CAPEC-159: Redirect Access to Libraries (Version 3.9)

Archived: 2026-04-05 16:52:53 UTC

Description

An adversary exploits a weakness in the way an application searches for external libraries to manipulate the execution flow to point to an adversary supplied library or code base. This pattern of attack allows the adversary to compromise the application or server via the execution of unauthorized code. An application typically makes calls to functions that are a part of libraries external to the application. These libraries may be part of the operating system or they may be third party libraries. If an adversary can redirect an application's attempts to access these libraries to other libraries that the adversary supplies, the adversary will be able to force the targeted application to execute arbitrary code. This is especially dangerous if the targeted application has enhanced privileges. Access can be redirected through a number of techniques, including the use of symbolic links, search path modification, and relative path manipulation.

Likelihood Of Attack

Typical Severity

Relationships

This table shows the other attack patterns and high level categories that are related to this attack pattern. These relationships are defined as ChildOf and ParentOf, and give insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as CanFollow, PeerOf, and CanAlsoBe are defined to show similar attack patterns that the user may want to explore.

This table shows the views that this attack pattern belongs to and top level categories within that view.

Execution Flow

Explore

1. **Identify Target:** The adversary identifies the target application and determines what libraries are being used.

Techniques
Find public source code and identify library dependencies.
Gain access to the system hosting the application and look for libraries in common locations.

Experiment

1. **Deploy Malicious Libraries:** The adversary crafts malicious libraries and deploys them on the system where the application is running, or in a remote location that can be loaded by the application.

Exploit

1. **Redirect Library Calls to Malicious Library:** Once the malicious library crafted by the adversary is deployed, the adversary will manipulate the flow of the application such that it calls the malicious library. This can be done in a variety of ways based on how the application is loading and calling libraries.

Techniques
Poison the DNS cache of the system so that it loads a malicious library from a remote location hosted by the adversary instead of the legitimate location
Create a symlink that tricks the application into thinking that a malicious library is the legitimate library.
Use DLL side-loading to place a malicious verison of a DLL in the windows directory.

Prerequisites

The target must utilize external libraries and must fail to verify the integrity of these libraries before using them.

Skills Required

[Level: Low] To modify the entries in the configuration file pointing to malicious libraries
[Level: Medium] To force symlink and timing issues for redirecting access to libraries
[Level: High] To reverse engineering the libraries and inject malicious code into the libraries

Consequences

This table specifies different individual consequences associated with the attack pattern. The Scope identifies the security property that is violated, while the Impact describes the negative technical impact that arises if an adversary succeeds in their attack. The Likelihood provides information about how likely the specific consequence is expected to be seen relative to the other consequences in the list. For example, there may be high likelihood that a pattern will be used to achieve a certain impact, but a low likelihood that it will be exploited to achieve a different impact.

Scope	Impact	Likelihood
Authorization	Execute Unauthorized Commands	
Access Control Authorization	Bypass Protection Mechanism	

Mitigations

Implementation: Restrict the permission to modify the entries in the configuration file.
Implementation: Check the integrity of the dynamically linked libraries before use them.
Implementation: Use obfuscation and other techniques to prevent reverse engineering the libraries.

Example Instances

In this example, the attacker using ELF infection that redirects the Procedure Linkage Table (PLT) of an executable allowing redirection to be resident outside of the infected executable. The algorithm at the entry point code is as follows... • mark the text segment writeable • save the PLT(GOT) entry • replace the PLT(GOT) entry with the address of the new lib call The algorithm in the new library call is as follows... • do the payload of the new lib call • restore the original PLT(GOT) entry • call the lib call • save the PLT(GOT) entry again (if its changed) • replace the PLT(GOT) entry with the address of the new lib call

Taxonomy Mappings

CAPEC mappings to ATT&CK techniques leverage an inheritance model to streamline and minimize direct CAPEC/ATT&CK mappings. Inheritance of a mapping is indicated by text stating that the parent CAPEC has relevant ATT&CK mappings. Note that the ATT&CK Enterprise Framework does not use an inheritance model as part of the mapping to CAPEC.

Relevant to the ATT&CK taxonomy mapping

Entry ID	Entry Name
1574.008	Hijack Execution Flow:Path Interception by Search Order Hijacking

References

Content History

Submissions

Submission Date	Submitter	Organization
2014-06-23 (Version 2.6)	CAPEC Content Team	The MITRE Corporation
Modifications		
Modification Date	**Modifier**	**Organization**
2015-11-09 (Version 2.7)	CAPEC Content Team	The MITRE Corporation
	Updated References	
2018-07-31 (Version 2.12)	CAPEC Content Team	The MITRE Corporation
	Updated Attack_Phases, Description, Description Summary, References, Related_Weaknesses	
2019-04-04 (Version 3.1)	CAPEC Content Team	The MITRE Corporation
	Updated Related_Attack_Patterns, Related_Weaknesses	
2020-07-30 (Version 3.3)	CAPEC Content Team	The MITRE Corporation
	Updated Taxonomy_Mappings	
2020-12-17 (Version 3.4)	CAPEC Content Team	The MITRE Corporation
	Updated References	
2021-06-24 (Version 3.5)	CAPEC Content Team	The MITRE Corporation
	Updated Related_Attack_Patterns, Taxonomy_Mappings	
2022-02-22 (Version 3.7)	CAPEC Content Team	The MITRE Corporation
	Updated Execution_Flow	
2022-09-29 (Version 3.8)	CAPEC Content Team	The MITRE Corporation
	Updated Related_Attack_Patterns	

More information is available — Please select a different filter.
