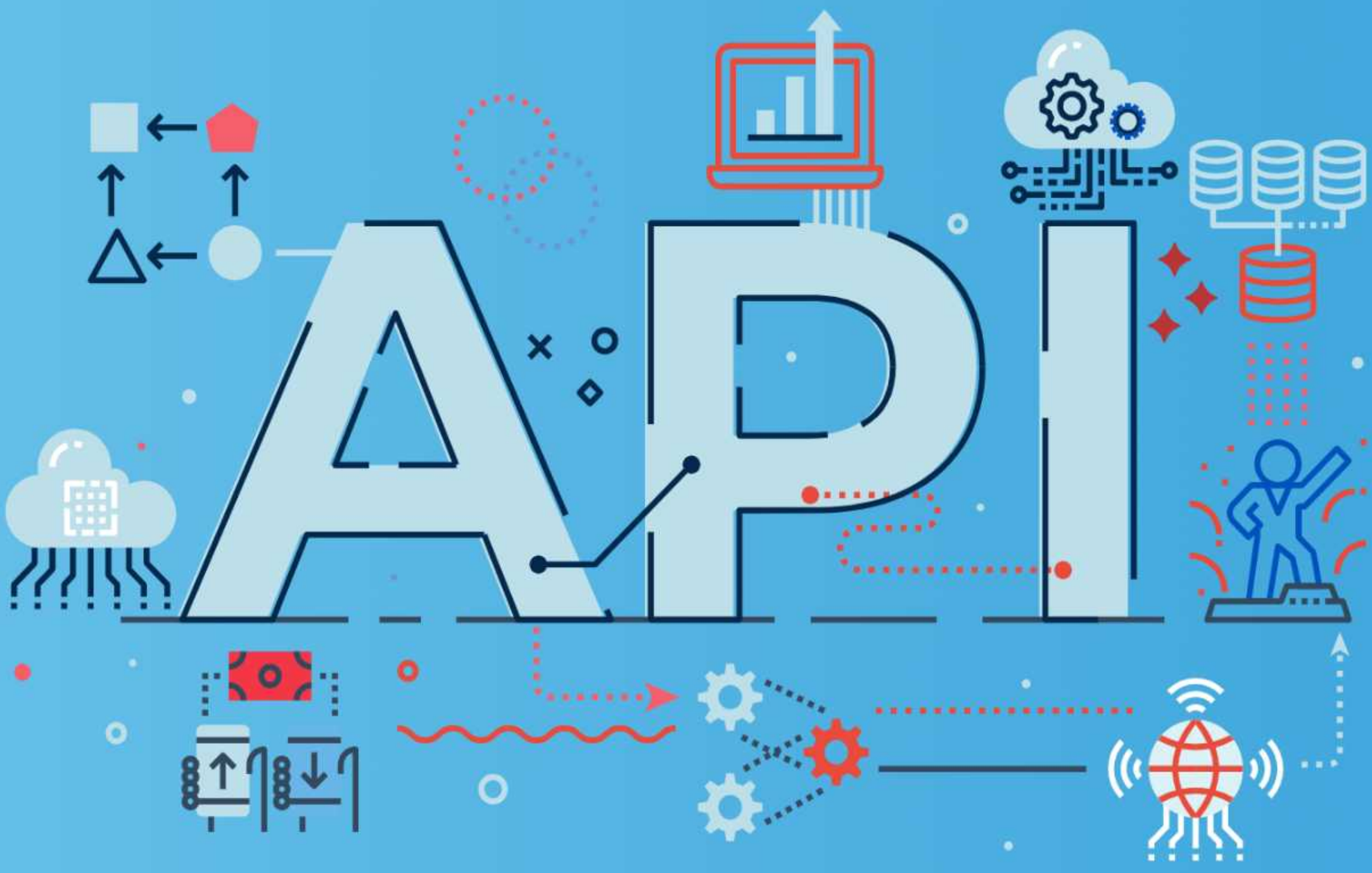
INDUSFACE™

WHITEPAPER

API Economy and Why Effective Security is Important?



