CAPEC-13: Subverting Environment Variable Values (Version 3.9)

Archived: 2026-04-05 23:38:23 UTC

Attack Pattern ID: 13 **Abstraction: Detailed**		

Description

The adversary directly or indirectly modifies environment variables used by or controlling the target software. The adversary's goal is to cause the target software to deviate from its expected operation in a manner that benefits the adversary.

Likelihood Of Attack

High

Typical Severity

Very High

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.

Nature	Type
ChildOf	S Standard Attack Pattern - A standard level attack pattern in CAPEC is focused on a specific methodology or technique used in an attac
PeerOf	D Detailed Attack Pattern - A detailed level attack pattern in CAPEC provides a low level of detail, typically leveraging a specific techni
CanPrecede	D Detailed Attack Pattern - A detailed level attack pattern in CAPEC provides a low level of detail, typically leveraging a specific techni

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

View Name	Top Level Categories
Domains of Attack	Software
Mechanisms of Attack	Subvert Access Control

Execution Flow

Explore

1. **Probe target application:** The adversary first probes the target application to determine important information about the target. This information could include types software used, software versions, what user input the application consumes, and so on. Most importantly, the adversary tries to determine what environment variables might be used by the underlying software, or even the application itself.

Experiment

1. **Find user-controlled environment variables:** Using the information found by probing the application, the adversary attempts to manipulate any user-controlled environment variables they have found are being used by the application,

or suspect are being used by the application, and observe the effects of these changes. If the adversary notices any significant changes to the application, they will know that a certain environment variable is important to the application behavior and indicates a possible attack vector.

Techniques
Alter known environment variables such as "$PATH", "$HOSTNAME", or "LD_LIBRARY_PATH" and see if application behavior changes.

Exploit

1. **Manipulate user-controlled environment variables:** The adversary manipulates the found environment variable(s) to abuse the normal flow of processes or to gain access to privileged resources.

Prerequisites

An environment variable is accessible to the user.
An environment variable used by the application can be tainted with user supplied data.
Input data used in an environment variable is not validated properly.
The variables encapsulation is not done properly. For instance setting a variable as public in a class makes it visible and an adversary may attempt to manipulate that variable.

Skills Required

[Level: Low] In a web based scenario, the client controls the data that it submitted to the server. So anybody can try to send malicious data and try to bypass the authentication mechanism.
[Level: High] Some more advanced attacks may require knowledge about protocols and probing technique which help controlling a variable. The malicious user may try to understand the authentication mechanism in order to defeat it.

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
Confidentiality Integrity Availability	Execute Unauthorized Commands	
Confidentiality Access Control Authorization	Bypass Protection Mechanism	
Availability	Unreliable Execution	
Confidentiality	Read Data	

Accountability	Hide Activities	

Mitigations

Protect environment variables against unauthorized read and write access.
Protect the configuration files which contain environment variables against illegitimate read and write access.
Assume all input is malicious. Create an allowlist that defines all valid input to the software system based on the requirements specifications. Input that does not match against the allowlist should not be permitted to enter into the system.
Apply the least privilege principles. If a process has no legitimate reason to read an environment variable do not give that privilege.

Example Instances

Changing the LD_LIBRARY_PATH environment variable in TELNET will cause TELNET to use an alternate (possibly Trojan) version of a function library. The Trojan library must be accessible using the target file system and should include Trojan code that will allow the user to log in with a bad password. This requires that the adversary upload the Trojan library to a specific location on the target. As an alternative to uploading a Trojan file, some file systems support file paths that include remote addresses, such as \\172.16.2.100\shared_files\trojan_dll.dll. See also: Path Manipulation (CVE-1999-0073)
The HISTCONTROL environment variable keeps track of what should be saved by the history command and eventually into the ~/.bash_history file when a user logs out. This setting can be configured to ignore commands that start with a space by simply setting it to "ignorespace". HISTCONTROL can also be set to ignore duplicate commands by setting it to "ignoredups". In some Linux systems, this is set by default to "ignoreboth" which covers both of the previous examples. This means that " ls" will not be saved, but "ls" would be saved by history. HISTCONTROL does not exist by default on macOS, but can be set by the user and will be respected. Adversaries can use this to operate without leaving traces by simply prepending a space to all of their terminal commands.

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
1562.003	Impair Defenses:Impair Command History Logging
1574.006	Hijack Execution Flow:Dynamic Linker Hijacking
1574.007	Hijack Execution Flow:Path Interception by PATH Environment Variable

References

[REF-1] G. Hoglund and G. McGraw. "Exploiting Software: How to Break Code". Addison-Wesley. 2004-02.

Content History

Submissions		
Submission Date	**Submitter**	**Organization**
2014-06-23 (Version 2.6)	CAPEC Content Team	The MITRE Corporation
Modifications		
Modification Date	**Modifier**	**Organization**
2017-01-09	CAPEC Content Team	The MITRE Corporation

(Version 2.9)	Updated Related_Attack_Patterns	
2018-07-31 (Version 2.12)	CAPEC Content Team	The MITRE Corporation
	Updated Attacker_Skills_or_Knowledge_Required, Examples-Instances, References	
2019-09-30 (Version 3.2)	CAPEC Content Team	The MITRE Corporation
	Updated Example_Instances	
2020-07-30 (Version 3.3)	CAPEC Content Team	The MITRE Corporation
	Updated Consequences, Mitigations, Taxonomy_Mappings	
2020-12-17 (Version 3.4)	CAPEC Content Team	The MITRE Corporation
	Updated Taxonomy_Mappings	
2021-06-24 (Version 3.5)	CAPEC Content Team	The MITRE Corporation
	Updated Taxonomy_Mappings	
2022-02-22 (Version 3.7)	CAPEC Content Team	The MITRE Corporation
	Updated Description, Example_Instances, Execution_Flow, Prerequisites	

More information is available — Please select a different filter.
