INDUSFACE™

OWASP TOP 10

2022 Playbook

Why Should You Read This Guide?

Website vulnerabilities have become a security nightmare for most businesses. Whether you're an *entrepreneur, an IT manager, an established business owner, a CIO, a director of security, a CTO, or something in between,* understanding and evaluating risks is critical. And that's exactly where OWASP 10 can help you.

The Open Web Application Security Project (OWASP) is a worldwide not-for-profit organization focused on improving the security of software. Their top-10 list is a broad consensus of the most critical web application security flaws. OWASP released the latest version of this list recently after a four-year gap, this playbook will serve as *a practical guide* to decoding OWASP-10 and preparing a response plan to counter these vulnerabilities.

This guide will also talk about how you can use AppTrana readily in your preparation to counter these vulnerabilities.

Inside This Playbook

List of Contents:

Introduction

Web applications play a critical role in empowering businesses and customers across the world. Yet somehow, despite the exponential digital growth, most businesses rarely consider the security loopholes in their web application.

In the fight against cybercriminals, OWASP's Top 10 Vulnerabilities serve as an ideal place to start covering the bases. In this playbook, we aim to break these vulnerabilities down and help you understand what they are all about.

We will also spell out exactly how AppTrana can be used to protect your sites against OWASP Top 10 vulnerabilities.

About AppTrana

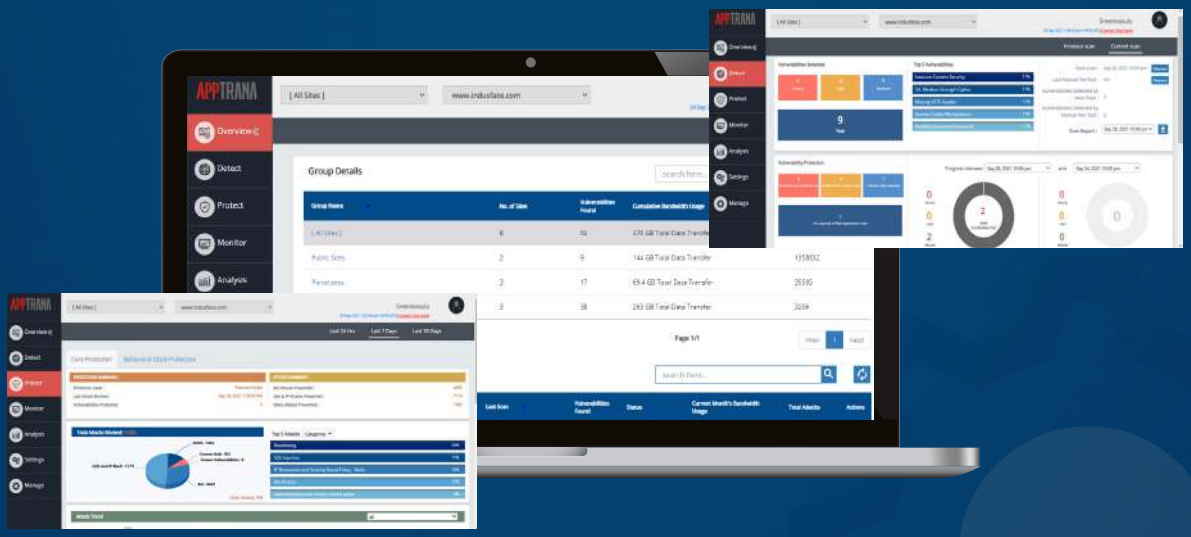
AppTrana is a complete security as a service solution that helps you identify vulnerabilities in your application and protect against them immediately through virtual patching at the WAF layer.

General Cloud security solutions, take a cookie-cutter approach when it comes to WAF. They provide a default set of rules that can be applied and then ask website owners to create/tweak the rules to meet their application needs. The problem with this approach is:

- With just default rules and no knowledge of the vulnerability, the security is weak
- Default rules will create false positives and tweaking it is time-consuming
- Creating your own custom rules is complex and requires expertise

Due to these reasons over time, the WAF solution becomes ineffective as organizations do not have the time and expertise to maintain the rules.

We at AppTrana approach the problem differently. We believe the security of the application is best handled by experts and our experts fine-tune the rules based on the application need to avoid false positives and ensure that your application remains secure round the clock. Our integrated solution of scanner and WAF helps you detect the vulnerabilities in your site and protect them immediately through custom made rules which assure zero false positives.



About OWASP

Open Web Application Security Project (OWASP) is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software. Its mission is to make software security a priority for individuals and organizations to take informed decisions. Operating as a community of like-minded professionals, OWASP issues software tools and knowledge-based documentation on application security. They publish a ranking of the 10 most-critical web application security flaws, which are known as the OWASP Top 10.

The OWASP Top 10 represents a broad consensus of the most critical web application security flaws. It's a widely accepted methodology for evaluating web application security and building mitigation strategies for websites and web-based applications. It outlines the top 10 areas where web applications are susceptible to attacks, and where common vulnerabilities are found in such workloads.

Though the coverage of AppTrana is well beyond OWASP's top 10, the scope of this document is to guide you through OWASP's top 10 vulnerabilities and help you understand how AppTrana can be of help. Let's Start!

OWASP TOP 10 Vulnerabilities

According to the latest release, the top vulnerabilities that organizations should be worried about are:

- A01:2021 - Broken Access Control
- A02:2021 - Cryptographic Failures
- A03:2021 - Injection
- A04:2021 - Insecure Design
- A05:2021 - Security Misconfiguration
- A06:2021 - Vulnerable and Outdated Components
- A07:2021 - Identification and Authentication Failures
- A08:2021 - Software and Data Integrity Failures
- A09:2021 - Security Logging and Monitoring Failures
- A10:2021 - Server-Side Request Forgery

With AppTrana, you will be able to detect if your application is vulnerable in any of these areas and ensure you are protected from any exploits immediately. Let's dig deeper and try to understand what each of these vulnerabilities means and what AppTrana can do for you.

