# CWE Detail – CWE-1220

## Description

The product implements access controls via a policy or other feature with the intention to disable or restrict accesses (reads and/or writes) to assets in a system from untrusted agents. However, implemented access controls lack required granularity, which renders the control policy too broad because it allows accesses from unauthorized agents to the security-sensitive assets.

## Extended Description

Integrated circuits and hardware engines can expose accesses to assets (device configuration, keys, etc.) to trusted firmware or a software module (commonly set by BIOS/bootloader). This access is typically access-controlled. Upon a power reset, the hardware or system usually starts with default values in registers, and the trusted firmware (Boot firmware) configures the necessary access-control protection. A common weakness that can exist in such protection schemes is that access controls or policies are not granular enough. This condition allows agents beyond trusted agents to access assets and could lead to a loss of functionality or the ability to set up the device securely. This further results in security risks from leaked, sensitive, key material to modification of device configuration.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2022-24985: A form hosting website only checks the session authentication status for a single form, making it possible to bypass authentication when there are multiple forms

**•** CVE-2021-36934: An operating system has an overly permission Access Control List onsome system files, including those related to user passwords (KEV)

## Related Attack Patterns (CAPEC)

* CAPEC-1
* CAPEC-180

## Attack TTPs

**•** T1574.010: Services File Permissions Weakness (Tactics: persistence, privilege-escalation, defense-evasion)

## Modes of Introduction

**•** Architecture and Design: Such issues could be introduced during hardware architecture and design and identified later during Testing or System Configuration phases.

**•** Implementation: Such issues could be introduced during hardware implementation and identified later during Testing or System Configuration phases.

## Common Consequences

**•** Impact: Modify Memory, Read Memory, Execute Unauthorized Code or Commands, Gain Privileges or Assume Identity, Bypass Protection Mechanism, Other — Notes:

## Potential Mitigations

**•** Architecture and Design: Access-control-policy protections must be reviewed for design inconsistency and common weaknesses. Access-control-policy definition and programming flow must be tested in pre-silicon, post-silicon testing. (Effectiveness: High)

## Applicable Platforms

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

## Demonstrative Examples

**•** In the above example, there is only one policy register that controls access to both read and write accesses to the AES-key registers, and thus the design is not granular enough to separate read and writes access for different agents. Here, agent with identities "1" and "2" can both read and write.

**•** The AXI ensures that only users with appropriate privileges can access specific peripherals. For instance, a ROM module is accessible exclusively with Machine privilege, and AXI enforces that users attempting to read data from the ROM must possess machine privilege; otherwise, access to the ROM is denied. The access control information and configurations are stored in a ROM.

**•** The security policy access control is not granular enough, as it uses one bit to enable both
 read and write access. This gives write access to an area that should only be readable
 by unprivileged agents. Access control logic should differentiate between read and write access and to have
 sufficient address granularity.