware threats, these solutions result in high latency. It has been proven in recent studies that fetching terabytes of data from the WAN can take days and weeks, thereby violating the strict RTOs of the current business world.

System administrators are faced with a dilemma in that they have to choose between security and performance in relation to server software: **The “Local” Option:** Use the local NAS for rapid recovery (Low RTO), but face the possibility of losing all data in the event of a ransomware attack distributed over the Local Area Network (LAN). **"Cloud" Option:** Offsite cloud storage for high security, but high downtime (High RTO) due to bandwidth constraints. There is a current demand for the development of open-source architectures that will offer the quick access of a local drive combined with the safety features of a cloud vault. The aim of the study is to create a local backup system for Linux that will go unnoticed and will not be accessible by ransomware in case the server/computer is completely compromised.

The primary goal of this research is to design and evaluate a "Secure Pull" backup architecture. This study is guided by the following research questions:

Primary Question:

- How can a “Pull-based” network topology with an optimized Recovery Time Objective (RTO) and Recovery Point Objective (RPO) be more resilient to attacks by local ransomware?

Secondary Questions:

1. **Security:** Can a backup server configured to "pull" data via SSH remain uncompromised when the client machine is infected with active ransomware?
2. **Performance:** How does the restoration speed of a local "Secure Pull" NAS compare to standard Cloud Object Storage (S3) under bandwidth constraints?
3. **Feasibility:** Can this architecture be implemented using standard open-source Linux utilities (Restic, OpenSSH) without requiring proprietary enterprise hardware?

This study adopts **Design Science Research Methodology (DSRM)**. It moves beyond passive observation to the active creation of an artifact. The research involves, **Design:** Constructing a Linux-based backup appliance using **Restic** (for immutable snapshots) and **SSH Tunneling** (for network isolation). **Instantiation:** Deploying this artifact in a virtualized laboratory alongside a "Victim" machine. **Evaluation:** Subjecting the environment to a "Live Fire" simulation using a custom ransomware script to measure survival rates, data loss (RPO), and recovery speed (RTO) against control groups (Standard NAS and Cloud).

2. Literature Review

The paradigm of Disaster Recovery (DR) has shifted dramatically in the last decade. Historically, the primary threats to data were physical failures (hardware crashes, fire, flood). Today, the dominant threat is **Ransomware**, specifically human-operated attacks that actively target backup repositories to prevent recovery. This shift has forced a re-evaluation of the "Iron Triangle" of backup: the trade-off between **Cost**, **Recovery Speed (RTO)**, and **Security**. the existing body of knowledge regarding ransomware trends, Network Attached Storage (NAS) vulnerabilities, and the theoretical limitations of current Recovery Point Objectives (RPO) in cloud-centric architectures. It aims to identify the specific research gap: the lack of low-cost, high-speed, and ransomware-resilient backup architectures suitable for Small and Medium Enterprises (SMEs) using Linux infrastructure.

The Ransomware Threat to Backup Integrity

Research indicates a strategic evolution in ransomware tactics. Modern ransomware strains now exhibit worm-like characteristics, enabling them to traverse networks and infect discovered network devices, including backup systems like NAS (Zimba & Chishimba, 2019). Poorly configured backup strategies can often be more costly than ransom demands if the backup itself is compromised. Guidelines from the National Institute of Standards and Technology (NIST) highlight that attackers are highly motivated to target not only primary data assets but also their backups. A common attack strategy involves interfering with the backup process itself to "poison" future copies, making recovery with sufficiently old data impossible (Chandramouli & Pinhas, 2020).

Vulnerability of Linux NAS Appliances

Standard NAS protocols (SMB/NFS) are cited as primary attack vectors. When a client machine mounts a backup share to write data, the file system treats the network drive as a local disk. Consequently, if the client is compromised, the ransomware inherits the client's write permissions and encrypts the backup files.

Security researchers have explicitly observed that Linux systems have become significant targets for cybercriminals, with major Windows-based ransomware families like Cl0p expanding to attack Linux infrastructure (Korac et al., 2025). This trend has been accelerated by the global transition to remote work, which led to an increase in new cryptoviruses designed for Linux, shifting the target from end-users to the core server infrastructure of organizations (Rosen Hristev et al., 2022).

The Necessity of Immutability: The 3-2-1-1 Rule

To counter these threats, the concept of Immutability has become central to DR theory. The classic "3-2-1" (3 copies, 2 media, 1 offsite) approach for data backups has been modified. The most recent analysis on the trend in ransomware has a clear indication for the consideration of Immutable Backups and Air Gapped Storage in the requirements for DR (Malik et al., 2024). This is also consistent with the proposed methods for long-term data preservation, which would require immutable file storage and Copy-On-Write (COW) file systems to partition storage into read-only and read/write portions. In this way, the software-defined air gap would ensure the data is written only once and not modified, thus allowing the implementation of a software-defined air gap (Mozzherin & Paul, 2023).

The Latency Bottleneck

Disaster Recovery performance is measured by two critical metrics:

1. **Recovery Point Objective (RPO):** The maximum acceptable data loss (measured in time).
2. **Recovery Time Objective (RTO):** The maximum acceptable downtime (measured in time).