A1-Broken Access Control

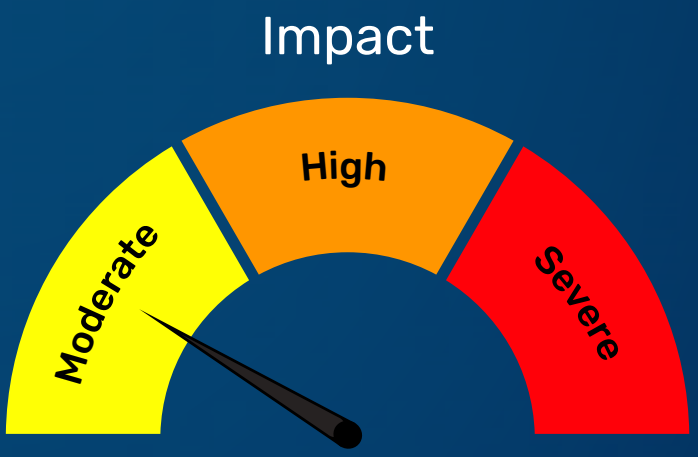
What is it?

Access control refers to the enforcement of restrictions on authenticated users to perform actions outside of their level of permission. Broken access control occurs when such restrictions are not correctly enforced. This can lead to unauthorized access to sensitive information, as well as its modification or destruction.

Here's how it is exploited. You are logged into an eCommerce portal and the URL shows the User ID like the illustration below.

Now if you change the digits '2340' to another set of digits that allows you to view the account of another user, it's a huge opportunity for hackers.

The threat also arises when non-privileged users have access to admin privileges. After all, a junior-level developer should not be able to gain admin access to the server. Unfortunately, most companies do not bother ensuring that only authorized accounts access privileged information.



What Are the Risks?

Such vulnerabilities lead to loss of data, ghost account creation, and admin account hijacking.

How to Prevent:

- Access control is only effective in trusted server-side code orserver-less API, where the attacker cannot modify the access control check or metadata.
- Except for public resources, deny by default.
- Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage.
- Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record.
- Unique application business limit requirements should be enforced by domain models.
- Disable webserver directory listing and ensure file metadata (e.g., git) and backup files are not present within web roots.
- Log access control failures, alert admins when appropriate (e.g., repeated failures).
- Rate limit API and controller access to minimize the harm from automated attack tooling.

How to Identify if Your Application is Vulnerable?

OWASP recommends the following checks to identify if you are vulnerable to Broken access control:

- Test Directory traversal/file include
- Test for Insecure Direct Object References
- Test for Local File Inclusion
- Test for Remote File Inclusion
- Test for Bypassing Authorization Schema
- Test for Bypassing Authentication Schema

How Can AppTrana Help?

Both the Automated scans and Premium scans of AppTrana cover this vulnerability and they can be readily blocked by Advanced, premium rules which AppTrana provides.

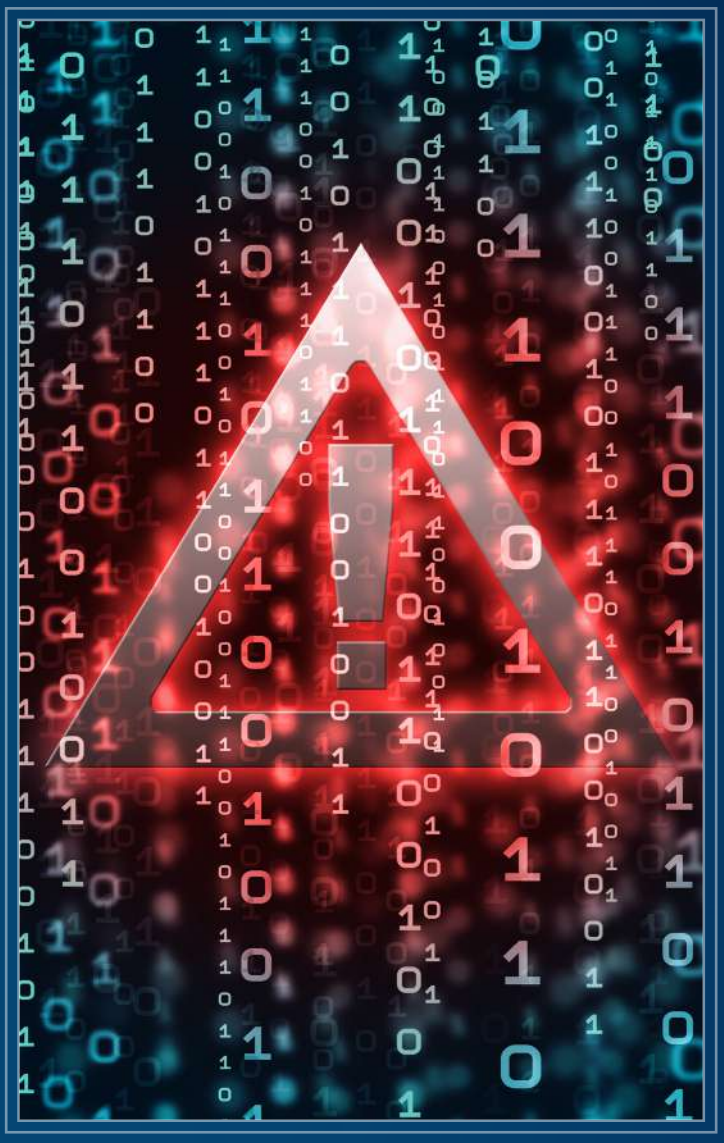
In case of application-specific vulnerability, which is not covered by the pre-written rules, you can request a custom rule which will be written by our experts.

