IBM Whitepaper

Addressing API threats as defined by OWASP with IBM API Connect

Priyanka Kohli
Product Manager – API Gateway Security

November 2021

Table of Contents

1. Introduction

The Open Web Application Security Project (OWASP) is a leading source of security information and recommendations for API's. Implementing their API Security Top 10 recommendations can instill confidence in the overall security of an API Solution. This whitepaper introduces and provides recommendations from IBM on how to help mitigate the unique threats and security risks of APIs, and GraphQL as in the OWASP API Security Top 10 list with the security capabilities built-in and integrated into the IBM API Management solutions.

The OWASP top 10 API security threats can be found at: https://owasp.org/www-project-api-security/

This whitepaper will also discuss how to mitigate the common attacks identified for GraphQL APIS from the following OWASP cheat sheet series: https://cheatsheetseries.owasp.org/cheatsheets/GraphQL_Cheat_Sheet.html

1.1 How does IBM offer API Management?

IBM API Connect provides a comprehensive set of API management capabilities to cover the entire lifecycle of an API. This includes the ability to create, manage, secure, and socialize APIs used by developers to expose core business assets and microservices that power modern digital applications across clouds and on-premises. API Security is only one of many important aspects of API Management and is the focus of this whitepaper.

IBM Cloud Pak for Integration provides an integration platform with a seamless experience across integration capabilities with Automation and AI to enable faster time to market and higher productivity. This includes App Integration, API Management, Messaging, Event Streams and more. IBM API Connect provides the API Management capabilities.

IBM is a market leader in the API Management market across analyst reports. See the references section for analysts and links to complimentary reports. [5] [6]

Please refer to this link for a detailed demo of the IBM API Management capabilities within API Connect V10 and IBM Cloud Pak for Integration: https://www.youtube.com/watch?v=HRU4aokd0Zs

What are the components related to API Security in IBM API Connect?

While there are many components of an API Management Solution, these two are most important to API Security.

- **API lifecycle manager:** Provides the ability to create APIs and API Products and manage the entire lifecycle, from initial creation through all development stages and inevitable retirement. During the creation of the API definition, Built-in and extensible policies are available to: Secure, control and mediate APIs at runtime.

- **API gateway:** Protects data and business assets by handling all routing requests, composition, policy enforcement (such as rate limiting and protocol translations) between clients and the services they're connected to. API gateways play a crucial role in ensuring the security of API connections by deploying key security authentication and enforcement protocols, including Transport Layer Security (TLS) 1.3 encryption and OAuth (Open Authorization) technology standards.

1.2 What is OWASP API Security Top 10 threats?

The Open Web Application Security Project is an open community that publishes a ranking of the most critical web application and APIs security flaws. They publish a top 10 list of what enterprises should be looking at when they’re applying the different types of threat protections. Since API Security is different in many ways than typical approach to web security,

the OWASP Top 10 Web Application Security Risks are also different from the OWASP API Security top 10.

For more information refer to the link below for the specific OWASP top 10 API security threats discussed: https://owasp.org/www-project-api-security/

1.3 What is API Security?

API security is the process of protecting APIs from attacks. Because APIs are very commonly used, and enable access to sensitive software functions and data, they are a primary target for attackers. Without secure APIs, rapid innovation would be impossible. API Security focuses on strategies and solutions to understand and mitigate the unique vulnerabilities and security risks of Application Programming Interfaces (APIs).

2. API Connect Addressing the OWASP Issues for APIs

2.1 API1:2019 Broken Object Level Authorization

Threat Description:
APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface Level Access Control issue. Object level authorization checks should be considered in every function that accesses a data source using an input from the user. *[1]*

Technical Relevance:
Broken Object level authorization is an access control vulnerability which occurs when a user supplied input is used to access other resources that they should not have access to regularly. In this type of vulnerability, an attacker can exploit the endpoints by manipulating the ID of the Object that is being sent in the request. It can allow unauthorized access to the sensitive data.

How can you mitigate this threat?

API Gateway validates that the token provided by the requesting client applications has not been tampered by using OAuth and OpenID connect protocols. API gateway helps in mitigating this vulnerability by enforcing API key verification, OAuth 2.0 authorization protocol flows, and JSON Web Token (JWT) policies.

Programmatic access via a custom policy using Gateway script or XSLT provides the ability to take advantage of more deeper gateway capabilities such as broad security standard support (like SAML, WS-Security, SSL/TLS), configurable AAA, crypto processing, and key protection. Gateway provides a centralized platform for security governance needs.

