

Static and Behavioral Evidence in Mod Malware Analysis

Static inspection and observed runtime behavior answer different questions; combining them requires careful scope, provenance and uncertainty.

Publication: October 2026 | Technical analysis, not an experimental benchmark | Read online: research.malguard.site/papers/static-vs-behavioral-analysis.html

Two kinds of evidence

Static analysis describes an artifact without executing its code. It can inventory file types, hashes, metadata, embedded strings and package structure. Behavioral analysis observes what software does in a controlled environment. Neither method is a complete description of every possible future behavior.

For game mods, the distinction is important because packages may include inert assets, scripts interpreted by a game engine or native libraries loaded into a running process.

What static inspection can establish

Static inspection can detect format mismatches, unexpected executable components, known indicators and suspicious packaging choices. These findings are useful for triage. However, a string mentioning a network address is not proof of communication, and the absence of a known indicator is not proof of harmlessness.

Obfuscation, packaging and incomplete parsers can reduce visibility. Static results should state exactly which components and formats were inspected.

What behavioral observation adds

A properly controlled observation can show process creation, file modifications, module loading or network requests that actually occurred during a defined test. It can support claims about that particular execution context, not about every environment or trigger condition.

Behavior depends on inputs, game version, privileges, network access and time. A program that appeared inactive during one short observation might behave differently under other conditions. This is

why 'nothing observed' must be scoped to the test.

Safety and ethical boundaries

Malware testing belongs in appropriately isolated, authorized environments managed by trained practitioners. Production machines, personal accounts and ordinary gaming sessions are not suitable test beds. This publication describes evidence categories rather than procedures for executing malicious samples.

A defensive product should fail closed when isolation or required analysis infrastructure is unavailable, and it should never silently substitute a simulated result for an observed one.

Combining results without overstating certainty

A report should distinguish static indicators, observed behaviors and external reputation. For each item, record method, source, time and coverage. Conflicting evidence should be visible. When no behavioral analysis occurred, label the field 'not performed', not 'no malicious behavior'.

An explicit unknown category is necessary. Converting unknown into safe is a reporting error, not a user-friendly simplification.

Limits and future validation

This article is a conceptual methodology note. It contains no real malware execution, accuracy benchmark, detection claims or independently validated performance figures. A future study would require ethical approval, controlled samples, ground-truth definitions and a transparent reproducibility plan.

Example of scoped reporting

Consider a hypothetical package containing image assets and a native library. A static inventory may accurately report the presence of the library, its file digest and its declared filename. Those facts do not establish that the library ran, contacted a network service or performed an unwanted action. A separate authorized behavioral observation would be required to support claims about an actual execution.

If the observation environment was unavailable, the appropriate record is “behavior not assessed.” If an authorized test observed no network requests during a specified interval, report the duration, monitored interface and conditions. Neither outcome establishes that network activity is impossible in other circumstances.

Reporting matrix for defensive tools

Every result should identify the question, method, observation, limitations and status. Suggested statuses are observed, not observed within stated coverage, not assessed, and inconclusive. A high-confidence label requires a documented basis and should not be inferred from the absence of a known signature.

Disagreement between static indicators and behavior is a reason to preserve both records, not to hide one. Repeatability depends on stable artifact identity, recorded tool versions and clearly described analysis boundaries. This note proposes reporting principles; it does not claim that MalGuard has empirically validated a detection engine.

References and further reading

These external references support background concepts. The analysis and examples above are original to MalGuard Research; they are not attributed experiments.

- MITRE ATT&CK: Hijack Execution Flow, DLL · <https://attack.mitre.org/techniques/T1574/002/>
- OWASP File Upload Cheat Sheet · https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html