Monitoring Serverless Architectures in AWS

aws partner network

Advanced
Technology Partner

SaaS Partner

The introduction of serverless architectures is a positive development from a security perspective. Splitting up services into single-purpose functions with well-defined inputs and outputs helps reduce exposure to many types of threats. However, the security of serverless architectures is an under-studied topic. Many practices and tools available in more traditional architectures don't apply.

This document will discuss methods of auditing and monitoring of AWS Lambda functions–a key component of serverless architectures in AWS – and how Cisco® Stealthwatch Cloud puts these methods into practice.

Lambda security topics

- **Access to the function:** Who or what can modify or invoke a function?
- **Access from the function:** What can the function access when it is running?
- **Unintended behavior:** Can the function be made to do something unexpected?

For each area, there is a question of auditing, or looking to see whether configurations match expectations, and monitoring, or keeping track of activities on an ongoing basis.

Contents

Auditing

The AWS Lambda documents describe how to use the AWS Identity and Access Management (IAM) API and Lambda API for auditing policies related to function access. It's important to audit policies periodically to check:

- Whether some policies are outdated. For example, they made sense in the past, but haven't been updated to reflect changes (for instance, a user who has since moved to another team still has rights to change an Amazon Lambda function's code)
- Whether some policies allow access to more resources than necessary (for example, an Amazon Lambda function that can write to any Amazon Simple Storage Service - Amazon S3 bucket

The AWS IAM API can be used to show which users (or roles) in an AWS account have the ability to change a Amazon Lambda function or its configuration. For example, we could query the AWS IAM API to see who has the Update Function Code permission. This would show who is allowed to change the code a function executes. The Amazon Lambda API can be used to show who (or what) is allowed to invoke a function. It's common to allow other AWS services like Amazon S3 or API Gateway to call a function. An example of something to look for would be cross-account access for Amazon S3. Note that the AWS documents provide this warning for this problem:

"If you add a permission for the Amazon S3 principal without providing the source ARN, any AWS account that creates a mapping to your function ARN can send events to invoke your Lambda function from Amazon S3."

The Amazon Lambda and AWS IAM APIs together can show what an Amazon Lambda function is allowed to do. For example, to see what permissions a particular Amazon Lambda function is granted (for instance, writing to an Amazon S3 bucket), you can use the Amazon Lambda API's GetFunction call. That will help you determine which role to look up with the AWS IAM API.

Monitoring

The Amazon Lambda documents describe how to use AWS CloudTrail and Amazon CloudWatch for monitoring.

AWS CloudTrail tracks API activities. We can use AWS CloudTrail logs to look for changes to functions (for example, the UpdateFunctionCode call described above) as well as changes made by functions. This helps keep track of access between audits.

Amazon CloudWatch Logs track function invocations and their output. This can be used for billing purposes (they include information about duration and memory usage, which are part of Amazon Lambda pricing) as well as for monitoring purposes. For example, a function can be made to write its input to the log, which could be used to help look for injection attacks.

Amazon CloudWatch Metrics also track function usage, such as the number of invocations, the duration of each run, and other metrics. Manual alarms can be set for things like excessive invocations. This can be used to monitor for unintended behavior and abuse. For example, if the function can be invoked from the outside, an alarm could show whether it is being abused and wasting resources.

Extending monitoring with VPC Flow Logs

The auditing and monitoring resources described in the AWS documentation help answer some of our security-related questions, but miss some others. For example, it would be difficult to use the services above to determine whether an Amazon Lambda function is accessing an internal Amazon Relational Database Service (RDS) or Amazon ElastiCache instance inside a VPC.

We can get more detailed monitoring of an Amazon Lambda function activity by understanding how functions communicate. As the AWS documentation describes, Amazon Lambda functions are allocated temporary Elastic Network Interfaces (ENIs) and IP addresses when they need to communicate with a VPC resource. These ENIs can be enumerated with the Amazon Elastic Compute Cloud (EC2) API and matched to the requesting Lambda function.

