QuellSecure

WHITE PAPER · VOL. 01

THREAT RESEARCH · 2026

Ransomware detection techniques: strengths, limitations, and the path forward.

How modern encryptors evade signature, behavioral, decoy, and entropy-based defenses, and where the next generation of detection must operate.

AUTHOR

Anson Dorsey, CTO

AUDIENCE

Security architects · CISOs

READING TIME

12 minutes

00 OVERVIEW

Defenses advance. So do attackers.

Despite vast investment in cybersecurity, the number of successful ransomware attacks keeps rising, fueled by a global industry well-funded by its own success and unwilling to quietly go away.

This paper explores QuellSecure's research into ransomware encryptors found in the wild, focusing on how developers have adapted to anti-ransomware techniques. Understanding how ransomware interacts with defensive tools helps defenders optimize their organization's defensive strategy to better resist attacker evolution.

While each of the following detection techniques is broadly effective on its own, combining multiple approaches significantly reduces an individual ransomware payload's ability to reach its destination.

To maximize the benefit of a layered approach, it is essential to understand the strengths and weaknesses of different tools and how they interact to mitigate risk.

Even as defensive tools have advanced, attackers who are more clever and persistent as ever, have demonstrated a long track record of successfully evading them. The pages that follow examine each major detection technique, the evasion strategies attackers have developed, and a new approach designed to operate where evasion has nowhere left to go.

TECHNIQUES COVERED	DETECTION LAYER	READING TIME
06	File system	12 min

TOC CONTENTS

What's Inside

Six detection techniques, the evasion strategies attackers use against each, and the file-system-level approach QuellSecure has built to close the gap.

01 SIGNATURE-BASED DETECTION

Static analysis: pattern-matching the payload.

Before a ransomware encryptor executes on a target machine, it must first bypass tools that perform **static analysis**. These tools evaluate the instructions inside executables before they are allowed to run, analyzing internal patterns. If the extracted fingerprints resemble those of known threats, the executable is proactively blocked.

Malware attempts to disguise its appearance. Small changes to executables, or to the algorithms compiled within them that perform malicious encryption, can alter the fingerprint enough to bypass detection.

> **LIMIT**
>
> Zero-day ransomware represents the extreme case and, by definition, cannot be detected by comparison to known fingerprints.

02 BEHAVIORAL ANALYSIS

Watching how processes behave.

Before ransomware can write to files, it must be granted access and permission. Behavioral analysis hooks into multiple components of the operating system to identify signals that indicate malicious activity, sustained high rates of file writes, deletion of shadow copy backups, and similar patterns.

Behavioral engines aggregate many variably weighted indicators of compromise into a single decision. Increasingly, they leverage faster CPUs and NPUs by running AI models that improve accuracy while reducing noise.

Hiding in plain sight.

Behavioral engines collect signals from system drivers and processes that are generally known to attackers. Ransomware often attempts to stymie these sources before proceeding. Failure to disable a signal, however, is itself a strong indicator of compromise, making this maneuver high-risk for attackers.

Instead, ransomware frequently masks itself within the complexity of the target system. Even simple real-world systems involve hundreds of processes constantly reading and writing files, with widely varying access patterns. Behavioral detection risks becoming too noisy if it flags all potentially malicious writes, so developers optimize their activity to **resemble benign software**.

Even when flagged as suspicious, ransomware aims to be scored as minimally dangerous, buying time before triggering a threshold or human review.

> **CASE STUDY · MEMORY-MAPPED FILES**
>
> QuellSecure has studied several families whose encryptors write via memory-mapped files, scrambling bytes in special file objects that exist only in RAM, then relying on the OS to flush them to disk alongside benign writes. This breaks the direct causal link between a process and the files persisted to permanent storage.

03 DECOY FILES

Bait, calibration, and the disguise arms race.

Decoy-based detection techniques bait ransomware with a set of specific files known only to the defender, flagging a process as malicious when enough of these files are accessed or modified. Alternatively, processes or services that ransomware typically tries to stop before execution can act as bait.

While static or behavioral analysis interprets ransomware activity, this technique is largely **agnostic to attacker appearance** and operates closer to the site of malicious encryption.

STRENGTH

Operates close to the site of encryption, independent of how the payload looks.

TRADEOFF

Too many tripwires create false-positive ambiguity; too few miss the threat entirely.

A successful decoy strategy requires determining an optimal number of decoys. Too many tripwires can interfere with normal operations or create false-positive ambiguity that obscures real threats; too few may be insufficient to prevent significant file system damage.