API Economy

API has become the lifeline for digital businesses and has opened up new business opportunities. No longer is API just a technology enabler but it has grown into a business enabler, with various business models and partnerships built on the strength of APIs. This has led to the term **API economy.** It can be defined as how organizations leverage API to improve their efficiency and profitability by opening up new revenue opportunities.

Some of the major recent deals in this sector include Okta acquiring Auth0 for $6.5 billion acquisition, Salesforce spending $6.5 billion for Mulesoft, and Google acquiring Apigee for $625 million are testimonials of the importance of API and how it is driving the digital economy. All these deals were done to strengthen the acquirers' capabilities to support the API economy. Such deals would have not happened unless there are developers that need the right tool to create, deploy, control, monitor, analyse, and secure APIs.

In this fast-moving world where technologies are rapidly evolving, the businesses that strive are the ones that leverage APIs and integrate. The winner in the current digital world is "Integration" and not "Best in breed".

Security Risk – How attackers target APIs

According to Gartner's research, 90% of applications will have more of their attack surface area exposed in the form of APIs instead of UI. It predicts by 2022, API abuses will be the most frequent attack vector.

By nature, APIs expose application logic and sensitive data and due to this, they become a natural target for attackers.

Since APIs are meant to communicate between applications, the APIs need to be well defined in nature following a specific structure, and often, this definition is available in public as it is required by the developers to use those APIs. With this, attackers can easily understand how the API is written, and the underlying structure, and armed with this knowledge launch targeted attacks to exploit any vulnerabilities found in APIs.

These risks are not something perceived but are real and in the recent past, there is a significant jump in attacks targeting APIs. For example.

The personal data of Ledger's customer, a French cryptocurrency wallet company, was exposed not because of the security flaws in the wallet but due to a misconfigured third-party API. Even companies like YouTube, Tesla, and Twitter were not spared.

Based on how the YouTube API was written and how the authorization was handled, hackers were able to upload videos to different accounts that they did not own. Tesla's backup-gateway API also had authentication-related vulnerabilities exposing system-related

information and also certain personal information of the users. Twitter's Fleet API allowed access to the past-posts older than 24 hrs, which ideally should have been deleted from the public accounts leaving its owners in the dark on who is accessing their tweets.

With the rapid explosion of APIs and the huge exchange of information through APIs, every organisation should be concerned about how their APIs are being consumed and how they are consuming other APIs. How secure are these APIs and how well are they protected against attacks

OWASP Top 10 API Threat

In line with the changing trends and acknowledging the boom of the API economy, OWASP, the trusted open source community lead project geared towards strengthening the security of software, released the API-specific top 10 threats. OWASP Top 10 is known for highlighting the major and common flaws from the security point of view which is prevalent and commonly exploited by hackers. They generally used to focus on threats found in software but now they have released the top 10 threats focused on API.

Let's quickly skim through the Top 10 threats highlighted by OWASP.

API1:2019 Broken Object Level Authorization

This is an attack that exploits gaps in the authorization layer of APIs. For example, if a getAPI returns details of a user based on the ID passed, as part of the getAPI call, the authorization of the user who is accessing the API should also be checked and validated if the user has permission for the ID.