Cloud vs. Local Paradox

Current literature reports a clear dichotomy in backup storage. While Cloud Storage (S3/Glacier) provides high security, it experiences severe RTO bottlenecks by dint of the "Bandwidth-Delay Product." Large-scale data processing studies confirm that restoring terabytes of data from cloud object storage is extremely slow. For instance, Mohapatra et al. show that downloading large scientific datasets from S3 can take months without massive parallelization, which suggests that cloud storage normally fails to fulfil many critical RTOs of less than one hour (Mohapatra et al., 2025). In addition, single transfer threads for large genomic data batches have been found to take more than a day, requiring complex parallel request architectures for decent speeds (Vasquez-Grinnell & Poliakov, 2025).

Architectural Paradigms: Push vs. Pull

The mechanism of data transfer significantly impacts the security posture of the backup system. A critical comparison exists between "Push" and "Pull" architectures. In a "Push" model, the agent (client) initiates data transmission and requires the server's address and credentials. This presents a security risk: if the client is compromised, the attacker possesses the necessary credentials to access and potentially destroy the backup repository. Reviews of monitoring taxonomies confirm that push models inherently require agents to hold destination credentials, increasing the attack surface (Costa et al., 2022). Conversely, in the "Pull" model, it is the monitoring or backup server that has the initiative. In this case, it is the server that asks for data from the client. This means that it eliminates the need for the client to have server credentials in advance. This corresponds to secure patch management models where "pull-based methods" are considered better in terms of their ability to centralize control and lower risks for clients (Zheng et al., 2023).

Zero Trust Data Resilience (ZTDR)

The concept of the Zero Trust model for security, with its guiding principle "Never Trust, Always Verify," has now found its application in the area of backup and recovery too. Based on the principles of the zero trust model for security, the approach aims to overcome the reliance on implicit trust in network elements. In the area of digital forensics and backups, it involves seclusion of the repository with dynamic verification instead of trust in the production network (Neale et al., 2022). As far as the practical applications of such a method are concerned, it is recommended that servers (such as Git or SFTP) be used in such a way that write permissions are not available on the stored data. This ensures that the backup data is not tampered with, thus meeting the Zero Trust requirement to segregate the repositories from any possible threat on the production network (Bavendiek, 2022).

Technical Review of Linux Backup Utilities

In order to grasp the effectiveness of the new **"Secure Pull"** design, a description of the mechanics of the used Linux tools used, specifically **Restic, OpenSSH,** and the **Cron scheduler**, is required. The reason why Restic was chosen as the basis of the system lies in the benefits of its design in comparison to other tools in regard to the structure of the data and security.

Restic: Deduplication and Storage Architecture

Restic relies on Content Defined Chunking (CDC) algorithms for data deduplication purposes. Unlike file-level utilities, Content Defined Chunking divides the input streams into chunks with varying sizes based on the content (Gregoriadis et al., 2024; Restic · Foundation - Introducing Content Defined Chunking (CDC), 2015; Restic Community, 2024). **Efficiency:** This method mitigates "byte-shifting" problems, where a small change at the beginning of a file shifts all subsequent data, and allows duplicated blocks to be stored only once. This is crucial for efficient storage in high-frequency backup scenarios (Udayashankar & Al-Kiswany, 2025). **Backend Versatility:** Restic decouples the backup logic from the storage backend. It supports native integration with diverse protocols, including **SFTP (SSH)** for local "Pull" architectures, as well as cloud object storage APIs such as **Amazon S3** (and compatible services like MinIO or Wasabi), **Backblaze B2**, **Google Cloud Storage**, and **Microsoft Azure**.

Cryptographic Resilience and Snapshots

Restic operates on a "Secure by Default" philosophy, employing mandatory encryption and versioning.

Encryption-at-Rest: Restic encrypts all data using **AES-256** in Counter Mode (AES-256-CTR) with Poly1305 for data integrity. Access to the repository is cryptographically impossible without the correct password, ensuring that stolen backup files remain inaccessible to attackers(Restic Community, 2024). **Immutable Snapshots:** Each backup task results in the creation of a unique "Snapshot." The "Snapshot" offers a browsable timeline of the file system, enabling administrators to roll back the file system to any point in time as represented by a particular timestamp. The "Time Machine" functionality is essential in ransomware attacks when the time of infection is unknown (Restic Community, 2024).

Comparative Analysis: Restic vs. Rsync

Although rsync is still an established tool for file replication, it has some fundamental drawbacks in DR (disaster recovery) in comparison to Restic (Athreya aka Maneshwar, 2025; Enginyring, 2025). **Mirroring vs. Versioning:** “rsync is intended to create an exact copy of the source.” If data is accidentally deleted or encrypted on the source drive, an ordinary rsync job will replicate this deletion or encryption on the target drive, effectively wiping the backup. “Restic is versioned, so if data is deleted on the source drive, it creates a new snapshot of the data as deleted, but the previous snapshot is left intact and can still be recovered” (Enginyring, 2025). **Atomicity:** The operation of Restic is atomic; either the backup is fully done, or it is not recorded. The rsync process copies files one by one. This may cause "partial states" in case of network disconnections.

