# CWE Detail – CWE-346

## Description

The product does not properly verify that the source of data or communication is valid.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2000-1218: DNS server can accept DNS updates from hosts that it did not query, leading to cache poisoning

**•** CVE-2005-0877: DNS server can accept DNS updates from hosts that it did not query, leading to cache poisoning

**•** CVE-2001-1452: DNS server caches glue records received from non-delegated name servers

**•** CVE-2005-2188: user ID obtained from untrusted source (URL)

**•** CVE-2003-0174: LDAP service does not verify if a particular attribute was set by the LDAP server

**•** CVE-1999-1549: product does not sufficiently distinguish external HTML from internal, potentially dangerous HTML, allowing bypass using special strings in the page title. Overlaps special elements.

**•** CVE-2003-0981: product records the reverse DNS name of a visitor in the logs, allowing spoofing and resultant XSS.

## Related Attack Patterns (CAPEC)

* CAPEC-111
* CAPEC-141
* CAPEC-142
* CAPEC-160
* CAPEC-21
* CAPEC-384
* CAPEC-385
* CAPEC-386
* CAPEC-387
* CAPEC-388
* CAPEC-510
* CAPEC-59
* CAPEC-60
* CAPEC-75
* CAPEC-76
* CAPEC-89

## Attack TTPs

**•** T1539: Steal Web Session Cookie (Tactics: credential-access)

**•** T1134.001: Token Impersonation/Theft (Tactics: defense-evasion, privilege-escalation)

**•** T1528: Steal Application Access Token (Tactics: credential-access)

**•** T1584.002: DNS Server (Tactics: resource-development)

**•** T1550.004: Web Session Cookie (Tactics: defense-evasion, lateral-movement)

**•** T1557.002: ARP Cache Poisoning (Tactics: credential-access, collection)

**•** T1134: Access Token Manipulation (Tactics: defense-evasion, privilege-escalation)

## Modes of Introduction

**•** Architecture and Design: N/A

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

## Common Consequences

**•** Impact: Gain Privileges or Assume Identity, Varies by Context — Notes: An attacker can access any functionality that is inadvertently accessible to the source.

## Applicable Platforms

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

## Demonstrative Examples

**•** This application does not check the origin of the intent, thus allowing any malicious application to remove a user. Always check the origin of an intent, or create an allowlist of trusted applications using the manifest.xml file.

**•** A call into native code can then be initiated by passing parameters within the URL:

## Notes

**•** Maintenance: This entry has some significant overlap with other CWE entries and may need some clarification. See terminology notes.

**•** Terminology: The "Origin Validation Error" term was originally used in a 1995 thesis [REF-324]. Although not formally defined, an issue is considered to be an origin validation error if either (1) "an object [accepts] input from an unauthorized subject," or (2) "the system [fails] to properly or completely authenticate a subject." A later section says that an origin validation error can occur when the system (1) "does not properly authenticate a user or process" or (2) "does not properly authenticate the shared data or libraries." The only example provided in the thesis (covered by OSVDB:57615) involves a setuid program running command-line arguments without dropping privileges. So, this definition (and its examples in the thesis) effectively cover other weaknesses such as CWE-287 (Improper Authentication), CWE-285 (Improper Authorization), and CWE-250 (Execution with Unnecessary Privileges). There appears to be little usage of this term today, except in the SecurityFocus vulnerability database, where the term is used for a variety of issues, including web-browser problems that allow violation of the Same Origin Policy and improper validation of the source of an incoming message.