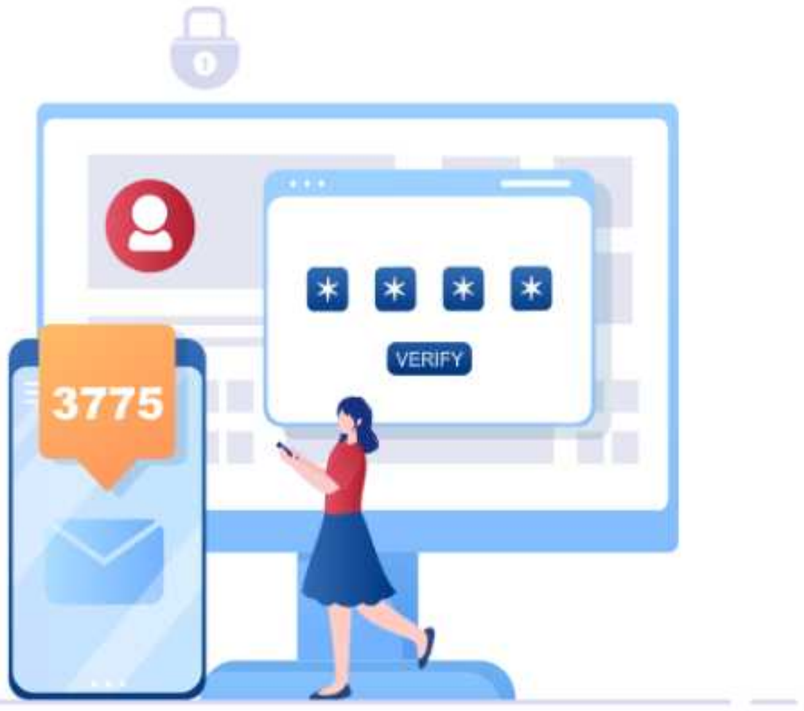
Such checks need to be done at each API call and are done by passing an authorization token as part of the API call and validation of the auth token. If such checks are not done properly then vulnerability can be exploited to gain access to data that the user does not have permission.

API2:2019 Broken User Authentication

This is a vulnerability that is introduced due to improper implementation of the authentication layer which allows attackers to exploit gaps to assume other user's identity temporarily or permanently. Such vulnerabilities are introduced because many times convivence is given more importance than security and either improper authentication mechanism like fixed API keys are used or proper validations of the token are not done.

API3:2019 Excessive Data Exposure

In many cases, developers do not consider the sensitivity and context before returning data for API calls. They end up exposing more information than needed which then can be exploited by hackers. For example for login API, as a response instead of saying success or failure, the response also contains additional information like device id, etc. Now this information can be captured by hackers to launch further exploits.



API4:2019 Lack of Resources & Rate Limiting

Normally there are no rate limits on the number of calls that a client can make, which is then exploited by hackers for denial of service attacks making the server not accessible for legitimate users. But such a lack of rate limits can also be used to exploit business logic. For example, if there is an OTP implemented in a login where OTP is sent to mobile which the user needs to enter but no rate limit is imposed on the POST API that gets the OTP from the user, then a hacker could launch a volumetric attack to guess OTP and bypass OTP.

API5:2019 Broken Function Level Authorization

This vulnerability is related to how the authorization is done on an API. For example, a customer's API can be used to both GET and Delete. So the authorization needs to be done at the operation level and not only at the API level, as the user may have access to the customer's API but only to access the data and not delete it.

API6:2019 Mass Assignment

Nowadays, many frameworks for ease of use, allow mass assignment where the entire input is assigned to a JSON object. For example, a GET user API may also return the permissions and if this is assigned to the object which is then used through HTML injection, the attacker could manipulate permissions and become a super-user. Such mass assignments should not be allowed, and selective assignments should be done to avoid such issues.

API7:2019 Security Misconfiguration

These kinds of vulnerabilities are introduced due to bad or unplanned configurations which lead to vulnerabilities that can be exploited by hackers. Common mistakes around such configurations are not disabling HTTP methods when HTTPS is only allowed in APIs, allowing unnecessary HTTP methods, permissive CORS, etc.

API8:2019 Injection

When inputs are provided to APIs and those are passed on by APIs without sanitizing, then it allows for such kinds of attacks. For example, say there is a search API to get the list of users from customer name, but instead of customer name, an SQL to delete is sent and API without sanitizing passes it on to DB, then the users get deleted and such an attack is called injection attacks.

API9:2019 Improper Assets Management

Containers/Kubernetes have enabled engineers to deploy rapidly. This opens up new doors for hackers as there are higher chances for older APIs to be deployed and forgotten, as new APIs are built, new containers are launched with no one worrying or thinking about the security of older APIs. Essentially it is important to track and document the available APIs and manage those assets.