API gateway also provides seamless integration with Imvision Runtime Protection for more advanced Artificial Intelligence and Machine Learning for detecting broken authorization attacks.

2.2 API2:2019 Broken User Authentication

Threat Description:
Authentication mechanisms are often implemented incorrectly, allowing attackers to compromise authentication tokens or to exploit implementation flaws to assume other user's identities temporarily or permanently. Compromising a system's ability to identify the client/user, compromises API security overall. *[1]*

Technical Relevance:
Authentication is a critical component of any application. Poor implementation of authentication methods, sensitive authentication details like auth tokens or passwords included in the URL, or misconfiguration of JWT can expose authentication threats.

How can you mitigate this threat?

API Connect supports different ways to authenticate users and applications such as API Key, and OAuth2.0 where API Key validation provides basic authentication and OAuth2.0 is a more comprehensive and substantial implementation. For access token management, gateway support JSON Web Token (JWT). The API Gateway oversees the creation and inspection of a JWT token. It also provides easy integration with many industry leading identity access-management solutions like LDAP and Active Directory.

API Gateway is a strong solution for providing a centralized point of control for message authentication. This IBM's solution implements a broad array of security standards (such as OIDC, SAML, Kerberos, SiteMinder, and more). API Gateway policies can also be crafted that will force all session cookies to have consistent expiration policies, to help prevent session hijacking of old or abandoned sessions. To further maximize security, it is recommended to use mutual TLS on all communications to support protection of session IDs and credentials. For more complex needs in data handling or rewriting rules, you can use a programmatic approach via custom Gateway script or XSLT policy to mitigate this threat.

2.3 API3:2019 Excessive Data Exposure

Threat Description:
Looking forward to generic implementations, developers tend to expose all object properties without considering their individual sensitivity, relying on clients to perform data filtering before displaying it to the user. [1]

Technical Relevance:
APIs can expose too much of data. Excessive data exposure can result in attackers using those extra details to gather information they should not have, or to use that information to formulate a more sophisticated attack.

How can you mitigate this threat?

API Connect encrypts data in transit and at rest and offers the capability of redacting message payloads to protect or remove any specific data fields. Gateway supports mutual TLS to secure the pipelines. Additionally, the payload themselves are secured via methods like encryption, hash, and redact. API Gateway provides the ability to rewrite the request and response to protect against any excessive data exposure. It lets you add or remove form parameters, headers, or query parameters to and from a message. You can typically add, change, or remove properties of either the request or response.

It also enforces message mediation policies at the API Gateway level to filter any data that is being returned from an API call, which will ensure that unnecessary, sensitive data will not be exposed to the client. These policies contain rules to reference code that should be executed while having access to all parts of a message being passed through the gateway.

IBM API Connect provides a fully-fledged API Management solution that allows users to configure policies for incoming, and outgoing messages through the gateway via allowlist and blocklist. It also provides a full control over what data should be logged. Unique data handling requirements can be met with scripting via GatewayScript and XSLT.

The API gateway also provides seamless integration with Imvision Runtime Protection for more advanced Artificial Intelligence and Machine Learning detection for excessive data exposure attacks.

2.4 API4:2019 Lack of Resources and Rate Limiting

Threat Description:
Quite often, APIs do not impose any restrictions on the size or number of resources that can be requested by the client/user. Not only can this impact the API server performance, leading to Denial of Service (DoS), but also leaves the door open to authentication flaws such as brute force attacks. *[1]*

Technical Relevance:
This threat occurs when measures have not been put into place to implement rate limiting that would protect the APIs or applications from Denial of Service or brute-force attacks. Without these measures in place, attackers can use an indefinite number of resources to gain access to the applications, limit the functionality, or just stop it from functioning all together.

How can you mitigate this threat?

API Gateway implements various mechanisms to prevent invalid requests from coming to the system like:

- Content filtering (including SQL and XPath Injection)
- Attachment filtering
- Schema Validation (for example: validating XSDs and JSON Schema)
- Provides various OOTB policies for rate limiting the overall number of inbound requests to backend system using mechanisms like subscription to APIs and quota enforcement policies
- Limit against a Payload size, label size, value size and Nest limit
- Provides Header and Query String validation
- XML and JSON Parser limits

- External reference handling
- DDoS (please note that the DDoS is a shared responsibility for the whole network infrastructure).

These protective measures help ensure that valuable back-end resources such as CPU and memory usage are not misused.