The complete competency matrix of AppTrana is as follows

OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
	Test Directory traversal/file include	Yes	Yes	Yes	Yes	Yes
	Test for Insecure Direct Object References	Yes	Yes	Yes	Yes	Yes
	Test for Local File Inclusion	Yes	Yes	Yes	Yes	Yes
	Test for Remote File Inclusion	Yes		The authorization process can only be improved in the Application. WAF can be used to block the identity, preventing malicious users from gaining access to unauthorized resources		
	Test for Bypassing Authorization Schema	Yes				
	Test for Bypassing Authentication Schema	Yes				

A2-Cryptographic Failures

What is it?



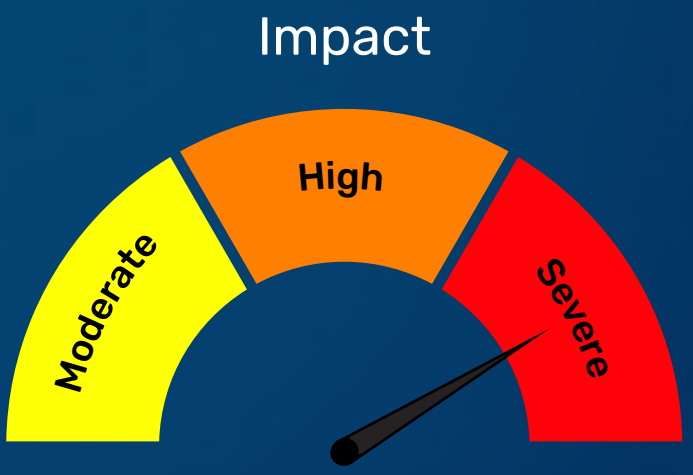
If somehow your application is breached, how easy is it for hackers to find the data that they want?

The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise. Consider database files, backups, financial transaction details, employment history, and every piece of internal and external information. Don't store sensitive data unnecessarily and discard it as soon as possible. If you have something, keep it encrypted.

For example - An application encrypts credit card numbers in a database using automatic database encryption. However, this data is automatically decrypted when retrieved, allowing a SQL injection flaw to retrieve credit card numbers in cleartext.

What Are the Risks?

Consider everything that comes with the loss of sensitive data. Loss of passwords, credit card information, addresses, and bank statements bring serious repercussions in real-world scenarios.



How to Prevent?

- Classify data processed, stored, or transmitted by an application. Identify which data is sensitive according to privacy laws, regulatory requirements, or business needs.
- Don't store sensitive data unnecessarily. Discard it as soon as possible or use PCI DSS compliant tokenization or even truncation. Data that is not retained cannot be stolen.
- Make sure to encrypt all sensitive data at rest.
- Ensure up-to-date and strong standard algorithms, protocols, and keys are in place; use proper key management.
- Encrypt all data in transit with secure protocols such as TLS with forward secrecy (FS) ciphers, cipher prioritization by the server, and security parameters. Enforce encryption using directives like HTTP Strict Transport Security (HSTS).
- Disable caching for a response that contains sensitive data.
- Apply required security controls as per the data classification.
- Do not use legacy protocols such as FTP and SMTP for transporting sensitive data.
- Store passwords using strong adaptive and salted hashing functions with a work factor (delay factor), such as Argon2, scrypt, bcrypt, or PBKDF2.
- Initialization vectors must be chosen appropriately for the mode of operation. For many modes, this means using a CSPRNG (cryptographically secure pseudo-random number generator). For modes that require a nonce, then the initialization vector (IV) does not need a CSPRNG. In all cases, the IV should never be used twice for a fixed key.
- Always use authenticated encryption instead of just encryption.

How to Identify if Your Application is Vulnerable?

OWASP recommends the following checks:

- Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)
- Testing for Padding Oracle (OTG-CRYPST-002)
- Testing for Sensitive information sent via unencrypted channels (OTG-CRYPST-003)
- Test HTTP Strict Transport Security (OTG-CONFIG-007)
- Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)

How Can AppTrana Help?

AppTrana can be effectively used to detect the vulnerability. Both Automated and Premium scans have test cases that will help find the vulnerability but since the vulnerability is around strong encryption, it is outside the scope of WAF, and it is best to fix the vulnerability in the application itself.

The complete competency matrix of AppTrana is as follows:

OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
	Test for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection	Yes	Yes	Stronger Encryption can be enforced only at the Application level. Out of scope of WAF. WAF can be used to block users based on identity, so that malicious users do not take advantage of any vulnerability		
	Test for Padding Oracle	Yes	Yes			
	Test for Sensitive information sent via unencrypted channels	Yes	Yes			
	Test HTTP Strict Transport Security	Yes				
	Test for Credentials Transported over an Encrypted Channel	Yes	Yes			

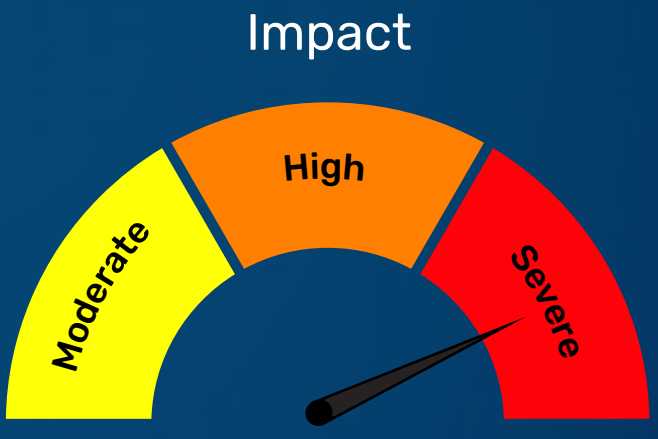
A3-Injection

What is it?