API10:2019 Insufficient Logging & Monitoring

All actions performed by the system and APIs need to be logged, monitored, and analyzed for abnormalities. The authentication and authorization layers are not sufficient to detect all types of issues. A valid token used from Asia and within 5 mins from Australia is abnormal and needs to be detected. Such abnormalities can only be identified if all the actions are sufficiently logged with relevant details and are monitored continuously for abnormalities.

API Discovery

APIs can only be protected if one knows about them. One of the biggest challenges of API security is API visibility. Due to following reasons organisations have challenges regarding API visibility:

- **Shadow APIs:** These are APIs that are built as part of an application but are known only to some groups and are not publicized. Such APIs are built for the functioning of the application and are considered as an implementation detail, so, not many know about them. This also means the security team is not aware that such APIs exist and hence, no security is applied to them. The same can happen to an API thatch is known and where additional parameters are added by the developers but are not documented, so, that part is never part of the testing cycle.

- **Older version of APIs:** APIs are developed and improved. Various versions of APIs are released over time and to maintain continuity, the older versions of APIs are not discontinued. These APIs that remain available publicly are discovered with concentration being given to the latest versions of the API. But the older versions are still accessible that could be exploited by hackers

API Security with AppTrana

At Indusface, we understand APIs are becoming the lifeline and how important the security of APIs is for organisations. With its risk-based approach, Indusface's AppTrana is the only solution in the market that provides accurate protection that is tailor-made according to the API's security posture.

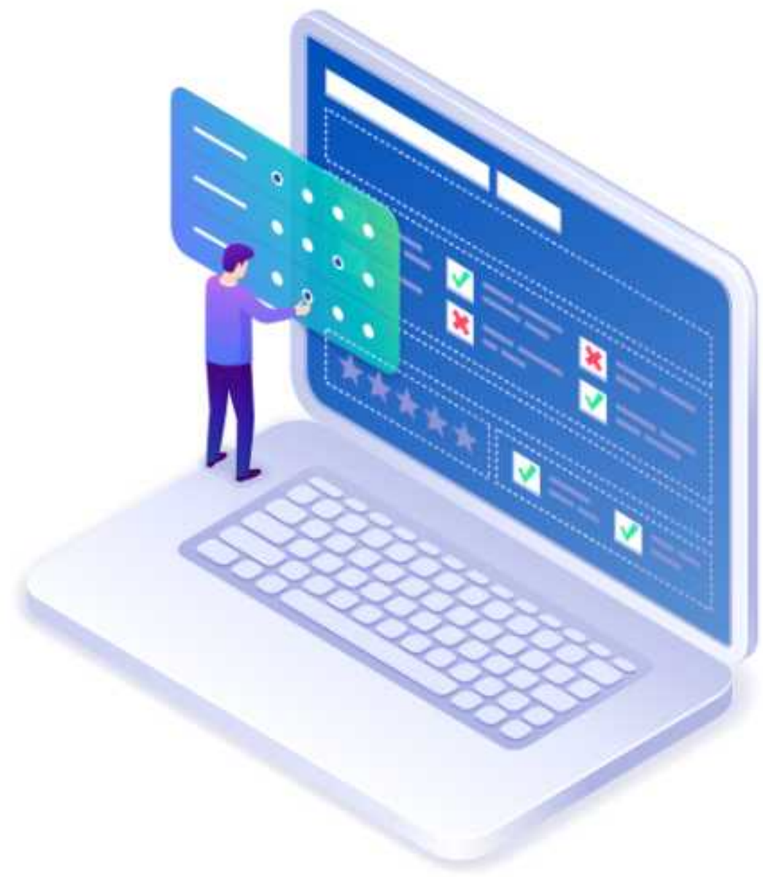
Risk Based Approach

By nature, APIs expose application logic and sensitive data, and due to this, they are an automatic target for hackers. For effective protection, it is imperative that one understands the hacker's mindset and their approach to prevent attacks. AppTrana with its risk-based approach enables customers to gauge the security posture of the application and helps detect and fix unidentified risks immediately.

