Secure your Serverless Infrastructure with Check Point's CloudGuard Dome9

In a previous blog, we had discussed how serverless security requires a security-centric approach. To recap, serverless security requires a holistic approach, where security of AWS Lambda functions, as well as various other cloud-native services (such as S3, and DynamoDB) are continuously protected. In this blog, we will dive specifically into a few aspects of securing AWS Lambda functions, communication to those services, and how CloudGuard Dome9 can help address some of those security challenges.

Serverless Security Challenges

1. **Securing Lambda Privileges**

As your application scales, you might be using step functions or scaling to 100s of Lambda functions. This results in more permissions that need to be managed and fine tuned which can be a challenging task in a highly dynamic cloud environment across thousands of AWS accounts. If a Lambda function is compromised, the most important security defense is to restrict what these compromised functions can do (aka privileges). Additionally, as the number of Lambda functions grow, it is important to ensure that there should not be any policies that grant blanket permissions ('*') to resources.

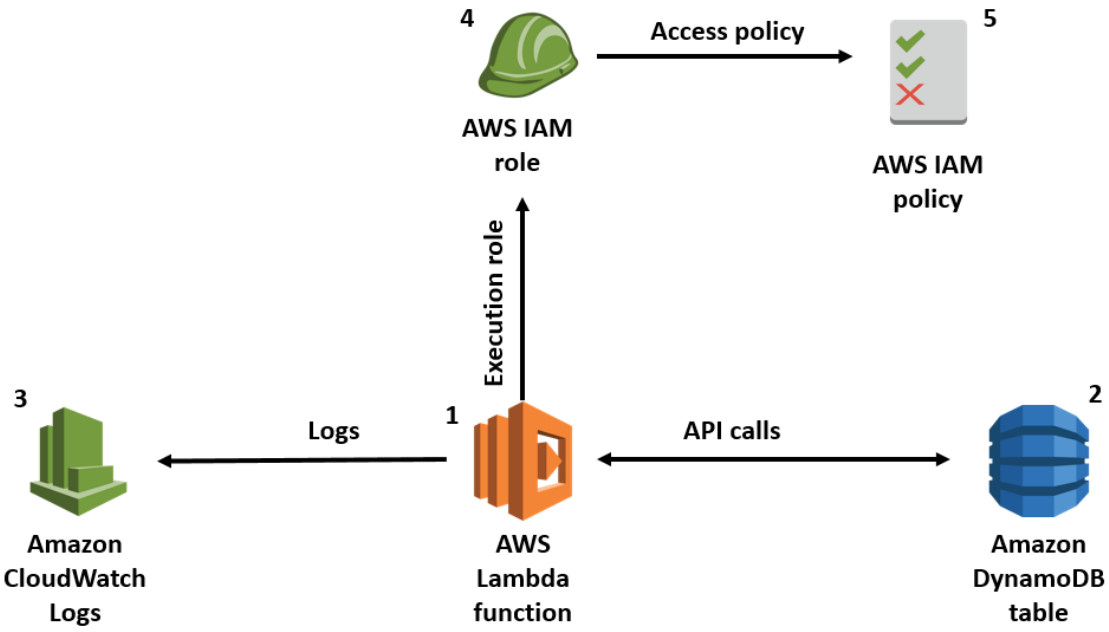
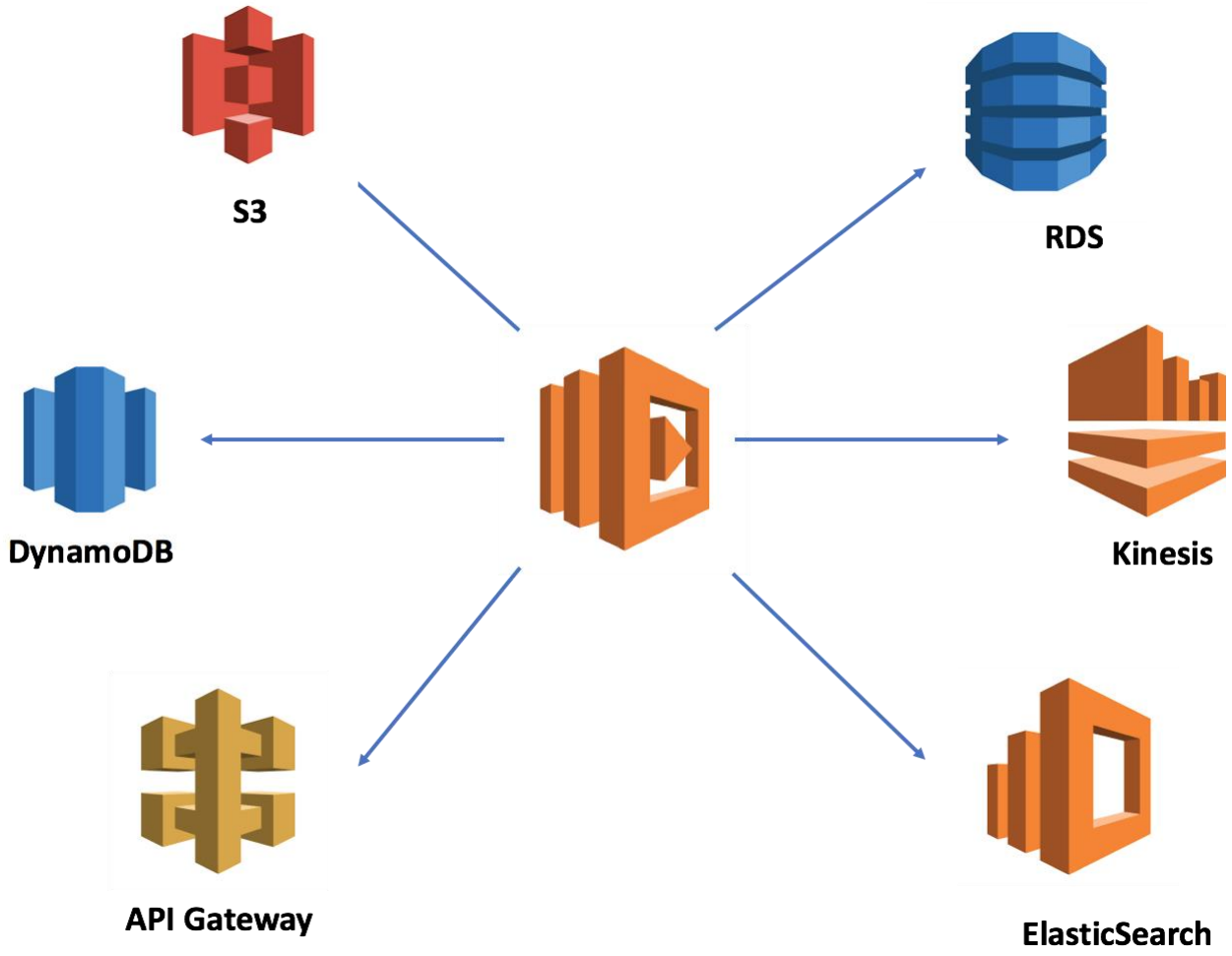