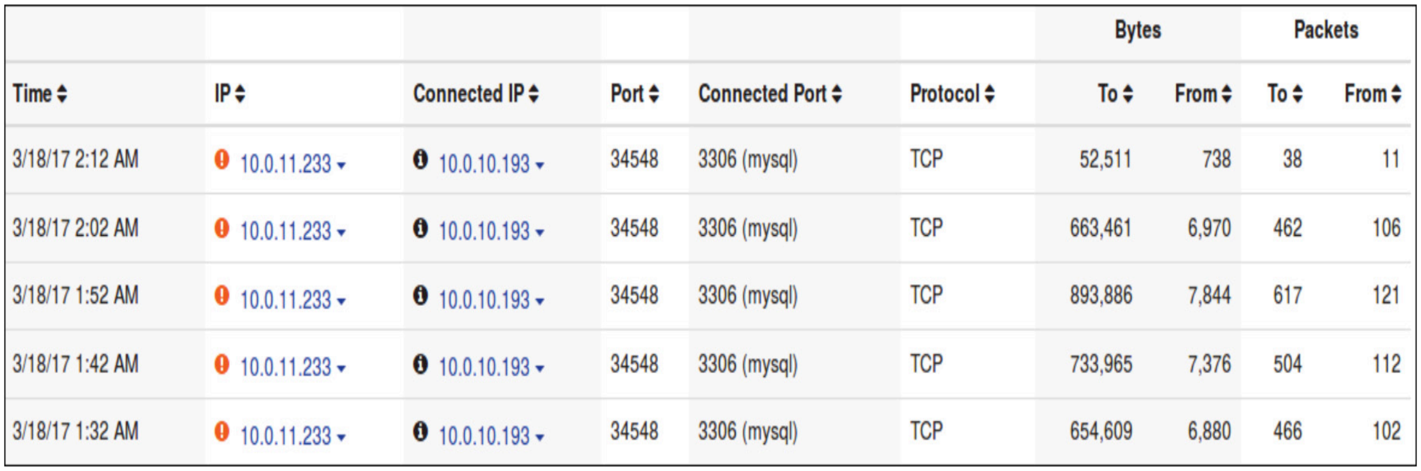
Once the ENIs and IP addresses are known, an Amazon Lambda function's activity can tracked using VPC Flow Logs. These flow logs describe all of the communication to, from, and within a VPC, so if a Lambda function interacts with an internal resource, the associated network activity will be logged.

Furthermore, Amazon Lambda functions need to use a Network Address Translation (NAT) instance or VPC NAT gateway to gain Internet access. This means their Internet activity will also go through the VPC and be logged as flow log records.

This level of detail helps answer other security questions about Amazon Lambda. More types of unintended behavior besides function invocation and API calls can be tracked. The effects of something exotic like container escape (a breach of the environment that runs the function) could be detected if it was followed by access to VPC resources.