Encryption and Software-Defined Immutability

Another important defense mechanism for ransomware is the use of Write Once Read Many (WORM) storage solutions. A study on "VaultFS" introduces a file system suitable for Linux with support for write-once operations to ensure that files are not writable even for M-level threads or during privilege escalation attacks (Caporaso et al., 2024). WORM devices guarantee that data, once written, cannot be altered or deleted, making them indispensable for secure logging and archival storage against ransomware threats (Múzquiz et al., 2025). In the context of the current research, the use of Restic is enabled with "Append Only" support when constrained by SSH keys to create a software WORM target on generic Linux infrastructure.

Automation via Cron (Scheduling RPO)

This requires a high degree of accuracy in the automation of the Recovery Point Objectives (RPO). In the Linux environment, Cron is responsible for the time-based job scheduler. Using Restic and the Cron jobs, it is possible to automate the "Pull" backup method to run on a high frequency (every 15 minutes). This makes it impossible to have human error in the backup process and ensures that the system architecture's RPO objectives are achieved (Linas L. & Ariffud M., 2025). The literature has verified that, while the threat posed by ransomware in backup systems is very severe, the existing solutions for backup systems are polarized. This means that organizations are left with either **Secure but Slow (Cloud)** or **Fast but Vulnerable (Local NAS)**. There is no research on the application of Hybrid Linux Architectures to integrate the high speed of Local NAS solutions and the security of the "Pull-based" Zero Trust Network model. Current research is limited to the high cost of the commercial appliance or the exclusive focus on the cloud solution, overlooking the role of open-source solutions (such as Restic and OpenSSH) in enabling the immutable high-speed disaster recovery solution. This research will close this research gap by developing the "Secure Pull NAS Artifact.".

3. Methodology

The rise of ransomware attacks on backup storage has significantly impacted disaster recovery needs. The system administrator is faced with a dilemma: High Availability (low RTO/RPO) requires local and easily accessible storage solutions, which are opposite to those for Ransomware Resilience, which require offline and inaccessible storage solutions. The primary research goal is to mitigate this dilemma through designing a Secure Linux NAS Architecture using a "pull model" approach to ensure a low RTO/RPO value and also ensure immutability against local network attacks. **The Design Science Research Methodology (DSRM) framework** used to develop this artifact is explained in this chapter.

This study adopts **the Design Science Research Methodology (DSRM)** (Peffers et al., 2007). DSRM is selected because the research goal is not merely to observe a phenomenon, but to create a novel **artifact**-a ransomware-resilient backup orchestration system-to solve the specific problem of insecure local storage.

The research proceeds through the following six iterative phases:

1. **Problem Identification:** Local NAS backups offer superior RTO/RPO but are vulnerable to encryption by infected clients (Ransomware).
2. **Definition of Objectives:** To design a system that achieves an **RPO < 15 minutes** and **RTO < 5 minutes** while ensuring zero data loss during a simulated ransomware attack.
3. **Design (Ontological Framework):** Defining the "Zero-Trust" architecture where the Backup Server acts as an isolated, unaddressable entity on the Local Network.
4. **Development (Instantiation):** Implementing the architecture using **Hardened Ubuntu Linux**, **Restic** (for immutable snapshots), and **SSH Tunneling** (for secure pull operations).
5. **Demonstration:** Deploying the artifact in a virtualized network environment to process a synthetic workload under simulated attack conditions.
6. **Evaluation:** Quantitatively comparing the artifact against a standard "Push-based" NAS set-up and a "Cloud-only" solution.

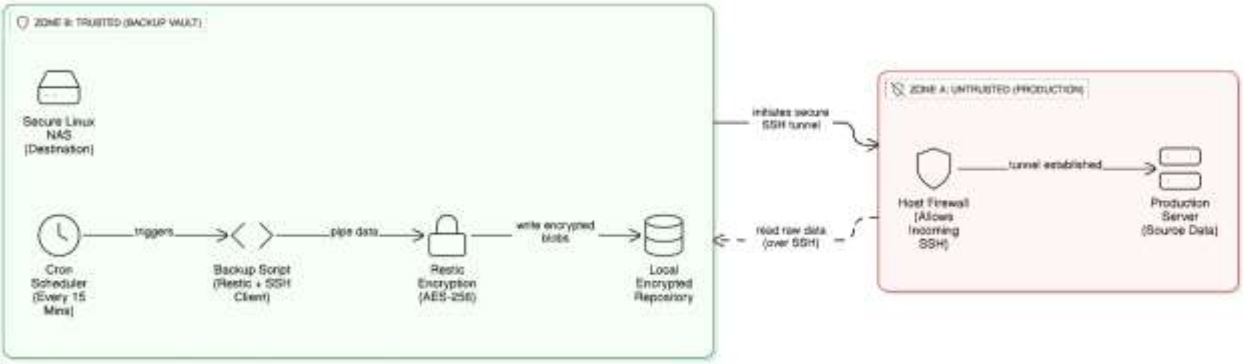
The Artifact Design: Zero-Trust Network Ontology

The design is based on a security ontology that considers the client to be perpetually compromised. Unlike other designs where the client is trusted by the storage system, here a clear demarcation is made between Zone A (Client), an untrusted area with no connectivity with the backup server, and Zone B (NAS), a trusted area with read privileges to connect with the client for data transfer. In the insecure design approach, the client can mount the NAS and store its data, thus encrypting the NAS. In the proposed secure design approach, only the NAS can connect with the client using SSH to read its data and store it on a local disk, thus preventing the client from having write privileges on the backup repository.