Imagine a hacker coming to your account login page and entering a string of code and that code or the command gets accepted and allows them to log in without even a valid account or password. *This is an injection attack.*

Websites that do not filter inputs and accept everything a user enters are at huge risk. Hackers can use commands to enter through this weakness and access server files. Injection attacks can happen from any input field, including the comments section.

What Are the Risks?



Think of the infinite possibilities when hackers have a direct way of interacting with your server. They can steal data, change it, delete it, deny access, and much more. In fact, injection attacks are responsible for some of the major data breaches last year.

How to Prevent?

- The preferred option is to use a safe API, which avoids using the interpreter entirely, pr vides a parameterized interface, or migrates to Object Relational Mapping Tools (ORMs). Note: Even when parameterized, stored procedures can still introduce SQL injection if PL/SQL or T-SQL concatenates queries and data or executes hostile data with EXECUTE IMMEDIATE or exec().
- Use positive server-side input validation. This is not complete security as many applications require special characters, such as text areas or APIs for mobile applications.
- For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter. Note: SQL structures such as table names, column names, and so on cannot be escaped, and thus user-supplied structure names are dangerous. This is a common issue in report-writing software.
- Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection.

How to Identify if Your Application is Vulnerable?

OWASP recommends the following checks to identify if you are vulnerable to Injection attacks:

- Test for SQL Injection
- Test for LDAP Injection
- Test for ORM Injection
- Test for XML Injection
- Test for SSI Injection
- Test for XPath Injection
- Test for IMAP/SMTP Injection
- Testing for Code Injection
- Testing for Command Injection
- Testing for Buffer Overflow

How Can AppTrana Help?

AppTrana with its advance and premium scans can help you detect all the vulnerabilities. Premium scans are only available as part of our premium offering where our experts do manual grey box pen-testing to identify vulnerabilities in your site. Note that buffer overflow checks are only done on request after customer approval.

When it comes to protection AppTrana takes a multi-prong approach. It provides a certain set of advanced rules created by experts that we feel is error-free and won't have any false positives in most of the workloads.

Apart from it, there are rules which are experiential in nature and may cause false positives, these rules will be monitored by our experts and moved to block mode after fine-tuning. This is available only as part of the premium plan.

Also, our experts can create custom rules on request based on the application when any specific vulnerabilities are found which is not protected by the prewritten rules.

The complete competency matrix of AppTrana when it comes to Injection is as follows:

OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
A3 Injection						
	Test for SQL Injection	Yes	Yes	Yes	Yes	Yes
	Test for LDAP Injection	Yes	Yes			Yes
	Test for ORM Injection	Yes		Yes	Yes	Yes
	Test for XML Injection	Yes				Yes
	Test for SSI Injection	Yes		Yes	Yes	Yes
	Test for XPath Injection	Yes	Yes	Yes	Yes	Yes
	Test for IMAP/SMTP Injection	Yes				Yes
	Test for Code Injection	Yes	Yes	Yes	Yes	Yes
	Test for Command Injection	Yes	Yes	Yes	Yes	Yes
	Test for Buffer Overflow	On Request				Yes

A4-Insecure Design (New)

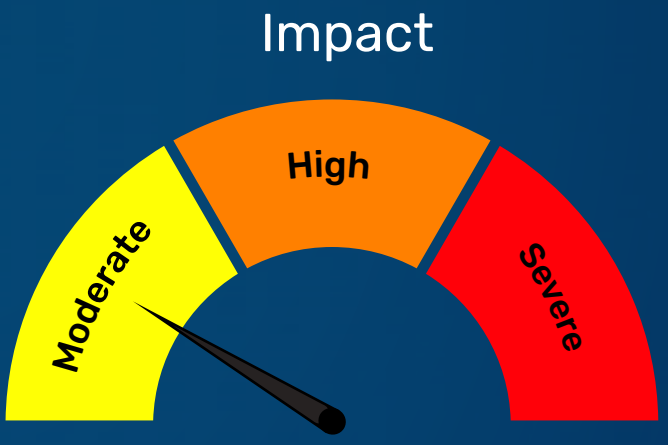
What is it?

A new category for 2021 focuses on risks related to design and architectural flaws, with a call for more use of threat modeling, secure design patterns, and reference architectures. Insecure design refers, in part, to the lack of security controls and business risk profiling in the development of software, and thereby the lack of proper determination of the degree of security design that is needed.

What Are the Risks?

Insecure application design can have severe consequences for the business, as it may allow attackers to interfere with the application logic and lead to sensitive information disclosure or web-application compromise.

How to Prevent?



- Establish and use a secure development lifecycle with AppSec professionals to help evaluate and design security and privacy-related controls
- Establish and use a library of secure design patterns or paved road ready to use components
- Use threat modeling for critical authentication, access control, business logic, and key flows
- Integrate security language and controls into user stories
- Integrate plausibility checks at each tier of your application (from frontend to backend)
- Write unit and integration tests to validate that all critical flows are resistant to the threat model. Compile use-cases and misuse-cases for each tier of your application.
- Segregate tier layers on the system and network layers depending on the exposure and protection needs
- Segregate tenants robustly by design throughout all tiers
- Limit resource consumption by user or service

How to Identify if Your Application is Vulnerable?

- Testing unsafe APIs
- OWASP Cheat Sheet: Secure Design Principles
- Testing usage of CORS (Cross-Origin Resources)
- Testing for Insecure Direct Object References
- Testing Missing user input

How Can AppTrana Help?

