Security & Production Safety of yc- 360 Script

Tier1app

Last updated: Jan’ 17, 2026

Is yc-360 Open-Source Script Secure to Run in Production?

Production troubleshooting in regulated environments requires deep diagnostic visibility without increasing operational or security risk. The yc-360 script was designed to capture comprehensive production artifacts while maintaining strict governance, minimal footprint, and clearly bounded execution.

This document explains the architectural and security principles that make yc-360 suitable for highly secure production environments.

What Is yc-360 Script?

When a production incident occurs, monitoring systems generate alerts. SRE engineers then log into the affected devices and manually collect troubleshooting artifacts such as:

- Thread dumps
- Heap dumps
- Application logs
- System metrics (top, vmstat, iostat)
- Network statistics (netstat)
- Disk usage
- Kernel logs

Under incident pressure, artifacts are often inconsistently collected or partially captured. In some cases, applications are restarted before diagnostic data is preserved, leaving development teams without sufficient evidence to determine root cause.

yc-360 standardizes this process. It is a lightweight, incident-triggered script that captures a comprehensive set of production artifacts in a structured, repeatable, and governance-aligned manner.

Architecture Overview

yc-360 follows a simple, externally executed, incident-driven architecture designed to minimize operational and security risk.

When the script is invoked, it performs the following sequence:

1. Executes standard operating system and runtime diagnostic commands to collect production state information.
2. Stores captured artifacts locally on the host system.
3. If explicitly configured, securely transmits artifacts to a designated yCrash server instance.
4. Terminates cleanly after artifact collection completes.

No persistent process remains on the system after execution.

yc-360 does not open listening ports, create background services, register startup tasks, or embed into application runtimes. All diagnostic data is gathered using officially supported OS and runtime utilities already available to administrators.

Network communication occurs only if explicitly configured. The script performs no continuous monitoring and maintains no long-term runtime presence.

This architecture intentionally avoids the persistent agent model commonly used by APM platforms, thereby reducing long-term operational exposure.

Security Architecture Framework

The security posture of yc-360 can be understood through three primary control domains:

1. Execution Model – Non-Intrusive and Non-Persistent
2. Permission Model – No Expansion of Privilege Boundary
3. Data Control Model – Enterprise-Governed Artifact Handling

---

1) Execution Model – Non-Intrusive and Non-Persistent

yc-360 operates as an externally executed diagnostic utility. It runs on the same host, sidecar, or container as the target application but does not embed itself within the runtime environment (JVM, .NET, Python, etc.).

It does not:

- Attach a -javaagent
- Inject bytecode
- Intercept method calls
- Modify JVM flags
- Alter application binaries

Unlike modern APM tools, which typically require persistent machine-level agents and runtime-level instrumentation yc-360 does not integrate into the application execution path. APM platforms often operate continuously, intercept method calls, stream telemetry, and maintain a long-term runtime presence inside production systems. yc-360 avoids this model entirely.

It performs no runtime interception, no continuous monitoring, and no deep integration into application code paths. The operational exposure window exists only during the brief artifact collection phase.

This non-intrusive, non-persistent execution model significantly reduces long-term attack surface and avoids the risk category associated with continuous instrumentation-based monitoring tools.

---

2) Permission Model – No Expansion of Privilege Boundary

yc-360 does not require elevated privileges beyond what administrators already use for manual troubleshooting. It operates under the same user-level permissions as the target application or the engineer responsible for incident response.

The script invokes standard, officially supported diagnostic utilities such as:

- jstack
- jmap
- jcmd
- top
- vmstat
- netstat
- dmesg

These tools are already present in the operating system and runtime environment. yc-360 does not introduce kernel modules, root-level services, or privilege escalation mechanisms.

In the absence of yc-360, engineers would execute these same commands manually or via custom scripts and third-party tools, often requiring similar or broader permissions. yc-360 does not expand the existing permission boundary; it standardizes and orchestrates approved diagnostic actions within the established security model of the production environment.

3) Data Control Model – Enterprise-Governed Artifact Handling

Production artifacts such as heap dumps, thread dumps, and logs may contain sensitive information depending on application behavior. yc-360 is designed with this assumption and ensures governance authority remains fully with the enterprise.

Artifact transmission is configurable and can be disabled entirely for fully on-premises deployments. No outbound communication occurs unless explicitly enabled.