Customers need to provide swagger/ postman files of their APIs and AppTrana scanner will take care of the rest by accurately identifying the risk posture of their APIs. Unlike other automated scanners, AppTrana also complements its automated scanner with manual pen-testing to ensure all vulnerabilities including OWASP top 10 vulnerabilities are identified accurately without any false positives. AppTrana also ensures the identification of business logic vulnerabilities which is the most common exploit in the case of APIs.

With the combination of automated scans and manual pen-testing, some of the vulnerabilities that AppTrana tests for in APIs include:

- Insecure Direct Object References
- Restricted Files can be viewed by Directory Listing
- Application is vulnerable to Cross-Origin Resource Sharing
- Sensitive Information Sent Over Unencrypted Channel
- Application is vulnerable to SQL injection attack
- Application is vulnerable to Directory Traversal attack
- Application is vulnerable to email flooding attack
- HTTP Request Smuggling attack
- Improper Token Management
- Application is vulnerable to OTP flooding attack
- Abuse of Send-Mail Functionality
- Abuse of Send-SMS Functionality
- Application is vulnerable to XML Injection attack
- OAuth/JWT/SAML Misconfiguration
- Valid account can be brute-forced
- XML-RPC is publicly available
- Application is vulnerable to Race Condition Attack
- Application is vulnerable to Remote Code Execution attack
- Application is vulnerable to a remote file inclusion attack
- Application is vulnerable to XML External Entity (XXE) Injection
- Application is vulnerable to Xpath Injection attack
- Insecure Deserialization
- ASP.NET tracing is enabled
- Application is vulnerable to SSRF attack
- Application's Apache Server - Status Enabled
- Application is vulnerable to Iframe injection attack
- Application is vulnerable to a Link injection attack
- Application is vulnerable to Log Injection attack
- Application is vulnerable to OS command injection attack
- Application is vulnerable to Price manipulation attack
- Application is vulnerable to Replay attack
- Application is vulnerable to SSI injection attack
- Application's OTP can be Bypassed
- ASP.NET debugging is enabled
- Authentication / 2FA bypass using response manipulation
- Cross-Site Web Socket Hijacking
- Local File Inclusion Attack – LFI
- Mobile No can be Bypassed and used to perform critical transactions
- Server-side Template Injection
- Application is vulnerable to HTTP Parameter pollution
- Application is vulnerable to URL Redirection attack



Visibility & Discovery of Shadow APIs

Protection is as effective as the weakest link and one can protect only what is known. The major challenge with the API ecosystem is that APIs evolve. Certain parameters to APIs are added without wider visibility, and newer versions of APIs are released without decommissioning older versions, resulting in a lot of undocumented APIs being available in public. These APIs are called Shadow APIs. AppTrana helps an organisation identify these Shadow APIs and ensure protection is applied to them

API Protection:

Once the APIs are bought under AppTrana protection, APIs are protected against attacks immediately. AppTrana is equipped with default policies that protect the APIs against the OWASP identified Top 10 API threats.

Virtual Patching for all vulnerabilities found:

Based on the risk posture identified through API scans, the API protection profile is further updated to ensure that vulnerabilities identified are patched ensuring there are no risk vectors that hackers can exploit. Patches can be automatic in nature due to the default API-specific policies applied on WAF or due to custom patches written by security experts based on need.identified are patched ensuring there are no risk vectors that hackers can exploit. Patches can be automatic in nature due to the default API-specific policies applied on WAF or due to custom patches written by security experts based on need.

Right Security Model Based on Need:

Based on the nature of the API definition provided and the kind of vulnerabilities found,

AppTrana uses a combination of positive & negative security models to provide effective protection. Since the protection is fine-tuned based on APIs needed after understanding how APIs are written, there are fewer chances of false positives, unlike the self-learning approach that alternative solutions take. Positive security policies are automatically created based on the definition provided ensuring the attack surface for APIs is drastically reduced.

