# CWE Detail – CWE-1329

## Description

The product contains a component that cannot be updated or patched in order to remove vulnerabilities or significant bugs.

## Extended Description

If the component is discovered to contain a vulnerability or critical bug, but the issue cannot be fixed using an update or patch, then the product's owner will not be able to protect against the issue. The only option might be replacement of the product, which could be too financially or operationally expensive for the product owner. As a result, the inability to patch or update can leave the product open to attacker exploitation or critical operation failures. This weakness can be especially difficult to manage when using ROM, firmware, or similar components that traditionally have had limited or no update capabilities. In industries such as healthcare, "legacy"
 devices can be operated for decades. As a
 US task force report [REF-1197] notes, "the inability
 to update or replace equipment has both
 large and small health care delivery
 organizations struggle with numerous
 unsupported legacy systems that cannot
 easily be replaced (hardware, software, and
 operating systems) with large numbers of
 vulnerabilities and few modern
 countermeasures." While hardware can be prone to this weakness, software systems can also be affected, such as when a third-party driver or library is no longer actively maintained or supported but is still critical for the required functionality.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2020-9054: Chain: network-attached storage (NAS) device has a critical OS command injection (CWE-78) vulnerability that is actively exploited to place IoT devices into a botnet, but some products are "end-of-support" and cannot be patched (CWE-1277). [REF-1097] (KEV)

## Modes of Introduction

**•** Requirements: Requirements development might not consider the importance of updates over the lifetime of the product or might intentionally exclude this capability due to concerns such as expense or speed to market.

**•** Architecture and Design: Lack of planning during architecture development and design, or external pressures such as speed to market, could ignore the capability to update.

**•** Architecture and Design: Designers might omit capabilities for updating a component due to time pressures to release the product or assumptions about the stability of the component.

**•** Implementation: The weakness can appear through oversight during implementation.

## Common Consequences

**•** Impact: Gain Privileges or Assume Identity, Bypass Protection Mechanism, Execute Unauthorized Code or Commands, DoS: Crash, Exit, or Restart, Quality Degradation, Reduce Maintainability — Notes: If an attacker can identify an exploitable vulnerability in one product that has no means of patching, the attack may be used against all affected versions of that product.

## Potential Mitigations

**•** Requirements: Specify requirements that each component should be updateable, including ROM, firmware, etc. (Effectiveness: N/A)

**•** Architecture and Design: Design the product to allow for updating of its components. Include the external infrastructure that might be necessary to support updates, such as distribution servers. (Effectiveness: N/A)

**•** Architecture and Design: With hardware, support patches that can be programmed in-field or during manufacturing through hardware fuses. This feature can be used for limited patching of devices after shipping, or for the next batch of silicon devices manufactured, without changing the full device ROM. (Effectiveness: Moderate)

**•** Implementation: Implement the necessary functionality to allow each component to be updated. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** N/A

**•** ROM does not have built-in application-programming interfaces (APIs) to patch if the code is vulnerable. Implement mechanisms to patch the vulnerable ROM code.

**•** However, the SoC's crypto key is hardcoded into the design and cannot be updated [REF-1387]. Therefore, if the key is leaked somehow, there is no way to reprovision the key without having the device replaced.