When transmission is configured:

- Data is encrypted in transit using AES-256 GCM with authenticated encryption.
- Encryption keys are exchanged using RSA public-key cryptography.
- Only the designated yCrash server can decrypt the artifacts.

Enterprises retain full control over:

- Storage location
- Retention duration

- Access permissions
- Encryption standards
- Proxy routing

Optional heap dump sanitization capabilities allow masking or removal of sensitive fields when regulatory requirements mandate data minimization.

yc-360 does not continuously stream telemetry, does not retain persistent background copies, and does not initiate uncontrolled outbound connections. All artifact handling remains under enterprise-defined governance policies.

Operational Impact

When triggered, yc-360 completes artifact collection within a short, bounded execution window, typically under 30 seconds depending on environment size and configuration. During this time, it invokes standard diagnostic utilities to capture system state.

Resource consumption is temporary and limited strictly to the collection phase. Once execution is completed, no CPU, memory, or network resources continue to be consumed.

Because yc-360 does not run as a persistent agent and does not intercept application calls, it avoids sustained performance overhead commonly associated with continuous monitoring solutions.

Performance validation details are available here:
https://blog.ycrash.io/ycrash-agent-overhead-performance/

Open-Source Transparency

yc-360 is distributed under the Apache 2.0 license and is fully open source:
https://github.com/ycrash/yc-360-script

Security teams may:

- Review the complete source code
- Audit executed commands
- Perform static and dynamic analysis
- Fork and host internally
- Apply checksum validation prior to execution

There are no hidden binaries or opaque runtime components. The script operates entirely within existing OS permission boundaries.

Compliance Alignment

yc-360 supports key regulatory and governance principles including:

- Least privilege
- Change management discipline
- Data minimization
- Controlled transmission
- Auditability

Because it introduces no persistent runtime service, no configuration changes, and no uncontrolled network egress, it aligns with security expectations commonly required in regulated industries.

Information Security Management Certification

The yCrash Enterprise Edition server environment operates under an Information Security Management System (ISMS) certified to ISO/IEC 27001. This certification covers defined operational, administrative, and technical controls governing artifact handling, encryption practices, access management, and infrastructure security within the certified scope.

Where yc-360 transmits artifacts to a yCrash Enterprise server, those artifacts are processed within this ISO 27001-certified security framework. Key management, access control policies, and operational safeguards are governed by the controls defined in the certified ISMS.

Enterprises retain authority over whether transmission is enabled and may operate yc-360 entirely on-premises if desired.

Risk Profile Compared to Continuous Monitoring Agents

Risk Category	yc-360	Continuous APM Agent
Persistent runtime presence	No	Yes
Bytecode instrumentation	No	Yes
Continuous network egress	No	Yes
Long-term data streaming	No	Yes
Operational footprint	Temporary	Permanent

yc-360 reduces long-term exposure by operating only during incident-triggered execution.

Conclusion

yc-360 was engineered for environments where production stability, security governance, and compliance discipline are critical.

It replaces ad-hoc artifact collection with a structured, controlled, and externally executed process, without introducing persistent agents, runtime interception, or uncontrolled data flows.

- Its execution is non-intrusive and temporary.
- Its privilege model is bound.
- Its data handling is enterprise-governed.

This architecture provides a low-risk method for capturing comprehensive production diagnostics while preserving runtime integrity and regulatory alignment.

Appendix A: STRIDE Threat Model Analysis

This appendix evaluates the yc-360 script using the STRIDE threat modeling framework. STRIDE classifies risks across six categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

The assessment below reflects yc-360's non-intrusive, non-persistent, externally executed design.

System Context

Scope Evaluated:

- yc-360 script execution on production host
- Diagnostic artifact collection
- Optional encrypted artifact transmission
- Artifact storage and handling

Out of Scope:

- Application code security
- Operating system baseline security
- Network perimeter defenses

yc-360 operates within existing OS permission boundaries and relies exclusively on standard diagnostic utilities.

---

1. Spoofing Identity

Threat Description:
An attacker impersonates a legitimate entity (user, server, or process) to gain unauthorized access to artifacts or influence execution.

Risk Evaluation:

- yc-360 does not open listening ports.
- It does not expose network services.
- It does not accept inbound connections.
- Artifact transmission (if enabled) uses RSA public-key cryptography for key exchange.
- Only the designated server with the corresponding private key can decrypt artifacts.