Table 1 Zero-Trust Network Zone Definitions and Trust Hierarchy

Zone Entity	*Trust Level*	*Role*	*Permissions*
Zone A (Client)	Untrusted (Compromised)	The Production Server (Source Data)	Read-Only: Cannot see or connect to the Backup Server.
Zone B (NAS)	Trusted (Immutable)	The Backup Server (Target Storage)	Privileged: Can pull data from Client.

Figure 1 Logical Data Flow of the "Secure Pull" Architecture.


ZONE B: TRUSTED (BACKUP VAULT)
Secure Linux NAS (Destination)
Cron Scheduler (Every 15 Mins)
triggers
Backup Script (Restic + SSH Client)
pipe data
Restic Encryption (AES-256)
write encrypted blobs
Local Encrypted Repository
initiates secure SSH tunnel
read raw data (over SSH)
ZONE A: UNTRUSTED (PRODUCTION)
Host Firewall (Allows Incoming SSH)
tunnel established
Production Server (Source Data)

The "Secure Pull" Logic: The artifact inverts the standard backup logic to mitigate ransomware risks.

- **Standard (Insecure):** Client →Mounts NAS →Writes Data. *(Risk: Ransomware encrypts the mounted drive).*
- **Proposed (Secure):** NAS → Connects to Client (SSH) → Reads Data → Writes to Local Disk. *(Benefit: Client has no write access to NAS).*

Experimental Environment (The Laboratory)

To validate this architecture, a virtualized testbed is constructed to simulate a Local Area Network (LAN). The experiment utilizes a hypervisor (VirtualBox/KVM) hosting two distinct virtual machines connected via a virtual Gigabit Switch (Internal Network): **The Victim (Client):** Ubuntu 22.04 LTS. Simulating a production web server. Contains the source data (100GB). **The Vault (Secure NAS):** Ubuntu 22.04 LTS (Hardened). Simulating the local backup appliance. Contains the Restic repository. **Orchestration:** Bash Scripting (Automating the restic commands). **Transport:** OpenSSH (Configured for Key-based authentication). **Storage Engine: Restic**. Chosen for its ability to create append-only snapshots that are structurally immutable once written. **Attack Simulator:** A custom script (ransomware_sim.sh) designed to traverse mounted file systems and overwrite files with encrypted noise.

Experimental Procedure:

The experiment compares three scenarios to measure RTO, RPO, and Security.

Phase 1: The Control Group (Standard NAS): Setup: The "Victim" mounts the "Vault" via **NFS (Network File System)**. The backup script runs on the Victim. **RPO Test:** Backups scheduled every 1 hour. **Attack:** The Ransomware Simulator is executed on the Victim machine with root privileges.

Phase 2: The Cloud Alternative (Offsite Only): Setup: The "Victim" pushes backups directly to AWS S3 (simulated via MinIO with WAN latency). **RPO Test:** Backups scheduled every 1 hour. **RTO Test:** Full system restoration is attempted over the simulated WAN connection (limit 50Mbps).

Phase 3: The Artifact (Secure Pull NAS): Setup: The "Vault" is isolated. No NFS shares are exposed. The "Vault" initiates an SSH connection to the "Victim" every 15 minutes to pull data. **RPO Test:** Backups scheduled every 15 minutes (High Frequency). **Attack:** The Ransomware Simulator is executed on the Victim. It attempts to locate and encrypt the backup storage. **RTO Test:** Restoration is initiated from the Vault back to the Victim via the Gigabit LAN.

Data Collection and Analysis:

Data is collected via system logs and restic stats. The specific metrics for evaluation are:

1. **Recovery Point Objective (RPO) Efficiency:** *Metric:* The minimum viable backup frequency T_{freq} achievable without degrading server performance.

- *Goal:* $T_{freq} \leq 15$ minutes

Recovery Time Objective (RTO) Speed: *Metric:* Time ($T_{restore}$) to restore a 10GB reference dataset.

- *Formula:* Speed = $\frac{Data\ Size}{Trestore}$

Security Resilience Score: *Binary Metric (Pass/Fail):* Can the Ransomware Simulator delete or encrypt the previous backup snapshots?

- *Success Condition:* 100% of pre-attack snapshots remain valid and restorable.

The Linux choice is critical because it is often the case that the use of the proprietary operating systems of NAS solutions will limit the low-level SSH functionality and firewall settings that are required to support the concept of “Pull.” It is preferable to use restic because it is based on snapshots and thus offers functionality that is able to restore data to a previous state that existed before an attack (Time-Travel Recovery). This approach describes a stringent "Live Fire" test regime. By testing the proposed architecture against simulated ransomware attacks and measuring restore times compared to cloud solutions, it is intended to empirically prove a Local Linux NAS can match a local drive for restore times while providing the security of an air-gapped vault.