The flaw exists in the application itself in the design so it should be fixed on the design level itself. The scanner can be used to detect a few of the vulnerabilities caused by this like Local File Inclusion, Remote File Inclusion, CORS, OS Command Injection, Sensitive information disclosure, etc. Some of them can be detected by the WAF side but it is outside the scope of WAF and it is best to fix the flaws in the application itself.

A5-Security Misconfiguration

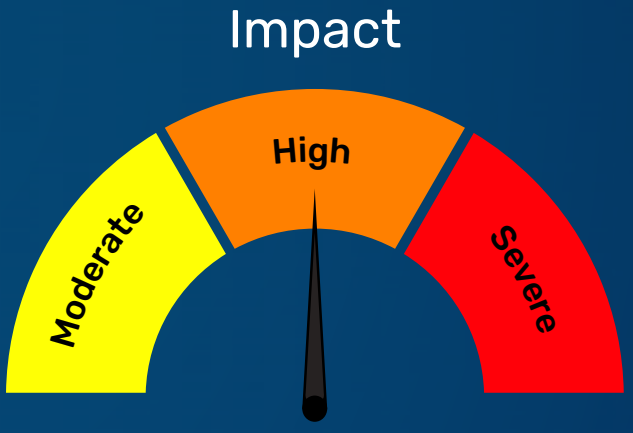
What is it?

Old sample apps, expired yet active features, default system passwords, etc. Hackers love all the additional information they can get. This vulnerability is about all these loopholes.

Attackers look for small issues, combine them, and try to make something big out of them. They simply try to misconfigure things which leads to an attack. They use default accounts, unused pages, unpatched flaws, unprotected files and directories, etc. to gain unauthorized access to or knowledge of the system. Mostly 75% of vulnerabilities take place in the cloud due to security misconfiguration.

What Are the Risks?

A5 can lead to complete loss of data through alteration, deletion and theft. Recovery is costly and highly unreliable, especially if the data gets encrypted.



How to Prevent?

- A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.
- A task to review and update the configurations appropriate to all security notes, updates, and patches as part of the patch management process (see A06:2021-Vulnerable and Outdated Components). Review cloud storage permissions (e.g., S3 bucket permissions).
- A segmented application architecture provides effective and secure separation between components or tenants, with segmentation, containerization, or cloud security groups (ACLs).
- Sending security directives to clients, e.g., Security Headers.
- An automated process to verify the effectiveness of the configurations and settings in all environments.

How to Identify if Your Application is Vulnerable?

OWASP recommends the following checks to identify if you are vulnerable to Security Misconfiguration:

- Fingerprint Web Server
- Fingerprint Web Application Framework
- Fingerprint Web Application
- Test Network/Infrastructure Configuration
- Test Application Platform Configuration
- Test File Extensions Handling for Sensitive Information
- Review Old, Backup and Unreferenced Files for Sensitive Information
- Enumerate Infrastructure and Application Admin Interfaces
- Test HTTP Methods
- Test RIA cross domain policy
- Testing for Error Code
- Testing for Stack Traces

How Can AppTrana Help?

Since these vulnerabilities are regarding server configurations and at the infra level, WAF can be used for mitigation purposes in most cases. Custom rules can be written to avoid some issues like banner grabbing to find server info and the like, but in most cases the patching must be done only at application. If the pattern of exploit is known based on the vulnerability, then custom rules can be written to block them.

AppTrana's premium scan and automated scan will find most of this type of vulnerability

The complete competency matrix of AppTrana is as follows:

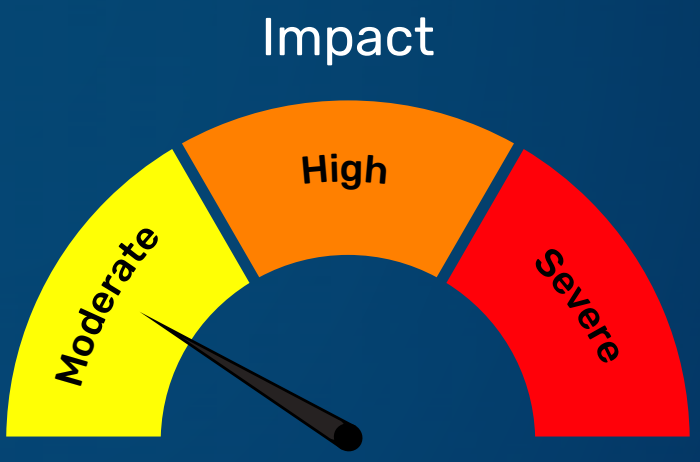
OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
A3 Injection						
	Fingerprint Web Server	Yes	Yes			Partially*
	Fingerprint Web Application Framework	Yes	Yes			Partially*
	Fingerprint Web Application	Yes	Yes			Partially*
	Test Network/Infra-structure Configuration	No		Can be fixed only at the Infra level. Out of Scope of WAF		
	Test Application Platform Configuration	No		Can be fixed only at the Infra level. Out of Scope of WAF		
	Test File Extensions Handling for Sensitive Information	Yes				Partially*
	Review Old, Backup and Unreferenced Files for Sensitive Information	Yes				Partially*
	Enumerate Infrastruc-ture and Application Admin Interfaces	Yes	Yes			Partially*
	Test HTTP Methods	Yes	Yes	Yes	Yes	Yes
	Test RIA cross-domain policy	Yes	Yes	NA		Partially*
	Test for Error Code	Yes	Yes			Partially*
	Test for Stack Traces	Yes	Yes			Partially*

A6-Vulnerable and Outdated Components

What is it?

This category was previously known as 'Using Components with Known Vulnerabilities' Most Web applications are developed using special frameworks that are provided by third parties? The 'Coding' world is filled with various open-source components and frameworks to build applications, which means there is a huge number of eyes looking at their source codes for any vulnerabilities. Unknown applications' codes may cause unlucky consequences and unwanted situations in the form of accent control breach, SQL injections etc.