Mitigation Characteristics:

- No externally exposed identity surface.
- No runtime authentication mechanism to spoof.
- Encryption handshake prevents unauthorized decryption.

Residual Risk:
Low — dependent on enterprise key management practices.

---

2. Tampering with Data

Threat Description:
Modification of artifacts in transit or alteration of system state by the script.

Risk Evaluation:

- yc-360 performs read-only diagnostic collection.
- It does not modify application data or configuration.
- Encryption uses AES-256 GCM, which provides authenticated encryption with tamper detection.
- No runtime injection or system configuration changes occur.

Mitigation Characteristics:

- Authenticated encryption prevents undetected modification during transmission.
- No persistent system changes.
- Script terminates after execution.

Residual Risk:
Low — limited to host-level compromise scenarios outside script scope.

---

3. Repudiation

Threat Description:
An actor denies having executed the script or denies data transmission.

Risk Evaluation:

- Execution is manual or controlled by incident workflow.
- Artifact creation occurs locally and can be logged by system audit mechanisms.
- Transmission (if enabled) is observable via enterprise logging controls.

Mitigation Characteristics:

- Enterprise logging and audit controls can record execution events.

- No hidden background execution.
- No autonomous scheduling behavior.

Residual Risk:
Low — subject to enterprise audit configuration.

---

4. Information Disclosure

Threat Description:
Sensitive data exposure through artifact collection or transmission.

Risk Evaluation:

- Diagnostic artifacts (heap dumps, logs) may contain sensitive information depending on application behavior.
- Transmission is configurable and can be disabled for on-prem operation.
- Encryption in transit uses AES-256 GCM.
- RSA key exchange ensures only the designated server can decrypt artifacts.
- Optional heap dump sanitization supports data minimization.

Mitigation Characteristics:

- Encryption protects data in transit.
- Enterprise controls retention, storage, and access.
- No continuous data streaming.
- No uncontrolled outbound communication.

Residual Risk:
Medium — inherent to diagnostic artifact handling, mitigated by encryption and governance controls.

---

5. Denial of Service (DoS)

Threat Description:
Script execution causes performance degradation or system instability.

Risk Evaluation:

- Execution window is short and bounded (typically under 30 seconds).
- No continuous runtime presence.
- No persistent CPU or memory consumption.

- Uses standard diagnostic tools already available to administrators.

Mitigation Characteristics:

- Temporary execution.
- No continuous interception.
- No runtime instrumentation.
- Resource impact limited to artifact capture phase.

Residual Risk:
Low — equivalent to manual execution of standard diagnostic commands.

---

6. Elevation of Privilege

Threat Description:
The script gains higher privileges than intended or expands access boundaries.

Risk Evaluation:

- yc-360 runs under the same OS user context as the executing administrator.
- No privilege escalation mechanisms.
- No kernel modules or root services installed.
- No modification of OS permission model.

Mitigation Characteristics:

- Operates within existing privilege boundaries.
- No expansion of trust domain.
- No runtime injection.

Residual Risk:
Low — equivalent to standard manual troubleshooting activity.

STRIDE Summary Matrix

Threat Category	Risk Level	Mitigation Basis
Spoofing	Low	No exposed services, RSA key exchange
Tampering	Low	AES-256 GCM authenticated encryption

Threat Category	Risk Level	Mitigation Basis
Repudiation	Low	Enterprise audit logging
Information Disclosure	Medium (controlled)	Encryption + sanitization + configurable transmission
Denial of Service	Low	Short, bounded execution window
Elevation of Privilege	Low	No privilege expansion

Overall STRIDE Assessment

yc-360 introduces no persistent attack surface, no runtime instrumentation risk, no privilege escalation vector, and no continuous network exposure. The primary inherent risk relates to handling of diagnostic artifacts that may contain sensitive information; this risk is mitigated through encryption, optional sanitization, and enterprise-controlled governance policies.