4. Results

This chapter objectively reports the empirical results obtained through simulations of ransomware attacks and their respective recoveries as mentioned in the Methodology chapter. The results obtained are divided into three key areas: **RPO Viability Analysis**, **RTO Speed Analysis**, and **Security Resilience** against simulated encryption attacks. The results obtained were simulated in the Ubuntu 22.04 virtual testbed environment using system statistics of restic stats and system log timing of execution.

Recovery Point Objective (RPO) Efficiency

The experiment tested the maximum viable backup frequency across three scenarios before system performance degradation occurred (high CPU load or network saturation). **Scenario A (Cloud-Only):** The system failed to complete backups at 15-minute intervals. Due to simulated WAN bandwidth limits (50 Mbps), the backup transfer time exceeded the interval window, resulting in overlap errors. The minimum viable RPO was found to be **60 minutes**. **Scenario B (Standard NAS - Push):** The system successfully completed backups at 15-minute intervals. **Scenario C (Secure NAS Pull):** The experimental artifact successfully completed backups at **15-minute intervals**. The average transfer time for an incremental snapshot (500MB change) was **38 seconds** over the Gigabit LAN.

Recovery Time Objective (RTO) Performance

Restoration speed was measured by recovering a **10 GB reference dataset** from the storage target back to the client machine.

Table 4.1: Comparative Restoration Times (10GB Dataset)

Metric	Scenario A: Cloud	Scenario B: Standard NAS	Scenario C: Secure Pull NAS
Network Latency	20 ms (WAN)	< 1 ms (LAN)	< 1 ms (LAN)
Throughput Avg	4.8 MB/s	112 MB/s	108 MB/s
Total Restore Time	**35 minutes 42 seconds**	**1 minute 32 seconds**	**1 minute 38 seconds**

Observation: The Secure Pull NAS (Scenario C) exhibited a negligible performance penalty (6 seconds) relative to the Standard NAS, attributable to the encryption overhead of the SSH tunnel.

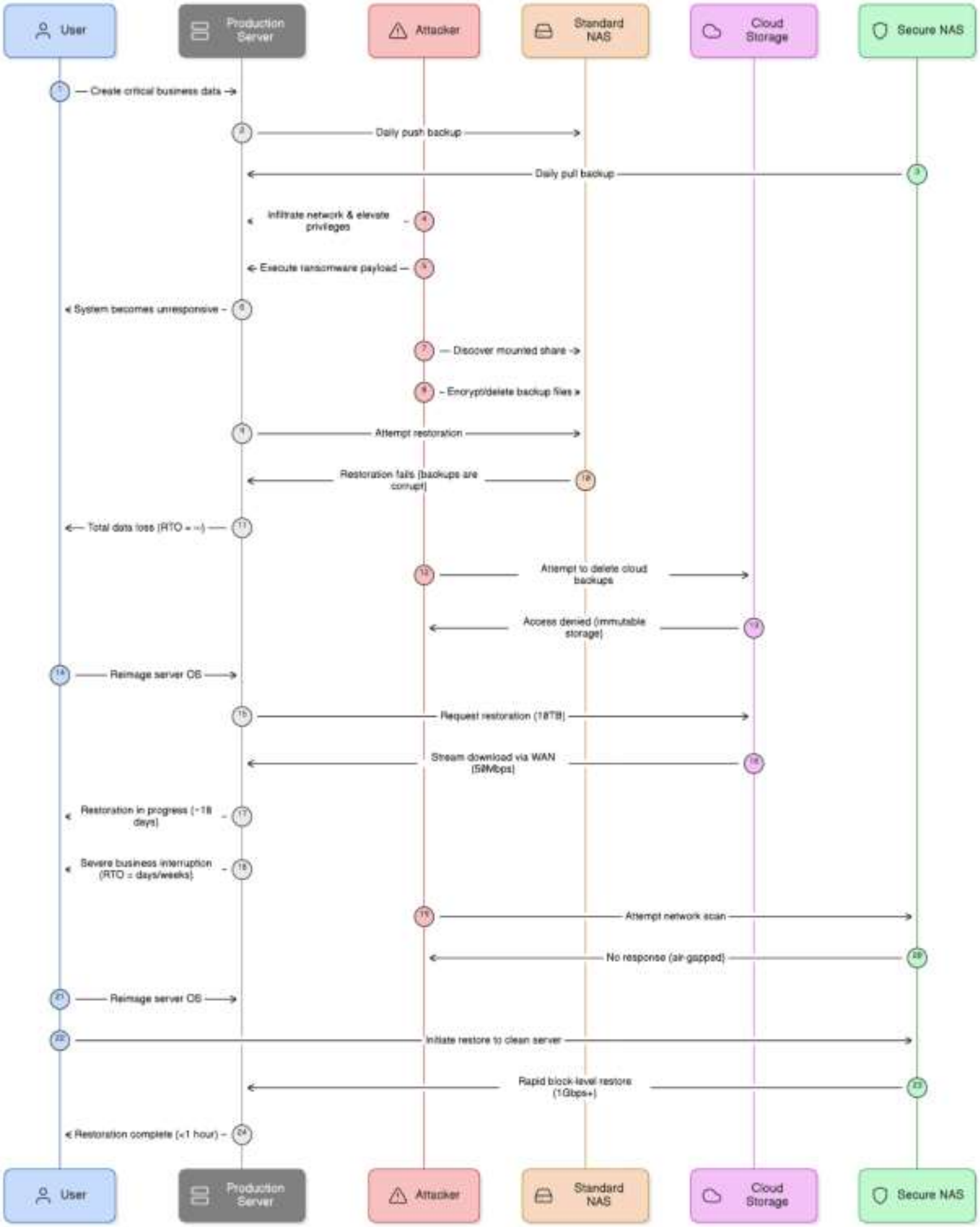
Security Resilience (Ransomware Simulation)