What Are the Risks?



If the software, systems, etc. are not patched timely or if one is using vulnerable applications then it might lead to a critical attack. Which can lead to information disclosure, data leakage, RCE, etc.

How to Prevent?

- Remove unused dependencies, unnecessary features, components, files, and documentation.
- Only obtain components from official sources over secure links.
- Monitor for libraries and components that are unmaintained or do not create security patches for older versions. If patching is not possible, consider deploying a virtual patch to monitor, detect, or protect against the discovered issue.

How to Identify if Your Application is Vulnerable?

OWASP recommends the following checks to identify if you are vulnerable for

- Enumerate Applications on Webserver

How Can AppTrana Help?

Since it is a business logic vulnerability this could be identified readily only through AppTrana's premium scan.

Our security experts will be able to write customized rules to patch these vulnerabilities virtually depending on application needs in certain cases. Also, OpenVAS helps to identify such kinds of unpatched or vulnerable setup vulnerabilities.

OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
A6- Vulnerable and Outdated Components						
	Enumerate Applications on Webserver	Yes	Yes			Based on the specific vulnerability, we will provide the custom signature

A7- Identification and Authentication Failures

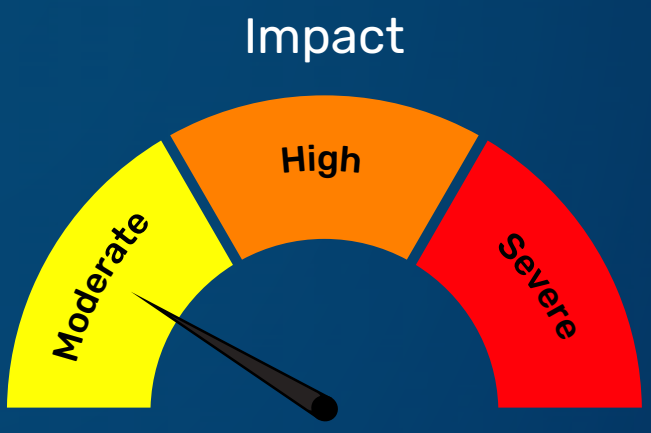
What is it?

Previously known as , this category slid down from the second position and now includes Common Weakness Enumerations (CWEs) related to identification failures. “Incorrect Password” error messages are a classic example of this type of vulnerability. If a hacker tries a random combination of User-name-Password and an error message tells them that the password is incorrect, he knows that at least the username is correct.

A brute force attacker now knows that an account is already present and he only needs the right password. There are dozens of automated tools from the dark web that can try millions of password combinations for a hacker. Improper sessions’ management is also a severe risk. Think of an online bank account that keeps you logged in even after closing the browser.

How to Prevent?

- Where possible, implement multi-factor authentication to prevent automated credential stuffing, brute force, and stolen credential reuse attacks.
- Do not ship or deploy with any default credentials, particularly for admin users.
- Implement weak password checks, such as testing new or changed passwords against the top 10,000 worst passwords list.
- Align password length, complexity, and rotation policies with the National Institute of Standards and Technology (NIST) 800-63b's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence-based password policies.



What Are the Risks?

Such vulnerabilities allow attackers to earn complete account access. In severe cases, hackers have stolen database records and sold them on the underground black market.

- Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes.
- Limit or increasingly delay failed login attempts but be careful not to create a denial-of-service scenario. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected.
- Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session identifier should not be in the URL, be securely stored, and invalidated after logout, idle, and absolute timeouts.

How to Identify if Your Application is Vulnerable?

OWASP recommends the following checks to identify if you are vulnerable to Identification and Authentication Failures:

- Test User Registration Process
- Test Account Provisioning Process
- Testing for Account Enumeration and Guessable User Account
- Testing for Weak or unenforced username policy
- Testing for Credentials Transported over an Encrypted Channel
- Testing for default credentials
- Testing for Weak lock out mechanism
- Testing for Bypassing Authentication Schema
- Testing for Vulnerable Remember Password
- Testing for Browser cache weakness
- Testing for Weak password policy
- Testing for Weak security question/answer
- Testing for weak password change or reset functionalities
- Testing for Weaker authentication in alternative channel
- Testing for Bypassing Authorization Schema
- Testing for Privilege escalation
- Testing for Session Management Schema
- Testing for cookies attributes
- Testing for Session Fixation

- Testing for Exposed Session Variables
- Testing for CSRF
- Testing for logout functionality
- Test Session Timeout
- Testing for Session puzzling

How Can AppTrana Help?

Since these vulnerabilities are application logic specific, automated scans have a limited scope, but all the vulnerabilities can be detected through AppTrana's premium scans.

Since session management and authentication vulnerability leads to unauthorized access, it becomes hard for the WAF layer to distinguish between valid request which is properly authenticated and requests which uses stolen identity. In such cases AppTrana takes a reactive approach and helps you block requests once you know that there has been a compromise. You could choose to blacklist certain identities or if you know the token, you can request a custom rule to block access for the token temporarily or permanently.

Apart from that some of the vulnerabilities can be completely or partially patched through custom rules. You can request the rule, if the scanner finds any of these vulnerabilities on your site.