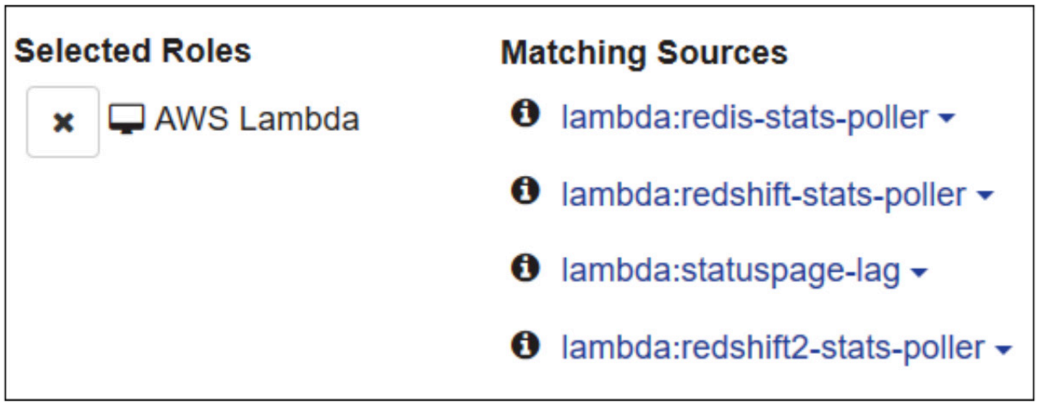
Stealthwatch Cloud integration

Cisco Stealthwatch Cloud uses each of the resources described above to track Amazon Lambda functions. VPC Flow Logs allow monitoring of Amazon Lambda functions with Stealthwatch Cloud's entity modeling service, which helps to enable a wide variety of security alert types. The screenshot below (from the Stealthwatch Cloud web portal) shows a set of VPC Flow Log records for a Amazon Lambda function accessing an Amazon RDS database running MySQL.



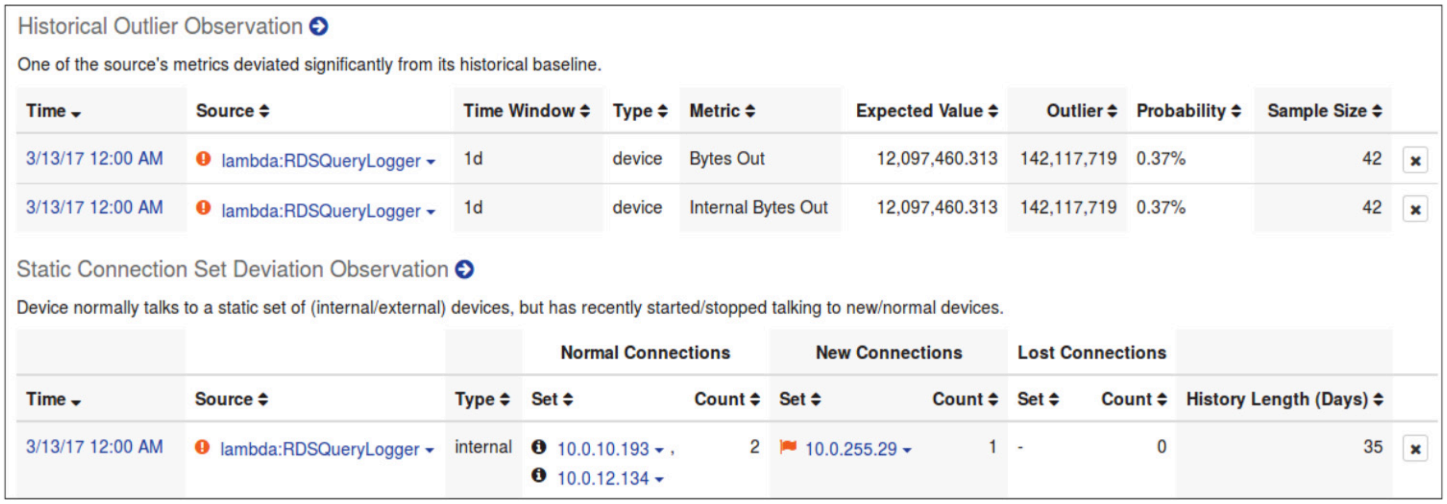
						Bytes		Packets	
Time ⬍	IP ⬍	Connected IP ⬍	Port ⬍	Connected Port ⬍	Protocol ⬍	To ⬍	From ⬍	To ⬍	From ⬍
3/18/17 2:12 AM	10.0.11.233 ▾	10.0.10.193 ▾	34548	3306 (mysql)	TCP	52,511	738	38	11
3/18/17 2:02 AM	10.0.11.233 ▾	10.0.10.193 ▾	34548	3306 (mysql)	TCP	663,461	6,970	462	106
3/18/17 1:52 AM	10.0.11.233 ▾	10.0.10.193 ▾	34548	3306 (mysql)	TCP	893,886	7,844	617	121
3/18/17 1:42 AM	10.0.11.233 ▾	10.0.10.193 ▾	34548	3306 (mysql)	TCP	733,965	7,376	504	112
3/18/17 1:32 AM	10.0.11.233 ▾	10.0.10.193 ▾	34548	3306 (mysql)	TCP	654,609	6,880	466	102

