RUNSAFE SECURITY

RunSafe Identify vs. FOSSA

RunSafe Identify provides complete SBOM generation and risk analysis for native-code systems, where FOSSA encounters major accuracy and coverage gaps. Identify offers build-time visibility, reachability filtering, and exploitability modeling—capabilities unavailable in FOSSA.

KEY DIFFERENTIATORS

- Build-time SBOM generation for C/C++ without package managers.
- Automatic OS-package ingestion and vulnerability correlation.
- Reachability analysis to reduce false positives.
- Zero-day exploitability modeling for syscall-level risk.
- Integrated pathway to memory-safety mitigation via RunSafe Protect.
- Flexible ingestion models: manual, on-prem, GitHub action, or API.
- Better alignment with FDA, medical-device, and robotics markets.

	RunSafe Identify	FOSSA
Full C/C++ build-time SBOM extraction	✓	✗ *Limited; partial heuristics*
OS package enumeration	✓	✗ *Often missing; requires manual additions*
Reachability filtering	✓	✗ *Not available*
Zero-day exploitability modeling	✓	✗ *Not available*
Memory-safety mitigation mapping	✓	✗ *Not available*
Flexible ingestion workflows	✓	✗ *Primarily SaaS + CLI*
Embedded/regulated environment suitability	✓	✓ *Strong*

Native Code SBOM Generation

RunSafe Identify integrates directly into the build environment, enabling precise enumeration of C/C++ components, headers, object files, and OS-level packages. This avoids the limitations of package-manager–driven tools. FOSSA, in contrast, struggles with unmanaged code and requires manual modeling for C/C++ and embedded environments.

Accuracy and Coverage

RunSafe Identify captures full build-time dependency graphs and operating system packages automatically. FOSSA customers frequently report missing OS packages and incomplete coverage of nested native dependencies.

Reachability and Vulnerability Reduction

RunSafe Identify recently introduced reachability analysis for C/C++ components, enabling suppression of non-reachable vulnerabilities (e.g., kernel modules not shipped in final artifacts). FOSSA lacks an equivalent capability and treats all vulnerabilities as reachable.

Zero-Day Exploitability Modeling

RunSafe Identify models syscall-level exploitability paths and quantifies how memory-safety protections reduce zero-day impact. FOSSA does not perform exploitability modeling and relies solely on CVE presence.

Integration into Consulting Workflows

RunSafe Identify supports manual uploads, on-prem build integration, and low-privilege GitHub ingestion. This aligns tightly with consulting workflows where clients may provide only zip files or SBOMs.

Compliance and FDA Considerations

With CycloneDX outputs, VEX data, reachability filtering, and exploitability metrics, RunSafe Identify provides high-value insights for regulated submissions. FOSSA provides standard SBOM and CVE lists but lacks the deeper context and mitigation pathways.

CVE RISK REDUCTION
DATE OF SBOM ANALYSIS: 1/1/25
100%
Memory Safety Vulnerability Reduction
with RunSafe Applied
77%
Critical/High Vulnerability Reduction
with RunSafe Applied
RELATED TO MEMORY SAFETY
37% related to memory safety
Critical
68%
High
36%
TOTAL # OF VULNERABILITIES
2,000
↑ 82% vs last montl
ZERO-DAY EXPOSURE
Remote Command Execution 31%
18%
31%
14%
22%
15%
1.6m
ROP Chains Found
1: Remote Command Execution
2: Remote File Manipulation
3: Local File System Manipulation
4: Local Manipulation, Indirect PrivEse
5: Limited System Manipulation

68% of critical vulnerabilities related to memory safety

36% of high vulnerabilities related to memory safety

CONCLUSION

RunSafe Identify offers significantly deeper native-code visibility, actionable vulnerability insights, and risk-reduction analytics. For organizations requiring accurate SBOMs in embedded or regulated environments, Identify delivers capabilities far beyond FOSSA's traditional SaaS scanning model.