2.5 API5:2019 Broken Function Level Authorization

Threat Description:
Complex access control policies with different hierarchies, groups, and roles, and an unclear separation between administrative and regular functions, tend to lead to authorization flaws. By exploiting these issues, attackers gain access to other users' resources and/or administrative functions. *[1]*

Technical Relevance:
Many modern API based applications have APIs with multiple resources to perform various tasks thereby increasing the need for proper authorization checks while considering the user hierarchy. It can result in attackers being able to access important and sensitive functionality and information that is only otherwise available for the authorized users. Generally, administrative functions are the key targets for this type of attack.

How can you mitigate this threat?

API Connect supports Role Based Access Control (RBAC) across all scopes for granular access control. It also supports a hierarchical organization (Org) structure and allows defining sub-admins for fine-grained access controls across all scopes. In API Connect, an Org can contain one or more Catalogs, and a Catalog can contain one or more Spaces.

- Each Org has its administrator(admins) that control and define teams/resources/processes at the Org scope.

- Each Catalog has its admin that control and define teams/ resources/processes at the Catalog scope.
- Each Space has its admin that control or define teams/ resources/ processes at the Space scope.

In API Gateway, OAuth scope-based access control functionality can also be used to mitigate this vulnerability. Once the OAuth token is received from the user, the gateway can validate it before allowing the request to pass through to the trusted zone.

API gateway also provides seamless integration with Imvision Runtime Protection for more advanced Artificial Intelligence and Machine Learning for detecting a broken authorization attack.

2.6 API6:2019 Mass Assignment

Threat Description:
Binding client provided data (e.g., JSON) to data models, without proper properties filtering based on an allowed list, usually lead to Mass Assignment. Either guessing object properties, exploring other API endpoints, reading the documentation, or providing additional object properties in request payloads, allow attackers to modify object properties they are not supposed to. *[1]*

Technical Relevance:
When APIs expose resources or variables, attackers can use these to craft their requests and calls to access resources inappropriately. API based applications contain objects that have many properties, of which only a few should be updated directly by a client. There are also properties that should not be edited by a client, such as permission related properties that are set only by admins. If these properties are identified by attackers, it could lead to privilege escalation, data tampering or could result in bypassing stringent security mechanisms.

How can you mitigate this threat?

API Gateway provides several useful features for ensuring robust data filtering implementations capability via allowlist and blocklist. It provides schema validation to prevent any exposure of confidential data to a client application. The gateway provides the ability of screening all API requests that contain parameters that are defined as sensitive and should only be changed internally from the application.

Additionally, it is capable of screening parameters passed in the API request by scope and allows one to define outputs and payloads explicitly. It provides Transport Level Security (TLS, HTTPS) and message level protection to ensure that the message integrity is protected against any vulnerability.

API Gateway also provides seamless integration with Imvision Runtime Protection for more advanced detection of mass assignment attacks.

2.7 API7:2019 Security Misconfiguration

Threat Description:

Security misconfiguration is commonly a result of unsecure default configurations, incomplete or ad-hoc configurations, open cloud storage, misconfigured HTTP headers, unnecessary HTTP methods, permissive Cross-Origin resource sharing (CORS), and verbose error messages containing sensitive information. *[1]*

Technical Relevance:

Misconfigured security settings in APIs or applications can lead to attackers exploiting vulnerable settings to exploit the application. These could be misconfigured TLS, unpatched services, outdated services or plugins, any unprotected files, or misconfigured CORS policies.

How can you mitigate this threat?

With API Gateway, you can establish and standardize the hardening and patching processes via role-based access control (RBAC) at the API connect level to provide full control over the configuration of the Gateway policies. IBMs solution allows you to develop a governing ecosystem around the API life cycle. It provides the ability to ensure adherence to security policies, configuration requirements and proactively identify, diagnose, and resolve the security incidents. Gateway also ensures that the configurations are stored securely.

By pulling security policies and functions away from the API implementations and centralizing them on the API Connect security gateway, the chance of security misconfiguration is reduced significantly because the number of systems that contain security processing code is being reduced.

Gateway also supports Open API definition and is well equipped to deal with this type of vulnerability by allowing CORS configuration (setting up of the access control mechanism for headers and methods).

2.8 API8:2019 Injection

Threat Description:

Injection flaws, such as SQL, NoSQL, Command Injection, etc., occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's malicious data can trick the interpreter into executing unintended commands or accessing data without proper authorization. *[1]*

Technical Relevance:

The data supporting an API is often of a critical nature. Injection attacks, such as SQL Injection can allow attackers to gain access or otherwise compromise this data.

