ERR00-C. Adopt and implement a consistent and comprehensive error-handling policy

A secure system is invariably subject to stresses, such as those caused by attack, erroneous or malicious inputs, hardware or software faults, unanticipated user behavior, and unexpected environmental changes that are outside the bounds of "normal operation." Yet the system must continue to deliver essential services in a timely manner, safely and securely. To accomplish this, the system must exhibit qualities such as robustness, reliability, error tolerance, fault tolerance, performance, and security. All of these system-quality attributes depend on consistent and comprehensive error handling that supports the goals of the overall system.

ISO/IEC TR 24772, section 6.39.1 [ISO/IEC TR 24772], says:

"Expectations that a system will be dependable are based on the confidence that the system will operate as expected and not fail in normal use. The dependability of a system and its fault tolerance can be measured through the component part's reliability, availability, safety and security. Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time [IEEE 1990 Glossary]. Availability is how timely and reliable the system is to its intended users. Both of these factors matter highly in systems used for safety and security. In spite of the best intentions, systems may encounter a failure, either from internally poorly written software or external forces such as power outages/variations, floods, or other natural disasters. The reaction to a fault can affect the performance of a system and in particular, the safety and security of the system and its users."

Effective error handling (which includes error reporting, report aggregation, analysis, response, and recovery) is a central aspect of the design, implementation, maintenance, and operation of systems that exhibit survivability under stress. Survivability is the capability of a system to fulfill its mission, in a timely manner, despite an attack, accident, or other stress that is outside the bounds of normal operation [Lipson 2000]. If full services cannot be maintained under a given stress, survivable systems degrade gracefully, continue to deliver essential services, and recover full services as conditions permit.

Error reporting and error handling play a central role in the engineering and operation of survivable systems. Survivability is an emergent property of a system as a whole [Fisher 1999] and depends on the behavior of all of the system’s components and the interactions among them. From the viewpoint of error handling, every system component, down to the smallest routine, can be considered to be a sensor capable of reporting on some aspect of the health of the system. Any error or anomaly, ignored or improperly handled, can threaten delivery of essential system services and, as a result, put at risk the organizational or business mission that the system supports.

The key characteristics of survivability include the 3 Rs: resistance, recognition, and recovery. Resistance refers to measures that harden a system against particular stresses, recognition refers to situational awareness with respect to instances of stress and their impact on the system, and recovery is the ability of a system to restore services after (and possibly during) an attack, accident, or other event that has disrupted those services.

Recognition of the full nature of adverse events and the determination of appropriate measures for recovery and response are often not possible in the context of the component or routine in which a related error first manifests. Aggregation of multiple error reports and the interpretation of those reports in a higher context may be required both to understand what is happening and to decide on the appropriate action to take. Of course, the domain-specific context in which the system operates plays a huge role in determining proper recovery strategies and tactics. For safety-critical systems, simply halting the system (or even just terminating an offending process) in response to an error is rarely the best course of action and may lead to disaster. From a system perspective, error-handling strategies should map directly into survivability strategies, which may include recovery by activating fully redundant backup services or by providing alternative sets of roughly equivalent services that fulfill the mission with sufficient diversity to greatly improve the odds of survival against common mode failures.

An error-handling policy must specify a comprehensive approach to error reporting and response. Components and routines should always generate status indicators, and all called routines should have their error returns checked. All input should be checked for compliance with the formal requirements for such input rather than be blindly trusted. Moreover, never assume, on the basis of specific knowledge about the system or its domain, that the success of a called routine is guaranteed. The failure to report or properly respond to errors or other anomalies from a system perspective can threaten the survivability of the system as a whole.

ISO/IEC TR 24772:2013, section 6.39.5 [ISO/IEC TR 24772:2013], describes the following mitigation strategies:

1. A strategy for fault handling should be decided. Consistency in fault handling should be the same with respect to critically similar parts.
2. A multi-tiered approach of fault prevention, fault detection, and fault reaction should be used.
3. System-defined components that assist in uniformity of fault handling should be used when available. For one example, designing a “runtime constraint handler” (as described in Annex K of the C Standard) permits the application to intercept various erroneous situations and perform one consistent response, such as flushing a previous transaction and restarting at the next one.
4. When there are multiple tasks, a fault-handling policy should be specified whereby a task may
   - halt, and keep its resources available for other tasks (perhaps permitting restarting of the faulting task)
   - halt, and remove its resources (perhaps to allow other tasks to use the resources so freed, or to allow a recreation of the task)
   - halt, and signal the rest of the program to likewise halt

Risk Assessment
Failure to adopt and implement a consistent and comprehensive error-handling policy is detrimental to system survivability and can result in a broad range of vulnerabilities depending on the operational characteristics of the system.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Remediation Cost</th>
<th>Priority</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR00-C</td>
<td>Medium</td>
<td>Probable</td>
<td>High</td>
<td>P4</td>
<td>L3</td>
</tr>
</tbody>
</table>

**Related Vulnerabilities**

Search for vulnerabilities resulting from the violation of this rule on the CERT website.

**Automated Detection**

| Tool                | Version | Checker                        | Description                                                          |
|---------------------|---------|--------------------------------|                                                                     |
| Polyspace Bug Finder| R2018a  | MISRA C:2012 Rule 17.1         | The features of `<stdarg.h>` shall not be used                      |

**Related Guidelines**

<table>
<thead>
<tr>
<th>ISO/IEC TR 24772:2013</th>
<th>Termination Strategy [REU]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MISRA C:2012</td>
<td>Rule 17.1 (required)</td>
</tr>
<tr>
<td>MITRE CWE</td>
<td>CWE-391, Unchecked error condition</td>
</tr>
<tr>
<td></td>
<td>CWE-544, Missing standardized error handling mechanism</td>
</tr>
</tbody>
</table>

**Bibliography**

[Fisher 1999]

[Horton 1990] Section 11, p. 168  
Section 14, p. 254

[Koenig 1989] Section 5.4, p. 73

[Lipson 2000]

[Lipson 2006]

[Summit 2005] C-FAQ Question 20.4