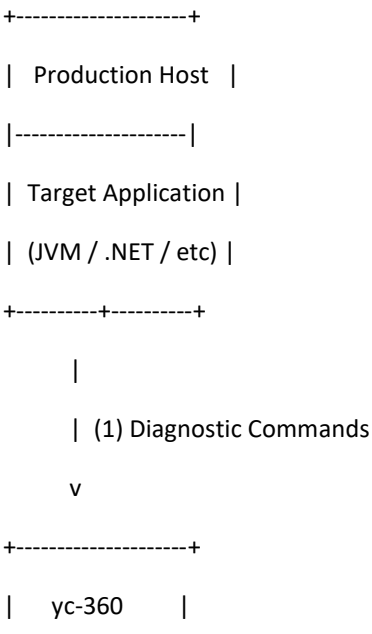
From a STRIDE perspective, yc-360 aligns with a low-risk, externally executed diagnostic utility operating within established production security boundaries.

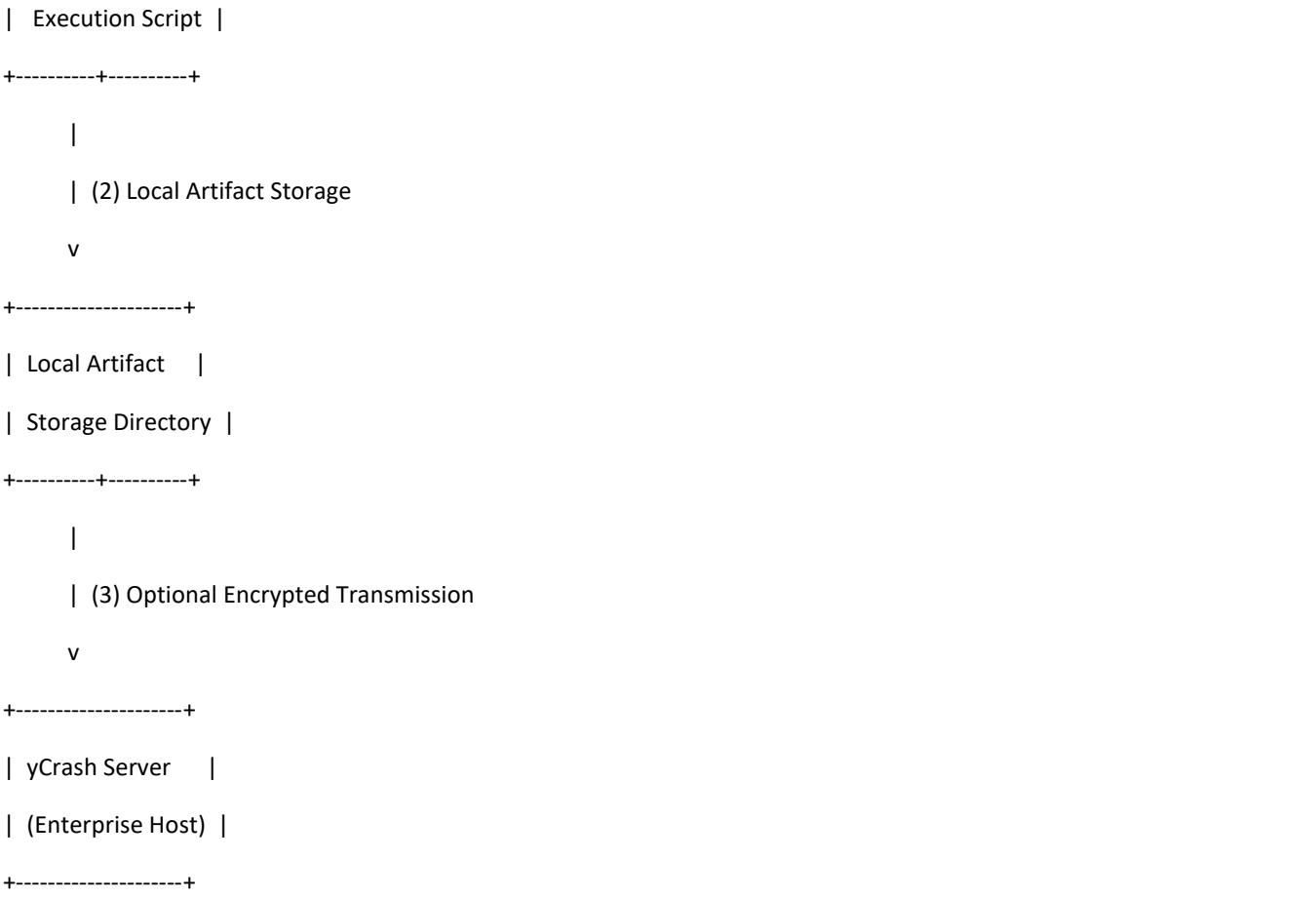
Appendix B: Data Flow Diagram (DFD)

System Overview

The following diagram illustrates the logical data flow when yc-360 is executed in a production environment.