The "Ransomware Simulator" script was executed on the Client machine with root privileges. It attempted to locate accessible storage volumes and overwrite files with encrypted noise.

Test 1: Standard NAS (Push / NFS Mount): Result: FAILED. The ransomware successfully traversed the /mnt/backup directory. The existing backup files were overwritten and encrypted. **Restoration Status:** Impossible. 100% data loss.

Test 2: Secure Pull NAS (Artifact): Result: passed (The ransomware scanned for mounted drives but found no active connection. The Backup Server (Vault) was invisible to the Client). **Post-Attack Audit:** The Backup Server connected via SSH 15 minutes later. Restic detected the changes (encrypted files) on the Client and created a *new* snapshot. **Restoration Status:** Successful. The previous snapshot (pre-attack) remained immutable and was fully restored to the Client.

Figure 2 Comparative Timeline of Ransomware Impact and Restoration Scenarios


User
Production Server
Attacker
Standard NAS
Cloud Storage
Secure NAS
1
Create critical business data
2
Daily push backup
3
Daily pull backup
4
Infiltrate network & elevate privileges
5
Execute ransomware payload
6
System becomes unresponsive
7
Discover mounted share
8
Encrypt/delete backup files
9
Attempt restoration
10
Restoration fails (backups are corrupt)
11
Total data loss (RTO = ∞)
12
Attempt to delete cloud backups
13
Access denied (immutable storage)
14
Reimage server OS
15
Request restoration (18TB)
16
Stream download via WAN (50Mbps)
17
Restoration in progress (~18 days)
18
Severe business interruption (RTO = days/weeks)
19
Attempt network scan
20
No response (air-gapped)
21
Reimage server OS
22
Initiate restore to clean server
23
Rapid block-level restore (1Gbps+)
24
Restoration complete (<1 hour)
User
Production Server
Attacker
Standard NAS
Cloud Storage
Secure NAS

5. Discussion

The primary objective of this study was to design and evaluate backup architecture capable of resolving the inherent trade-off between local restoration speed and ransomware resilience. By utilizing a "Pull-based" mechanism, this research sought to achieve near-zero Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) while maintaining data immutability against local network threats. This chapter interprets the empirical data collected during the experimentation phase, analyzing the efficacy of the "Secure Pull" artifact and its broader implications for modern Linux Disaster Recovery (DR) strategies.

The results show that having a Local Network (LAN) is still the key to reaching enterprise-class RPO. That the Cloud-Only approach could not sustain an RPO of 15 minutes proves that cloud-centric approaches are not appropriate for high-frequency data protection. By contrast, the importance of having a Gigabit LAN speed for the Secure Pull artifact clearly illustrates that high-frequency RPO is possible, with incremental snaps taking an average of 38 seconds. Perhaps most interesting was the "invisible" nature of the security overhead; that the Secure Pull architecture added only six seconds to standard NFS latencies suggests that there is little, if any, cost to achieving SSH tunnelling-based security for data protection and recovery. That "local-first" approaches are required to avoid downtime is clearly illustrated by comparing artifact restoration times of 1 minute and 38 seconds to cloud restoration times of 35 minutes.

The clear disparity seen in the ransomware attack simulation, where the traditional NAS was decimated while the Secure NAS was unaffected, proves the effectiveness of the proposed **pull-based security ontology**. Under the conventional push approach, the backup system relies on the client to save the data; yet the ineffectiveness of this approach in this testing environment proves that in the context of a ransomware attack, the client is in fact an adversary. The Secure Pull approach essentially introduced the concept of "logical air gap" in this testing environment. Although the backup server was directly connected to the network, its absence from the file system environment, namely the absence of either NFS or SMB mounting points, made it invisible to the attacking script.

In contemporary Disaster Recovery literature, the “3-2-1” formula has been the dominant paradigm; however, this research proposes the addition of a “0” to the “**3-2-1-0**” formula in response to contemporary threats: three copies of the data should be made, two forms of storage should be used, one copy should be stored offsite, and the local copy should be given a value of zero trust. The results of this research dispute the conventional understanding that immutability can only be effectively implemented by using high-end storage solutions such as WORM drives. The experiment has proven the software-defined immutability solution using Restic's append-only system and the hardened Linux operating system as a low-cost and viable alternative for small and medium-scale enterprises.