How can you mitigate this threat?

API GW has built-in OOTB policies which customers can apply for different threat protections or different enforcements.

- Schema Validation: The gateway will act as an enforcement point so that you can have the valid requests go through and invalid requests being blocked with appropriate error reporting to the client as well as proper and accurate logging of those error messages.
- Provides data type checking for invalid input
- XML and JSON Threat Protection
- Provides SQL/XSS Injection filter and regular expression filter configuration: It could be any common regex or string that malicious client tries to inject to manipulate the data. You can look for different regexes like DELETE, EXECUTE, DROP, INSERT, or SQL code key parameters that a malicious client could send in the incoming request.
- Provides integration with Imvision Runtime Protection for powerful Artificial Intelligence and Machine Learning capabilities to detect Injection attacks.

2.9 API9:2019 Improper Assets Management

Threat Description:
APIs tend to expose more endpoints than traditional web applications, making proper and updated documentation highly important. Proper hosts and deployed API versions also play an important role to mitigate issues such as deprecated API versions and exposed debug endpoints. *[1]*

Technical Relevance:
Non-production or earlier versions of API's that are still in use or not decommissioned properly become the potential targets for attackers.

How can you mitigate this threat?

This threat can be addressed by leveraging the API Connect matured capabilities to manage the full API life cycle allowing you to create a comprehensive governance model. API Gateway Supports HTTPS and TLS 1.3 protocols for secured communication across network infrastructure.

2.10 API10:2019 Insufficient Logging and Monitoring

Threat Description:

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 to tamper with, extract, or destroy data. Most breach studies demonstrate the time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring. *[1]*

Technical Relevance:

Improper logging and alerting can allow attacks to go undetected and therefore a strategy should be required to obtain insights over critical client events.

How can you mitigate this threat?

API analytics provides you with the ability to see client traffic, user-based traffic, and product-based traffic in a transaction flow. It provides a rich analytics engine that allow you to dive into each individual policy to learn more about how API program or API proxies are performing. API analytics stores the API event logs as they are processed from the gateway service, process API event logs from the gateway and map GUIDs to provide usable analytics fields for almost real-time visibility into a complete API transaction flow.

The API Gateway can also be integrated with IBM Security Qradar for intelligent security analytics for having insights into the most critical threats.

API monitoring tool called DataPower Operations dashboard can be used for deep API transactional insights and capabilities like:

- Visualizing the aggregated metric data from the API events, so that the API providers can better understand their APIs' health, and consumption
- Surfacing the API calls raw log data to help developers debug

Customers can also offload the API event records to a target location (like Splunk, Syslog, Kafka, HTTP, etc.). In the gateway you can send logging within the proxy layer. With threat protections, you can look for a particular string or validation when the application sends it.

API Connect Testing and Monitor is another powerful tool that is easy to use and enables the customers to test an API from the outside world to assess the health on the API level. For auditing or administrative actions, the gateway logs all activity, along with essential metadata such as date/time stamps, source IP addresses, and user IDs. By default, the log records cannot be deleted or modified.

3. API Connect Addressing the OWASP Issues for GraphQL

GraphQL is an open-source query language used to build APIs as an alternative to REST and SOAP. The various areas that need to be considered when working with GraphQL are as below:

- Applying proper input validation checks on all incoming data
- Prevention against DDoS attacks
- Access control checks
- Disable insecure default configurations (eg. introspection, GraphQL, excessive errors, etc.)

3.1 Common Attack 1: Injection

Threat Description:
Attackers will feed the API with malicious data through whatever injection vectors are available. *[2]*

Technical Relevance:
This attack could include any SQL or NoSQL injection, OS command injection, a server-side request forgery (SSRF) injection, or a carriage return line feed (CRLF) injection. This GraphQL attack is similar to the injection threat in REST APIs.

How can you mitigate this attack?

API gateway provides you the capability to perform GraphQL schema validation on the input by providing a data type checking. API Gateway provides the ability to validate all incoming data via allowlist and blocklist on the values. In the schema tab you can remove any type or field. Gateway provides the ability to validate the encoding of incoming data to prevent user control over data flow.

3.2 Common Attack 2: Denial of Service

Threat Description:
DoS is an attack against the availability and stability of the API that can make it slow, unresponsive, or completely unavailable. [2]

Technical Relevance:
Attackers will attack the system on its availability and stability of the API. Due to its query nature, the Big-O complexity can increase exponentially. This GraphQL attack is similar to the Denial of Service (DoS) or DDoS attack in REST APIs.