To counter this approach, ransomware developers attempt to identify decoy files and ignore them. Defensive tools must therefore continually develop new methods to disguise bait, which is an ongoing cycle of adaptation that requires frequent updates to remain effective.

04 FILE INTEGRITY ANALYSIS

What ransomware must accomplish.

To successfully disrupt business operations for financial gain, ransomware must satisfy several objectives:

1. Damage file integrity.
2. Ensure recovery is possible only with attacker-owned keys.
3. Maximize damaged files by moving quickly across systems.

Entropy as a signal.

From the file system's perspective, ransomware does not differ greatly from benign software: both write blocks of binary data to specific locations. However, analyzing this data as a binary signal stream can be revealing. When a low-entropy file such as plain text suddenly becomes high-entropy encrypted or compressed data, ransomware encryption is likely involved. Behavioral engines often use this as one of many indicators.

EVASION · WRITE STRIPING

Modern encryptors use write striping to limit entropy increases by partially corrupting files rather than fully encrypting them, destroying integrity while writing fewer encrypted bytes. Mixed intact and corrupted writes can hide entropy changes from statistical analysis, while reducing CPU usage so more files are encrypted, faster.

05

A NEW AND DIFFERENT APPROACH

Detect at the moment of malicious encryption where evasion has nowhere left to go.

The last opportunity to prevent widespread damage is the moment of malicious encryption itself. QuellSecure has invented and patented a new method that operates at this stage, with an emphasis on simplicity of operation, configuration, and management.

WHERE EACH TECHNIQUE OPERATES

PRE-EXECUTION	Static signature analysis of the executable	○
RUNTIME	Behavioral hooks aggregate indicators of compromise	○
PROCESS	Decoy file access and entropy heuristics	○
FILE SYSTEM	Inline analysis of writes where Quell operates	●
STORAGE	Persisted, encrypted bytes that are too late to prevent	○

05 HOW QUELL'S ALGORITHM DIFFERS

Inline. Low-overhead. Evasion-resistant.

Strategically, the file system is an ideal layer for detection because, regardless of attack strategy, malicious encryption must ultimately write to files, and those writes must pass through the file system. Quell's algorithm is differentiated in several key ways.

1. **Few bytes, accurate determination**
   Requires very few bytes to make an accurate determination, making it resistant to write striping and similar evasion techniques.
2. **Inline, as the data is written**
   Analyzes data inline as it is written, keeping overhead extremely low and allowing it to keep pace with optimized ransomware.
3. **Below the noise floor**
   At this depth in the operating system, noise from user behavior and business applications is removed. New variants, encryption methods, write striping patterns, and memory-mapped files appear fundamentally similar, reducing the need for frequent updates.

Taken together, these attributes enable rapid and accurate detection of ransomware while ignoring benign processes.

06 KEY HARVESTING

Recovery, not prevention.

Key harvesting does not prevent or halt ransomware attacks, but it can assist recovery without requiring full restoration from backup or accepting data loss. These tools hook into encryption APIs and operating system cryptographic tools, recording keys in case a process is later identified as malicious. Harvested keys can then be used to decrypt affected files.

How attackers complicate it.

Ransomware attempts to complicate key harvesting by using **unique keys per file**, turning recovery into a large-scale brute-force problem. Some families rely on asymmetric encryption, where the private key is never present on the victim system. Others use custom cryptographic implementations that bypass operating system APIs, limiting visibility for key-harvesting tools.

> **RAAS · TWO-LAYER KEY STRUCTURE**
>
> In ransomware-as-a-service ecosystems, two layers of keys are often used. One key encrypts the per-file decryption keys and is controlled by ransom operators; the other encrypts the files themselves and is controlled by toolkit operators. This structure prevents insiders from stealing ransom payments and further complicates recovery.

§ 07 · CONCLUSION

Fewer places to hide. Fewer ways to evolve.

Using a variety of ransomware detection techniques increases defensive resilience, but selecting the right combination remains challenging. Each approach introduces tradeoffs that attackers actively exploit. A deeper understanding of how ransomware interacts with defensive tools enables organizations to balance risk and optimize resource deployment.

Quell's strategy focuses on eliminating the attacker's ability to adapt. By operating at the final stage of malicious encryption where files must be written to gain extortion leverage, file system–level detection strips away complexity and reduces opportunities for evasion.

At this depth, there are fewer places to hide and fewer ways for attackers to evolve.

LAYER	OVERHEAD	UPDATES
File system	**Minimal**	**None**
Inline write analysis	Few bytes per decision	Variant-agnostic by design