Level 0 – High-Level Data Flow





Data Flow Explanation

Step 1 – Artifact Collection

yc-360 invokes standard OS and runtime diagnostic utilities such as jstack, jmap, jcmd, top, vmstat, and netstat. These commands gather runtime and system state information in read-only mode.

No inbound network connections are opened.
No runtime instrumentation is performed.

Step 2 – Local Storage

Artifacts are stored locally on the production host in a designated directory controlled by OS-level permissions.

No background process continues after execution.
No hidden replication occurs.

Step 3 – Optional Encrypted Transmission

If transmission is configured:

- Artifacts are encrypted using AES-256 GCM.
- Encryption keys are exchanged using RSA public-key cryptography.
- Only the designated yCrash server instance can decrypt artifacts.
- Transmission may be routed through enterprise proxy infrastructure.

If transmission is not configured, artifacts remain on-premises.

Trust Boundaries

The following trust boundaries exist:

1. **Production Host Boundary**
    - yc-360 runs under existing OS user privileges.
    - No privilege escalation occurs.
2. **Network Boundary (Optional)**
    - Encrypted outbound transmission only.
    - No inbound exposure.
    - No listening ports opened.
3. **Enterprise Analysis Server Boundary**
    - Decryption controlled by enterprise-managed key.

DFD Security Properties

- No persistent network listener
- No external inbound attack surface
- No runtime embedding
- No continuous telemetry streaming
- Enterprise-controlled encryption and storage

Appendix C: Frequently Asked Questions from Security Teams

Below are common questions raised during regulated enterprise reviews.

---

1. Does yc-360 introduce a new attack surface?

No. yc-360 does not open listening ports, expose APIs, create background services, or embed into application runtimes. It executes temporarily and terminates. No persistent runtime component remains.

---

2. Does it require root or elevated privileges?

No additional privileges beyond what administrators already use for manual troubleshooting are required. yc-360 operates within the existing OS permission boundary and does not introduce privilege escalation mechanisms.

---

3. How does it differ from APM agents already running in our environment?

APM tools typically:

- Run continuously
- Instrument application runtime
- Intercept method calls
- Stream telemetry

yc-360:

- Runs only when triggered
- Does not instrument runtime
- Does not intercept application calls
- Does not maintain continuous presence

It is an externally executed diagnostic utility, not a monitoring agent.

---

4. Can sensitive data be exposed?

Diagnostic artifacts may contain sensitive information depending on application behavior. yc-360 mitigates this risk through:

- Configurable transmission (can be disabled)
- AES-256 GCM encryption in transit
- RSA-based key exchange

- Optional heap dump sanitization
- Enterprise-controlled retention and access policies

No continuous data streaming occurs.

---

5. Where are encryption keys stored?

Encryption keys are exchanged using RSA public-key cryptography. The private key remains on the designated yCrash server controlled by the enterprise. Key management policies are governed by the organization's internal security framework.

---

6. Can yc-360 be run fully on-prem without outbound connectivity?

Yes. Transmission can be disabled entirely. In this configuration, artifacts remain local and are handled according to enterprise policy.

---

7. Can yc-360 modify system configuration or application state?

No. yc-360 performs read-only diagnostic collection using officially supported utilities. It does not modify configuration files, application binaries, runtime flags, or OS settings.

---

8. What happens if the script is interrupted?

Because yc-360 does not modify runtime behavior or system configuration, interruption does not impact application execution. Partial artifacts may exist locally, governed by OS permissions.

---

9. Does yc-360 remain installed after execution?

The script file itself may remain on disk as any standard utility would, but it does not install background services, startup tasks, or persistent runtime components. No resident process remains active.

---

10. Has yc-360 been used in regulated industries?

yc-360 has been deployed in financial services, healthcare, and other regulated enterprise environments. Its externally executed, non-persistent architecture was intentionally designed to meet the operational constraints of such environments.

Final Security Position

With:

- A bounded execution window
- No runtime instrumentation
- No privilege expansion
- No persistent attack surface
- Encrypted and configurable data transmission
- Enterprise-controlled governance

yc-360 aligns with low-risk diagnostic utility principles under the STRIDE model.