Protection Against API Abuses:

AppTrana being a cloud-based WAAP provides a unified solution with integrated CDN, DDOS & BOT protection along with its WAF. This means that any API abuses are identified through its behaviour-based BOT/DDoS anomaly detection policies ensuring maximum protection.

Protection of Sensitive Data:

AppTrana provides a data loss prevention option, where policies can be enabled for APIs to ensure sensitive data is not sent through APIsThis avoids any accidental leakage of critical data.

API Security Key Capabilities

Comprehensive API Protection	
Features	**Benefits**
Risk Detection	
Managed API Scanning	Identify OWASP Top 10 API Vulnerabilities and others using Automated API Scans
Manual Pen-testing of APIs	Identify Business logic vulnerabilities through complementary manual pen-testing
Manual verification of vulnerabilities by experts	Security experts verify the vulnerabilities and remove false positives before results are published
Remediation Guidance to fix vulnerabilities	Get guidance on how to fix the vulnerabilities
Vulnerability Revalidation Checks	Rescan your APIs any number of times to ensure vulnerabilities identified are fixed
Discover Shadow APIs	Identify unknown, shadow APIs are that are part of the API Host configured behind AppTrana
Risk Protection	
Layer 7 Web Application Firewall	Get AppTrana to be in line with your API traffic and have it inspected traffic and allow only legit traffic to your site
API Specific Rules	Get immediate protection for your APIs with API specific policies
Zero-day vulnerability Protection	Get instantaneous protection for zero-day vulnerabilities through continuous updates written by security experts
Restrict by IP & Geo	Quickly block IP & Geo-based on traffic patterns
Ability to exempt certain API & IP through whitelisting	Whitelist API, to ensure that certain critical URI are not blocked accidentally
PCI DSS 3.2 Compliance	AppTrana is PCI Compliant and helps you to meet your compliance needs
Reduce attack surface through positive security policies	Get automatic positive security policies for your APIs ensuring the attack surface is reduced

DDOS Mitigation	
Protection of Layer 3, 4 Volumetric Attacks	Get instant protection against Layer 3 & 4 DDOS attacks
Behaviour Based Layer 7 DDOS Protection	Get behaviour based Layer 7 DDOS protection for your APIs
Protection of Origin IP	Ensure API Server is not compromised by allowing requests only from AppTrana
Behaviour Based DDOS Protection for APIs	Get granular API specific policies to provide behaviour based controls at the API level
BOT Mitigation	
Allow Good bots/ Bot Pretender Checks	Ensure Good bots are allowed but pretenders are blocked
Tor IP based detection	Identify traffic coming from Tor IPs and configure protection based on the need
Validation of bot signatures and blocking bad bots	Block bots based on their signature
Datacenter Based Detection	Identify traffic coming from data centers and configure protection based on the need
Scanner /Exploitable tools Checks	Block unknown scanners and exploitation tools
Suspicious Countries	Block suspicious countries
Behaviour-based detection	Identify bots based on behaviour patterns and block suspicious behaviour
Risk Monitoring	
24*7 management of WAF by certified application security experts	Get 24*7 monitoring of WAF done by security experts
Site Availability Notifications	Get various notifications around site availability and protection
Expert written custom rules	Get experts to write custom rules specific to the APIs ensuring tailored protection
Monitoring for False Positives in Rules	All API host is monitored for 14 days from onboarding to ensure there are no false positives. Rules are fine-tuned to avoid False positives
Adv DDOS Host-based Monitoring	Get experts to monitor for DDOS and alert in case of abnormal activities

API security is complex, and one needs a comprehensive solution that can tune its protection based on the API's needs.

Get the most comprehensive and risk-based API Protection for your organization with Risk detection, API Threat detection, API Positive Security policies, API-Specific DDoS policies, API-Specific Bot modules, and API Discovery with AppTrana's API Protection

Source: https://www.cloudvector.com/api-data-breaches-in-2020/

API Economy and Why Effective Security is Important?

Get Instant Protection

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.

Bengaluru | Vadodara | Mumbai | New Delhi | San Francisco

Contact: marketing@indusface.com