# CWE Detail – CWE-64

## Description

The product, when opening a file or directory, does not sufficiently handle when the file is a Windows shortcut (.LNK) whose target is outside of the intended control sphere. This could allow an attacker to cause the product to operate on unauthorized files.

## Extended Description

The shortcut (file with the .lnk extension) can permit an attacker to read/write a file that they originally did not have permissions to access.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2019-19793: network access control service executes program with high privileges and allows symlink to invoke another executable or perform DLL injection.

**•** CVE-2000-0342: Mail client allows remote attackers to bypass the user warning for executable attachments such as .exe, .com, and .bat by using a .lnk file that refers to the attachment, aka "Stealth Attachment."

**•** CVE-2001-1042: FTP server allows remote attackers to read arbitrary files and directories by uploading a .lnk (link) file that points to the target file.

**•** CVE-2001-1043: FTP server allows remote attackers to read arbitrary files and directories by uploading a .lnk (link) file that points to the target file.

**•** CVE-2005-0587: Browser allows remote malicious web sites to overwrite arbitrary files by tricking the user into downloading a .LNK (link) file twice, which overwrites the file that was referenced in the first .LNK file.

**•** CVE-2001-1386: ".LNK." - .LNK with trailing dot

**•** CVE-2003-1233: Rootkits can bypass file access restrictions to Windows kernel directories using NtCreateSymbolicLinkObject function to create symbolic link

## Modes of Introduction

**•** Operation: N/A

## Common Consequences

**•** Impact: Read Files or Directories, Modify Files or Directories — Notes:

## Potential Mitigations

**•** Architecture and Design: Follow the principle of least privilege when assigning access rights to entities in a software system. Denying access to a file can prevent an attacker from replacing that file with a link to a sensitive file. Ensure good compartmentalization in the system to provide protected areas that can be trusted. (Effectiveness: N/A)

## Applicable Platforms

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

## Notes

**•** Research Gap: Under-studied. Windows .LNK files are more "portable" than Unix symlinks and have been used in remote exploits. Some Windows API's will access LNK's as if they are regular files, so one would expect that they would be reported more frequently.