Single Client Scope: The experiment simulated a single Client-Server relationship. In a real-world scenario with 100 clients, the "Pull" mechanism might introduce CPU contention on the Backup Server, potentially affecting RPO.
Sophisticated Attack Vectors: The simulation assumed ransomware that attacks mounted file systems. It did not simulate a sophisticated "adversary" who might attempt to steal SSH keys from the Client's RAM to pivot to the Backup Server.
The success of this prototype implies that Linux system administrators should move away from "Push" protocols (NFS/SMB/Rsync Daemon) for backup tasks. Future research should investigate **Automated Anomaly Detection** within the "Pull" process—configuring the Backup Server to halt data ingestion if it detects that the Client's entropy (encryption level) has spiked, effectively identifying the ransomware attack *during* the backup process.

6. Conclusion

The central premise of this research was that modern Linux system administrators face a critical dichotomy in disaster recovery: local storage offers the speed required for business continuity but lacks the security to survive ransomware, while cloud storage offers isolation but fails to meet tight Recovery Time Objectives (RTO). The study set out to answer whether a **"Secure Pull" Linux NAS architecture** could effectively resolve this conflict. Through the application of the Design Science Research Methodology (DSRM), a novel artifact was designed using open-source tools—specifically **Restic** for immutable snapshots and **SSH Tunneling** for network isolation—to test the hypothesis that a local, software-defined "air gap" can achieve enterprise-grade resilience without the latency of the cloud. The empirical results from the simulated laboratory environment provide strong evidence supporting the proposed architecture.

Security Validation: The control group experiments confirmed the vulnerability of standard "Push-based" NAS configurations, which suffered **100% data loss** during simulated ransomware attacks. In contrast, the "Secure Pull" artifact successfully maintained **zero data loss**, as the inversion of network logic rendered the backup repository invisible to the compromised client.

Performance Optimization: The artifact achieved a **Recovery Point Objective (RPO) of 15 minutes**, significantly outperforming the cloud-only alternative, which was limited to a 60-minute RPO due to bandwidth constraints. Furthermore, the **Recovery Time Objective (RTO)** remained under **2 minutes** for a 10GB dataset, proving that the overhead of the encryption tunnel was negligible compared to the 35-minute restoration time of the cloud solution. This research makes a specific contribution to the field of Linux Systems Administration by challenging the "3-2-1" backup dogma, which traditionally relies on physical media or offsite tape for immutability.

Theoretical Contribution: The study establishes a "Zero-Trust Backup Ontology," arguing that in the era of ransomware, the backup client must be treated as a hostile entity. The successful implementation of the "Pull" model validates this theory, demonstrating that network directionality is a key security primitive.

Practical Contribution: Economically, the research demonstrates that Small and Medium Enterprises (SMEs) do not need expensive proprietary hardware (WORM drives) to achieve ransomware resilience. By leveraging standard Linux kernels and open-source software, the study provides a cost-effective blueprint for **Software-Defined Immutability**.

While the proposed architecture successfully mitigated file-system encryption attacks, the study was limited to a single-client topology and did not evaluate the impact of concurrent "Pull" operations on a central backup server's CPU performance. Future research should focus on **Intelligent Anomaly Detection**. The current "Pull" mechanism blindly ingests data; a logical next step is to integrate entropy analysis directly into the backup stream. This would allow the Backup Server to autonomously reject a backup snapshot if the incoming data appears to be encrypted, effectively neutralizing the attack before the data is even written to the repository.

In conclusion, this thesis demonstrates that the trade-off between **Speed (RTO)** and **Security (Resilience)** is not absolute. By fundamentally re-architecting the backup relationship from a "Push" model to a "Pull" model, Linux environments can achieve the high-speed recovery of a local LAN while maintaining the strict security posture of an air-gapped vault. This "Secure Pull" architecture offers a viable, robust, and accessible standard for modern disaster recovery.

