# CWE Detail – CWE-405

## Description

The product does not properly control situations in which an adversary can cause the product to consume or produce excessive resources without requiring the adversary to invest equivalent work or otherwise prove authorization, i.e., the adversary's influence is "asymmetric."

## Extended Description

This can lead to poor performance due to "amplification" of resource consumption, typically in a non-linear fashion. This situation is worsened if the product allows malicious users or attackers to consume more resources than their access level permits.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-1999-0513: Classic "Smurf" attack, using spoofed ICMP packets to broadcast addresses.

**•** CVE-2003-1564: Parsing library allows XML bomb

**•** CVE-2004-2458: Tool creates directories before authenticating user.

**•** CVE-2020-10735: Python has "quadratic complexity" issue when converting string to int with many digits in unexpected bases

**•** CVE-2020-5243: server allows ReDOS with crafted User-Agent strings, due to overlapping capture groups that cause excessive backtracking.

**•** CVE-2013-5211: composite: NTP feature generates large responses (high amplification factor) with spoofed UDP source addresses.

**•** CVE-2002-20001: Diffie-Hellman (DHE) Key Agreement Protocol allows attackers to send arbitrary numbers that are not public keys, which causes the server to perform expensive, unnecessary computation of modular exponentiation.

**•** CVE-2022-40735: The Diffie-Hellman Key Agreement Protocol allows use of long exponents, which are more computationally expensive than using certain "short exponents" with particular properties.

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

**•** Operation: N/A

## Common Consequences

**•** Impact: DoS: Amplification, DoS: Resource Consumption (CPU), DoS: Resource Consumption (Memory), DoS: Resource Consumption (Other) — Notes: Sometimes this is a factor in "flood" attacks, but other types of amplification exist.

## Potential Mitigations

**•** Architecture and Design: An application must make resources available to a client commensurate with the client's access level. (Effectiveness: N/A)

**•** Architecture and Design: An application must, at all times, keep track of allocated resources and meter their usage appropriately. (Effectiveness: N/A)

**•** System Configuration: Consider disabling resource-intensive algorithms on the server side, such as Diffie-Hellman key exchange. (Effectiveness: High)

## Applicable Platforms

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

## Demonstrative Examples

**•** This code sends a DNS record to a requesting IP address. UDP allows the source IP address to be easily changed ('spoofed'), thus allowing an attacker to redirect responses to a target, which may be then be overwhelmed by the network traffic.

**•** This code first reads a specified file into memory, then prints the file if the user is authorized to see its contents. The read of the file into memory may be resource intensive and is unnecessary if the user is not allowed to see the file anyway.

**•** N/A

**•** The regular expression has a vulnerable backtracking clause inside (\w+\s?)\*$ which can be triggered to cause a Denial of Service by processing particular phrases. To fix the backtracking problem, backtracking is removed with the ?= portion of the expression which changes it to a lookahead and the \2 which prevents the backtracking. The modified example is:

**•** N/A