Image taken from AWS

2. Securing External Data

Communication to all AWS services from your Lambda function needs to be encrypted and authenticated. The same due diligence must also be applied when storing sensitive data. A handful of AWS services also offer server-side encryption for your data at rest—S3, RDSand Kinesis streams, and Lambda has built-in integration with KMS to encrypt your functions' environment variables. For services/DBs that do not offer built-in encryption—eg. DynamoDB, Elasticsearch, etc. it is easy to forget. In the case of a data breach, this can cost organizations,



How CloudGuard Dome9 Can Help

Typically admins use IAM roles to ensure fine grained control over who can invoke which actions on which resources. But it is still imperative to ensure you are adhering to the Principle of Least Privilege (POLP) when configuring Lambda permissions. In the serverless framework, the default behaviour is to use the same IAM execution role for all functions in the service. While this might seem an easy choice, it is not a good practice to follow.

The recommended practice is to have one IAM role per each Lambda function in order to follow the POLP. With CloudGuard Dome9, you can ensure that your Lambda functions will have the minimum privileges needed to perform the required tasks.

List<Lambda> should not have items groupBy [executionRoleArn] length() > 1

It is also recommended and considered a standard security best practice to grant minimal or least privilege, allowing only the permissions required to perform a task. As best practice, security teams need to determine the specific permissions needed by your Lambda functions, and then craft IAM policies for these permissions only, instead of full administrative privileges. With CloudGuard Dome9, you can quickly assess if any of your functions have blanket permissions.

Lambda should not have executionRole.combinedPolicies contain [policyDocument.Statement contain-any [Effect = 'Allow' and (Resource ='*' or Resource contain[$='*']) and Action contain ['%*%'] or Action ='*' or Action contain [$='*']]]

Use secure transport when transmitting data to and from services (both external and internal ones). If you're building APIs with API Gateway and Lambda, then you're forced to use HTTPS by default, which is a good thing. However, the API Gateway is always publicly accessible and you need to take the necessary precautions to secure access to internal APIs.

DynamoDB / Kinesis / S3

It is recommended to enable Server Side Encryption (SSE) of your AWS Kinesis Server data at rest, in order to protect your data and metadata from breaches or unauthorized access, and fulfill compliance requirements for data-at-rest encryption within your organization. With CloudGuard Dome9, you can quickly scan and assess whether Kinesis has encryption turned on.

Kinesis should have encrypted=true

You can also easily verify if server side encryption is enabled on DynamoDB tables with the innovative Dome9 Governance Specification Language (GSL) found within the Compliance Engine of CloudGuard Dome9.

DynamoDbTable should have encrypted=true

For S3 buckets, best practice dictates that all data in the cloud be encrypted both at rest as well as in flight when data is read from or written to a bucket. Read more about how to encrypt data in flight to S3 buckets can help protect against man-in-the-middle and sniffing attacks.

With CloudGuard Dome9, you can quickly ensure that S3 Buckets have server side encryption at-rest enabled to protect sensitive data for at-rest data.

S3Bucket should have encryption.serverSideEncryptionRules

You can also ensure that S3 Buckets enforce encryption of data transfers using Secure Sockets Layer (SSL) for in-transit communication.

S3Bucket should have policy.Statement contain [Effect='Deny' and Condition.Bool.aws:SecureTransport='false'] and policy.Statement contain [Action contain ['s3:GetObject'] or Action contain ['s3:*']]

For further guidance, feel free to check out our open source Cloud Security Posture Repository (CSPR) of comprehensive security controls and compliance checks for your cloud environments.

Dome9 is now part of the Check Point Software Technologies family as a core offering to enhance its cloud security portfolio. Check Point CloudGuard IaaS and CloudGuard Dome9 combine to offer a comprehensive security solution for public cloud environments. To learn more, please visit CloudGuard Dome9.