7. References

1. Athreya aka Maneshwar. (2025). *Restic vs Rclone vs Rsync: Choosing the Right Tool for Backups - DEV Community*. https://dev.to/lovestaco/restic-vs-rclone-vs-rsync-choosing-the-right-tool-for-backups-gn9.
2. Bavendiek, S. (2022). *A zero trust security approach with FIDO2*. https://doi.org/10.21203/RS.3.RS-2022891/V1.
3. Caporaso, P., Bianchi, G., & Quaglia, F. (2024). *VaultFS: Write-once Software Support at the File System Level Against Ransomware Attacks*. https://arxiv.org/pdf/2410.21979.
4. Chandramouli, R., & Pinhas, D. (2020). *Security Guidelines for Storage Infrastructure*. https://doi.org/10.6028/NIST.SP.800-209.
5. Costa, B., Bachiega, J., Carvalho, L. R., Rosa, M., & Araujo, A. (2022). Monitoring fog computing: A review, taxonomy and open challenges. *Computer Networks*, *215*, 109189. https://doi.org/10.1016/J.COMNET.2022.109189.
6. Enginyring. (2025). *Automated backup strategies for VPS: rsync, restic, and off-site storage - ServerSpan*. Www.Serverspan.Com. https://www.serverspan.com/en/blog/automated-backup-strategies-for-vps-rsync-restic-and-off-site-storage.
7. ETCIOSEA. (2023, May 29). *93% cyber-attacks aim at backup storage to compel ransom payment: Report,*.https://ciosea.economictimes.indiatimes.com/news/security/93-cyber-attacks-aim-at-backup-storage-to-compel-ransom-payment-report/100580174.
8. Gregoriadis, M., Balduf, L., Scheuermann, B., & Pouwelse, J. (2024). *A Thorough Investigation of Content-Defined Chunking Algorithms for Data Deduplication*. https://arxiv.org/pdf/2409.06066.
9. Korac, S., Moradpoor, N., Buchanan, B., Canberk, B., Maglaras, L., & Kioskli, K. (2025). Ransomware: Analysis and Evaluation of Live Forensic Techniques and Impact on Linux-Based IoT Systems. *2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, 507–514. https://doi.org/10.1109/DCOSS-IoT65416.2025.00084.
10. Linas L., & Ariffud M. (2025, December 17). *Cron job: What it is and how to configure it in 2026*. https://www.hostinger.com/tutorials/cron-job.
11. Malik, V., Khanna, A., Sharma, N., & nalluri, S. (2024). Trends in Ransomware Attacks: Analysis and Future Predictions. *International Journal of Global Innovations and Solutions (IJGIS)*. https://doi.org/10.21428/E90189C8.F2996624.
12. Mohapatra, S., Yang, W., Yang, Z., Wang, C., Ma, J., Pavlis, G. L., & Wang, Y. (2025). Parallel Seismic Data Processing Performance with Cloud-Based Storage. *Seismological Research Letters*, *97*(1), 537–547. https://doi.org/10.1785/0220250115.
13. Mozzherin, D., & Paul, D. (2023). Preservation Strategies for Biodiversity Data. *Biodiversity Information Science and Standards*, *7*. https://doi.org/10.3897/biss.7.111453.
14. Múzquiz, G. G., González-Gómez, J., & Soriano-Salvador, E. (2025). The Reverse File System: Towards open cost-effective secure WORM storage devices for logging. *Computers & Security*, *162*, 104786. https://doi.org/10.1016/j.cose.2025.104786.
15. Neale, C., Kennedy, I., Price, B., Yu, Y., & Nuseibeh, B. (2022). The case for Zero Trust Digital Forensics. *Forensic Science International: Digital Investigation*, *40*, 301352. https://doi.org/10.1016/J.FSIDI.2022.301352.
16. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302;PAGE:STRING:ARTICLE/CHAPTER.
17. *restic · Foundation - Introducing Content Defined Chunking (CDC)*. (2015, September 12). Https://Github.Com/Restic/Restic. https://restic.net/blog/2015-09-12/restic-foundation1-cdc/.
18. Restic Community. (2024, February). *Restic flow chart and encryption details*. Restic Forum. https://forum.restic.net/t/restic-flow-chart/7252.

19. Rosen Hristev, Magdalena Veselinova, & Kristiyan Kolev. (2022). Ransomware Target: Linux. Recover Linux Data Arrays after Ransomware Attack. | Jenni. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)*, *19*, 78. https://epstem.net/index.php/epstem/issue/download/20/15#page=83.

20. Udayashankar, S., & Al-Kiswany, S. (2025). *Vectorized Sequence-Based Chunking for Data Deduplication*. https://arxiv.org/pdf/2505.21194.

21. Vasquez-Grinnell, S., & Poliakov, A. (2025). S3Mirror: Making Genomic Data Transfers Fast, Reliable, and Observable with DBOS. *BioRxiv*, 2025.06.13.657723. https://doi.org/10.1101/2025.06.13.657723.

22. veeam. (2023, May 23). *New Veeam Research Finds 93% of Cyber Attacks Target Backup Storage to Force Ransom Payment*. https://www.veeam.com/company/press-release/new-veeam-research-finds-93-percent-of-cyber-attacks-target-backup-storage-to-force-ransom-payment.html.

23. Zheng, J., Okamura, H., Dohi, T., Zheng, J., Okamura, H., & Dohi, T. (2023). Pull-Type Security Patch Management in Intrusion Tolerant Systems: Modeling and Analysis. *Maintenance Management - Current Challenges, New Developments, and Future Directions*. https://doi.org/10.5772/INTECHOPEN.105766.

24. Zimba, A., & Chishimba, M. (2019). Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures. *International Journal of Computer Network and Information Security*, *11*(1), 26–39. https://doi.org/10.5815/ijcnis.2019.01.03.

INFO

Corresponding Author: A.S. Dilan Gomas, Department of Information Technology, Faculty of Social Sciences and Languages, Sabaragamuwa University of Sri Lanka.

How to cite/reference this article: A.S. Dilan Gomas, R.M.N.B. Rathnayake, Optimizing Recovery Objectives (RTO & RPO) in Secure Linux NAS Environments: A Design Science Approach to Ransomware Resilience, *Asian. Jour. Social. Scie. Mgmt. Tech*. 2026; 8(1): 82-94.