The complete competency matrix of AppTrana when it comes to Broken Authentication is as follows:

OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
A7- Identification and Authentication Failures						
	Test Role Definitions	Yes		Proper role definition must be done on the application Out of the scope of WAF		
	Test User Registration Process	Yes		The registration process can only be improved in the aApplication. WAF can be used to block on identity. Preventing malicious users from registering multiple times		
	Test Account Provisioning Process	Yes		Process improvements are needed. Out of the scope of WAF		
	Test for Account Enumeration and Guessable User Account	Yes				Partially*
	Test for Weak or unenforced username policy	Yes				Yes
	Test for Credentials Transported over an Encrypted Channel	Yes	Yes	Encryption should be enforced at application. Out of scope of WAF		
	Test for default credentials	Yes		Strong password enforcement should happen at the code level. Out of scope of WAF		
	Test for Weak lock out mechanism	Yes				Yes
	Test for Bypassing Authentication Schema	Yes				
	Test for Vulnerable Remember Password	Yes		Strong authentication enforcement should happen at the code level. Out of scope of WAF		
	Test for Browser cache weakness	Yes				
	Test for Weak password policy	Yes				
	Test for Weak security question/answer	Yes				
	Test for weak password change or reset functionalities	Yes				
	Test for Weaker authentication in alternative channel	Yes				

	Test for Bypassing Authorization Schema	Yes				
	Test for Privilege escalation	Yes				Partially*
	Test for Session Management Schema	Yes				Partially*
	Test for cookies attributes	Yes	Yes			Yes
	Test for Session Fixation	Yes				Partially*
	Test for Exposed Session Variables	Yes	Yes			Partially*
	Test for logout functionality	Yes		Can be fixed only at the code level. Out of scope of WAF		
	Test Session Timeout	Yes				
	Test for Session puzzling	Yes				

A8-Software and Data Integrity Failures (New)

What is it?

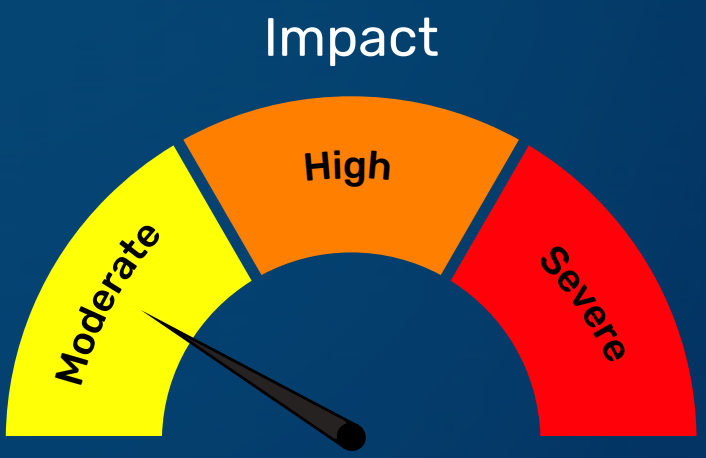
Software and data integrity failures relate to code and infrastructure that do not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs). An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code, or system compromise. Lastly, many applications now include auto-update functionality, where updates are downloaded without sufficient integrity verification and applied to the previously trusted application. Attackers could potentially upload their own updates to be distributed and run on all installations. Another example is where objects or data are encoded or serialized into a structure that an attacker can see and modify is vulnerable to insecure deserialization.

What Are the Risks?

Such vulnerabilities allow attackers to earn complete account access. In severe cases, hackers have stolen database records and sold them on the underground black market.

How to Prevent?

- Use digital signatures or similar mechanisms to verify the software or data is from the expected source and has not been altered.
- Ensure libraries and dependencies, such as npm or Maven, are consuming trusted repositories. If you have a higher risk profile, consider hosting an internal known-good repository that's vetted.



- Ensure that there is a review process for code and configuration changes to minimize the chance that malicious code or configuration could be introduced into your software pipeline.
- Ensure that unsigned or unencrypted serialized data is not sent to untrusted clients without some form of integrity check or digital signature to detect tampering or replay of the serialized data

How to Identify if Your Application is Vulnerable?

OWASP recommends the following checks to identify if you are vulnerable for https://www.owasp.org/index.php/Testing_for_CSRF_(OTG-SESS-005)

Test for Insecure Deserialization of User-supplied Data

How Can AppTrana Help?

Since these vulnerabilities are application logic specific, automated scans have limited scope, but all the vulnerabilities can be detected through AppTrana's premium scans, and they can be readily blocked by Advanced, premium rules which AppTrana provides.

Our security experts will be able to write customized rules to patch these vulnerabilities virtually depending on application needs in certain cases.

The complete competency matrix of AppTrana when it comes to covering one aspect in this category is as follows:

OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
A8- Software and Data Integrity Failures						
	Test for Insecure Deserialization of User-supplied Data	Yes	Yes	Yes	Yes	Yes

A9-Security Logging and Monitoring Failures

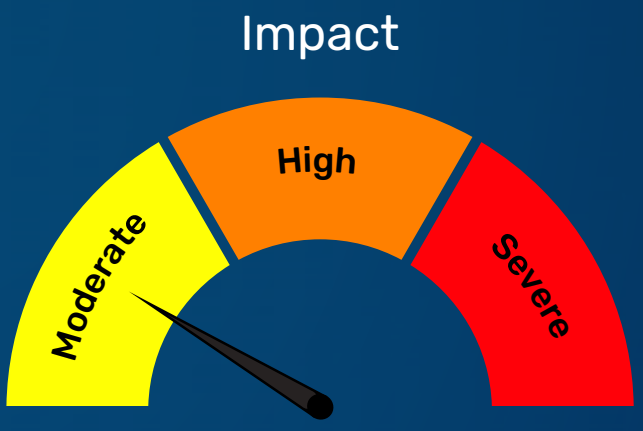
What is it?

Logging and monitoring can be challenging to test, often involving interviews or asking if attacks were detected during penetration testing? There isn't much CVE/CVSS data for this category but detecting and responding to breaches is critical. Still, it can be very impactful for accountability, visibility, incident alerting, and forensics.

What Are the Risks?

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

How to Prevent?



