**Module 8: Anypoint Security**

1. Why should we use Anypoint Security when Mulesoft provides security through API

Manager’s API policy?

* Any point Security provides advanced defense for your APIs and integrations.
* Protect sensitive data, stop threats at the edge, and automatically enforce security best practices to protect and govern your application network.
* Anypoint Security provides a layered approach to secure your application network.
* These layers work together to protect both the application network and the network’s individual nodes by controlling access to APIs, enforcing policies, and proxying all inbound or outbound traffic to mitigate external threats and attacks.  
  Anypoint Security provides you with a dedicated endpoint to detect attacks and validate traffic without taxing your network implementations.
* Anypoint Security features a high-performance, reliable, and scalable service that works with Anypoint Runtime Fabric to enforce security policies on nodes deployed to your Anypoint Runtime Fabric. Anypoint Security provides denial-of-service (DoS), IP whitelist, HTTP limits, and Web Application Firewall (WAF) policies to protect your APIs.
* You can use the Anypoint Security policies to manage all traffic to your Runtime Fabric, and leverage API Manager policies to apply specific behaviours to specific APIs.

2. Why is Anypoint security not available in deployment models other than RTF?

3. Can you explain the architecture of the Anypoint security?

* Anypoint Security provides a layered approach to secure your application network.
* These layers work together to protect both the application network and the network’s individual nodes by controlling access to APIs, enforcing policies, and proxying all inbound or outbound traffic to mitigate external threats and attacks.
* Anypoint Security provides you with a dedicated endpoint to detect attacks and validate traffic without taxing your network implementations.



4. When should I go with Anypoint Security for securing my applications.

* If we want to provide advanced defence to our APIs and applications that are deployed in RTF deployment model – use Anypoint security

5. Can you provide some advantages and disadvantages of Anypoint security over API

Manager’s API policies?

6. How can we enforce security policies in Anypoint Security?

7. Can you name some of the EDGE policies that can be applied in Anypoint Security?

* IP Whitelist policy
* Denial of Service Policy
* Http Limit Policy
* Web application firewall policy

8. How do I whitelist certain ips using Anypoint security?

By creating an explicit list of IP addresses that can access the deployed application.

IP addresses that are not in the white list will be rejected when they try to access application

9. What is the use of Denial Of Service Policy in Anypoint Security?

* Prevent attackers from flooding network to prevent legitimate network traffic to your APIs
* If the policy encounters more errors than your configured threshold coming from the same IP address, the policy can either drop the connection silently or drop the connection immediately and return a 503 error.
* DoS policies are designed to protect your network nodes against malicious clients trying to flood your network to prevent legitimate traffic to your APIs.

10. What are the capabilities of EDGE policies?

* Anypoint Security EDGE policy features a high-performance, reliable, and scalable service that works with Anypoint Runtime Fabric to enforce security policies on nodes deployed to your Anypoint Runtime Fabric.
* Anypoint Security provides denial-of-service (DoS), IP whitelist, HTTP limits, and Web Application Firewall (WAF) policies to protect your APIs.
* You can use the Anypoint Security policies to manage all traffic to your Runtime Fabric, and leverage API Manager policies to apply specific behaviors to specific APIs.

11. What do you mean by Web Application Firewall Policy?

* WAF (Web application firewall protection) for all API’s
* A WAF or Web Application Firewall helps protect web applications by filtering and monitoring HTTP traffic between a web application and the Internet.
* **Advantages:**
  + Offers more visibility into sensitive application data
  + Can prevent application layer attacks
  + It’s easy to deploy and manage
* The Web Application Firewall (WAF) security policy is available for request and response traffic.
* Policies are grouped into the major threat categories for requests and responses.
* WAF policies are fully integrated with the existing Any point Security policy DoS (Denial of Service).
* When the WAF policy detects errors, it triggers the thresholds configured in the DoS, which can be optionally configured to take actions such as shaping or blocking traffic for an IP address from a malicious source.

12. How do you prevent SQL Injection using Anypoint Security?

* SQL injection is an attack used to take over the database servers by dynamically poisoning SQL queries.
* SQL injection can be prevent by enabling WAF ruleset options. SQL injection is one of the threat category in WAF Request Ruleset
* We can enable the option **Detect and reject violations** if there is a SQL injection threat
* The request is rejected and returns a response status of HTTP/1.1 400 BAD REQUEST - web application firewall

13. How do you manage your certificates in Anypoint Security?

Anypoint Security provides a secure vault for you to store the TLS certificates and keystores used by your deployments.

14. What QoS policies are available in Anypoint Security?

Denial of Service

15. What do you mean by Certificate Revocation List?

A Certificate Revocation List (CRL) is a list of [digital certificates](https://searchsecurity.techtarget.com/definition/digital-certificate) that have been revoked by the issuing [Certificate Authority](https://searchsecurity.techtarget.com/definition/certificate-authority) (CA) before their scheduled expiration date and should no longer be trusted. CRLs are a type of blacklist and are used by various endpoints, including [Web browsers](https://searchwindevelopment.techtarget.com/definition/browser), to verify whether a certificate is valid and trustworthy.

16. How do you protect sensitive data using a Tokenization method in Anypoint Security?

Tokenization is a highly effective way to protect sensitive data. When you tokenize data, sensitive data elements are substituted with randomly generated non-sensitive data elements.

Examples of sensitive information that are suitable for tokenization protection include:

* Primary Account Number (PAN)
* Personally Identifiable Information (PII)
* Protected Health Information (PHI), or any information deemed sensitive

17. How do tokens stored in Anypoint Security?

18. How do you retrieve the original values in the Tokenization method?

Detokenization is the process of returning the previously masked sensitive data back into its original value to reduce the risk of compromising sensitive information.

Detokenized sensitive data must always be read under strict security controls.

19. Can you provide some examples of you using tokenization in your project?

20. In which scenario you will MASK sensitive information rather than Encrypting the

information.

* Masking is a one-way process, thus we usually go with Masking when we don’t want to retrieve the sensitive information
* Masking of account number, phone number
* Common use case of masking
* Log obfuscation – mask sensitive data while logging
* Login anonymization - when we login to a system password will be masked

21. How do you encrypt sensitive information in Anypoint security?

Anypoint security follows a format preserving encryption. In this the data is encrypted using some techniques while the format of the data is maintained. Mule Security offers the

ability to encrypt or decrypt message content in within a Mule flow by utilizing Mule message processors, allowing you to maintain the integrity of your messages

22. What is the use of a Secrets manager in Anypoint Security?

Secret Manager is used to read, write and manage the secret keys and TLS (transport layer security) certificates