How can you mitigate this attack?

API gateway provides you the ability to limit the query depth, and the amount of data for incoming queries via setting up a proper rate limiting policy and adding a reasonable timeout for the incoming queries. The gateway also allows you to add pagination to limit the amount of data that can be returned in a single response.

API gateway provides you with a unique capability of setting up a proper cost analysis for query lookup to prevent any DoS or DDoS attacks by enforcing a maximum allowed cost per query. You can also enforce rate limiting on incoming requests based on IP or user to prevent DoS attacks.

Please visit the below links to learn more about IBM's solution to prevent Denial of service attack while working with GraphQL APIs.

a) Securing GraphQL with Cost derivatives:
https://www.youtube.com/watch?v=_WpM7qylvYg

b) GraphQL cost derivatives specification
https://ibm.github.io/graphql-specs/
https://ibm.github.io/graphql-specs/cost-spec.html

3.3 Common Attack 3: Abuse of broken authorization

Threat Description:
The user issuing the GraphQL request does not or should not have the process permission to the query/data. [2]

Technical Relevance:
This threat is about the broken authorization level either via an improper or an excessive access, or any insecure direct object references (IDOR).

How can you mitigate this attack?

API gateway ensures that GraphQL has the proper access control by validating that the requester is authorized to view or update the data it is requesting via Role Based Access Control (RBAC). RBAC capability helps API gateway to prevent any insecure direct object reference threat, broken function level authorization or broken object level authorization and validates that the user issuing the GraphQL has the necessary process permission to the query and the data. It also ensures that the authorization checks are applied at both the edges and nodes of a graph.

API gateway allows you to disable introspection query by the requester. In the assembly policy, it provides the ability to customize the security signature. You can also change the default endpoint/url. API gateway, provides the schema filtering per plan and provides a more fine-grained filtering on runtime context depending on the user or the IP.

3.4 Common Attack 4: Batching

Threat Description:
This attack is a GraphQL-specific form of brute force attack. Attackers can enumerate every object and query them all in a single request transaction. This is very related to Common Attack 2: DoS.

Technical Relevance:

By design, GraphQL optimizes for the client by letting it batch many requests in a single transaction query.

How can you mitigate this attack?

Many of the mitigations against general DoS attacks, listed above, apply here as well.

API gateway provides you the ability to limit the amount of data for incoming queries via setting up a proper rate limiting policy and adding a reasonable timeout for the incoming queries.

API gateway provides you with a unique capability of setting up a proper cost analysis for query lookup, as an input to threat protection and rate limiting. This can allow appropriate batching while preventing excessive batching, customized per use case. Please visit the links above to videos and the formal specification.

3.5 Common Attack 5: Configuration

Threat Description:

This attack relates to GraphQL server configuration and is similar to the Security misconfiguration threat in REST APIs. Introspection provides a great way for attackers to get insight of the system.

Technical Relevance:

By design, GraphQL implementations have some insecure configurations like introspection ability, or returning excessive data in error messages.

How can you mitigate this attack?

API gateway provides the capability to disable the introspection or change the endpoint/url. You can also limit the system to return a generic error message and avoid sending too much information in the error message to the end client.

4. Contributors

The following were involved in deciding and validating the content along with the author for this whitepaper:

- Shiu Fun Poon
- Morris Matsa
- Nate Ziemann
- Chris Khoury
- Bob Johnson
- Andrew White

5. References

[1] OWASP API Security Project: https://owasp.org/www-project-api-security/

[2] GraphQL APIS from OWASP cheat sheet series which are listed here: https://cheatsheetseries.owasp.org/cheatsheets/GraphQL_Cheat_Sheet.html

[3] IBM API Connect v10.x Deployment WhitePaper, by Christopher Philips: https://community.ibm.com/community/user/integration/viewdocument/api-connect-deplyoment-whitepaper-v?CommunityKey=2106cca0-a9f9-45c6-9b28-01a28f4ce947&tab=librarydocuments

[4] IBM (2012) WebSphere DataPower: Build a more-secure web application infrastructure. *IBM* Available at http://hosteddocs.ittoolbox.com/

[5] Gartner Magic Quadrant for Full Life Cycle API Management published 28 September 2021, by Shameen Pillai, Kimihiko Iijima, Mark O'Neill, John Santoro, Akash Jain, Fintan Ryan: https://www.ibm.com/account/reg/signup?formid=urx-51300

[6] Forrester Wave: API Management Solutions, Q32020, complimentary reprint, http://ibm.biz/TheForresterWave-APIM2020