- Ensure that logs are generated in a format that log management solutions can easily consume.
- Ensure log data is encoded correctly to prevent injections or attacks on the logging or monitoring systems.
- Ensure high-value transactions have an audit trail with integrity controls to prevent tampering or deletion, such as append-only database tables or similar.
- DevSecOps teams should establish effective monitoring and alerting such that suspicious activities are detected and responded to quickly.

How to Identify if Your Application is Vulnerable?

The following checks are recommended.

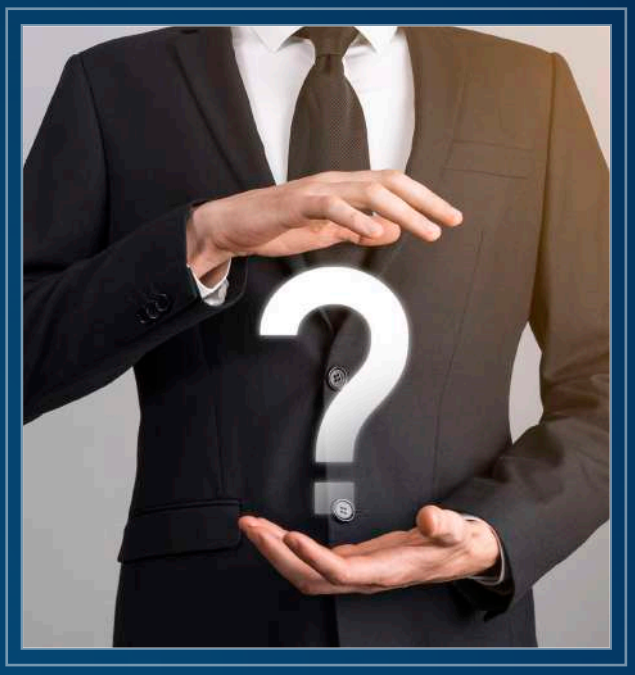
- OWASP Proactive Controls: Implement Logging and Intrusion Detection
- OWASP Application Security Verification Standard: V8 Logging and Monitoring
- OWASP Testing Guide: Testing for Detailed Error Code
- OWASP Cheat Sheet: Logging

How Can AppTrana Help?

This is not a specific vulnerability, and it does not apply to Apptrana. It refers to a lack of best practices that could hinder incident analysis & attack detection. This must be verified externally, manually during auditing, or reviewing backend/architecture.

A10- Server-Side Request Forgery (SSRF)(New)

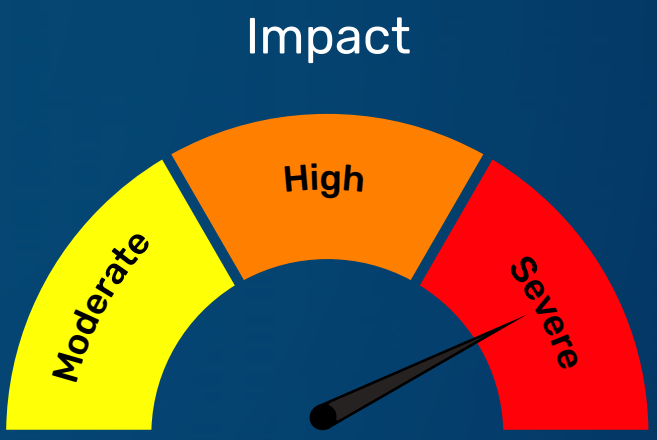
What is it?



SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

As modern web applications provide end-users with convenient features, fetching a URL becomes a common scenario. As a result, the incidence of SSRF is increasing. Also, the severity of SSRF is becoming higher due to cloud services and the complexity of architectures.

How to Prevent?



- Sanitize and validate all client-supplied input data
- Enforce the URL schema, port, and destination with a positive allow list
- Do not send raw responses to clients
- Disable HTTP redirections
- Be aware of the URL consistency to avoid attacks such as DNS rebinding.

How to Identify if Your Application is Vulnerable?

Testing for Server-Side Request Forgery

How Can AppTrana Help?

OWASP Top 10 Vulnerabilities (2021)	Tests Recommended by OWASP	Detection Coverage		Protection Coverage		
		Premium Scans	Automated Scans	Advance Rules	Premium Rules	Custom Rules
A10- Server-side request forgery						
	Testing for Server-Side Request Forgery	Yes	Yes			Yes

The scanner has coverage for SSRF detection, whereas protection is added through custom WAF rules for the WAF side. The detection also has coverage for out-of-band SSRF vulnerabilities.

About Indusface

Indusface is a leading application security SaaS company that secures critical Web, Mobile, and API applications of 3000+ global customers using its award-winning fully managed platform that integrates web application scanner, web application firewall, DDoS & BOT Mitigation, CDN, and threat intelligence engine.

Indusface has been funded by Tata Capital Growth Fund II, is the only vendor to be named Gartner Peer Insights™ Customers' Choice' in all the 7 segments for Web Application and API Protection Report 2022, is a "Great Place to Work" certified SaaS product company, is PCI, ISO27001, SOC 2, GDPR certified and has been the recipient of many prestigious start-up awards. such as the Economic Times Top 25, NASSCOM DSCI Top Security Company, Deloitte Asia Top 100, among others.

CONTACT US - +91 265 6133021 | +1 866 537 8234
EMAIL - sales@indusface.com

Authors

SigDev Team

Pavithra Hanchagaiah
Head of Security Research, Indusface

Product Team

Vivek Gopalan
Head of Product Management, Indusface

Saketh Rasakatla
Product Marketing Manager, Indusface

Indusface is the Only Vendor To Be Named Gartner® Peer Insights™ Customers' Choice in All the 7 Segments of Voice of Customer WAAP 2022 Report.