# CWE Detail – CWE-284

## Description

The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

## Extended Description

Access control involves the use of several protection mechanisms such as: Authentication (proving the identity of an actor) Authorization (ensuring that a given actor can access a resource), and Accountability (tracking of activities that were performed) When any mechanism is not applied or otherwise fails, attackers can compromise the security of the product by gaining privileges, reading sensitive information, executing commands, evading detection, etc. There are two distinct behaviors that can introduce access control weaknesses: Specification: incorrect privileges, permissions, ownership, etc. are explicitly specified for either the user or the resource (for example, setting a password file to be world-writable, or giving administrator capabilities to a guest user). This action could be performed by the program or the administrator. Enforcement: the mechanism contains errors that prevent it from properly enforcing the specified access control requirements (e.g., allowing the user to specify their own privileges, or allowing a syntactically-incorrect ACL to produce insecure settings). This problem occurs within the program itself, in that it does not actually enforce the intended security policy that the administrator specifies.

## Threat-Mapped Scoring

Score: 3.25

Priority: P2 - Serious (High)

## 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-2022-29238: Access-control setting in web-based document collaboration tool is not properly implemented by the code, which prevents listing hidden directories but does not prevent direct requests to files in those directories.

**•** CVE-2022-23607: Python-based HTTP library did not scope cookies to a particular domain such that "supercookies" could be sent to any domain on redirect

**•** CVE-2021-21972: Chain: Cloud computing virtualization platform does not require authentication for upload of a tar format file (CWE-306), then uses .. path traversal sequences (CWE-23) in the file to access unexpected files, as exploited in the wild per CISA KEV. (KEV)

**•** CVE-2021-37415: IT management product does not perform authentication for some REST API requests, as exploited in the wild per CISA KEV. (KEV)

**•** CVE-2021-35033: Firmware for a WiFi router uses a hard-coded password for a BusyBox shell, allowing bypass of authentication through the UART port

**•** CVE-2020-10263: Bluetooth speaker does not require authentication for the debug functionality on the UART port, allowing root shell access

**•** CVE-2020-13927: Default setting in workflow management product allows all API requests without authentication, as exploited in the wild per CISA KEV. (KEV)

**•** CVE-2010-4624: Bulletin board applies restrictions on number of images during post creation, but does not enforce this on editing.

## Related Attack Patterns (CAPEC)

* CAPEC-19
* CAPEC-441
* CAPEC-478
* CAPEC-479
* CAPEC-502
* CAPEC-503
* CAPEC-536
* CAPEC-546
* CAPEC-550
* CAPEC-551
* CAPEC-552
* CAPEC-556
* CAPEC-558
* CAPEC-562
* CAPEC-563
* CAPEC-564
* CAPEC-578

## Attack TTPs

**•** T1037: Boot or Logon Initialization Scripts (Tactics: persistence, privilege-escalation)

**•** T1027.009: Embedded Payloads (Tactics: defense-evasion)

**•** T1014: Rootkit (Tactics: defense-evasion)

**•** T1543: Create or Modify System Process (Tactics: persistence, privilege-escalation)

**•** T1574.011: Services Registry Permissions Weakness (Tactics: persistence, privilege-escalation, defense-evasion)

**•** T1542.003: Bootkit (Tactics: persistence, defense-evasion)

**•** T1547: Boot or Logon Autostart Execution (Tactics: persistence, privilege-escalation)

**•** T1080: Taint Shared Content (Tactics: lateral-movement)

**•** T1562.009: Safe Mode Boot (Tactics: defense-evasion)

**•** T1543.003: Windows Service (Tactics: persistence, privilege-escalation)

**•** T1505.005: Terminal Services DLL (Tactics: persistence)

**•** T1562.002: Disable Windows Event Logging (Tactics: defense-evasion)

**•** T1562.004: Disable or Modify System Firewall (Tactics: defense-evasion)

**•** T1543.004: Launch Daemon (Tactics: persistence, privilege-escalation)

**•** T1546.008: Accessibility Features (Tactics: privilege-escalation, persistence)

**•** T1562.007: Disable or Modify Cloud Firewall (Tactics: defense-evasion)

**•** T1546.001: Change Default File Association (Tactics: privilege-escalation, persistence)

**•** T1547.006: Kernel Modules and Extensions (Tactics: persistence, privilege-escalation)

**•** T1562.001: Disable or Modify Tools (Tactics: defense-evasion)

**•** T1556.006: Multi-Factor Authentication (Tactics: credential-access, defense-evasion, persistence)

**•** T1546.004: Unix Shell Configuration Modification (Tactics: privilege-escalation, persistence)

**•** T1553.004: Install Root Certificate (Tactics: defense-evasion)

**•** T1562.008: Disable or Modify Cloud Logs (Tactics: defense-evasion)

**•** T1543.001: Launch Agent (Tactics: persistence, privilege-escalation)

**•** T1546.016: Installer Packages (Tactics: privilege-escalation, persistence)

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

**•** Operation: N/A

## Common Consequences

**•** Impact: Varies by Context — Notes:

## Potential Mitigations

**•** Architecture and Design: Very carefully manage the setting, management, and handling of privileges. Explicitly manage trust zones in the software. (Effectiveness: N/A)

**•** Architecture and Design: Compartmentalize the system to have "safe" areas where trust boundaries can be unambiguously drawn. Do not allow sensitive data to go outside of the trust boundary and always be careful when interfacing with a compartment outside of the safe area. Ensure that appropriate compartmentalization is built into the system design, and the compartmentalization allows for and reinforces privilege separation functionality. Architects and designers should rely on the principle of least privilege to decide the appropriate time to use privileges and the time to drop privileges. (Effectiveness: N/A)

## Notes

**•** Maintenance: This entry needs more work. Possible sub-categories include: Trusted group includes undesired entities (partially covered by CWE-286) Group can perform undesired actions ACL parse error does not fail closed