Amazon Lambda functions are tracked by name, so as they change ENIs and IP addresses, their activity will be updated:



Selected Roles	Matching Sources
AWS Lambda	lambda:redis-stats-poller
	lambda:redshift-stats-poller
	lambda:statuspage-lag
	lambda:redshift2-stats-poller

Amazon Lambda functions are good examples of static devices, or entities that express a very narrow set of behaviors. This makes identifying even subtle changes that may indicate a security concern possible. The screen-shot below shows an alert for an Amazon Lambda function that normally connects to two internal resources connecting to an unexpected third.



Historical Outlier Observation

One of the source's metrics deviated significantly from its historical baseline.

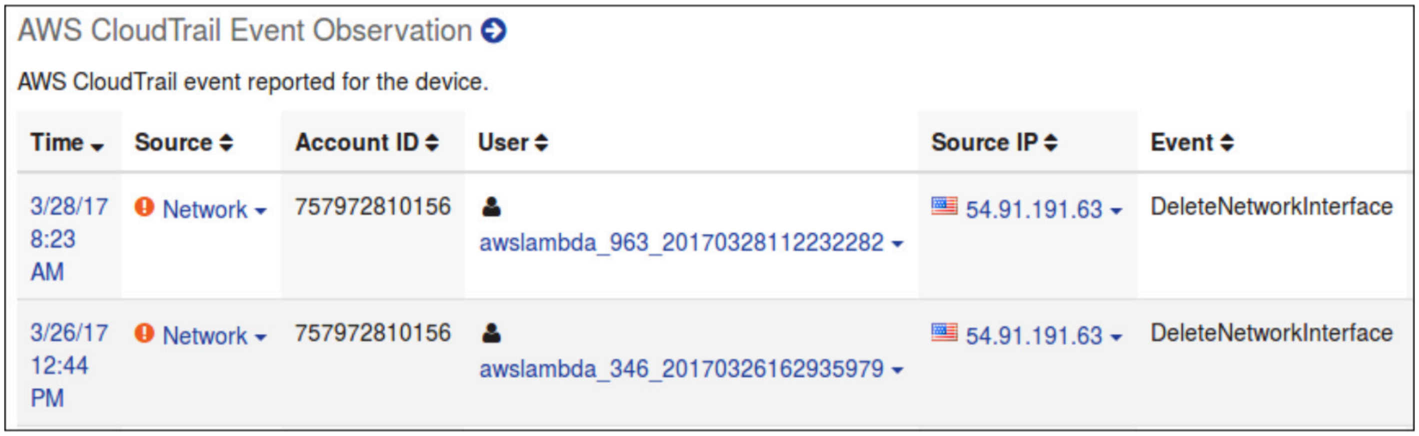
Time	Source	Time Window	Type	Metric	Expected Value	Outlier	Probability	Sample Size
3/13/17 12:00 AM	lambda:RDSQueryLogger	1d	device	Bytes Out	12,097,460.313	142,117,719	0.37%	42
3/13/17 12:00 AM	lambda:RDSQueryLogger	1d	device	Internal Bytes Out	12,097,460.313	142,117,719	0.37%	42

Static Connection Set Deviation Observation

Device normally talks to a static set of (internal/external) devices, but has recently started/stopped talking to new/normal devices.

			Normal Connections		New Connections		Lost Connections		
Time	Source	Type	Set	Count	Set	Count	Set	Count	History Length (Days)
3/13/17 12:00 AM	lambda:RDSQueryLogger	internal	10.0.10.193 , 10.0.12.134	2	10.0.255.29	1	-	0	35

The Stealthwatch Cloud service also reads from AWS CloudTrail, so if an Amazon Lambda function executes an unusual API call (due to code injection, for example), that can generate an alert.

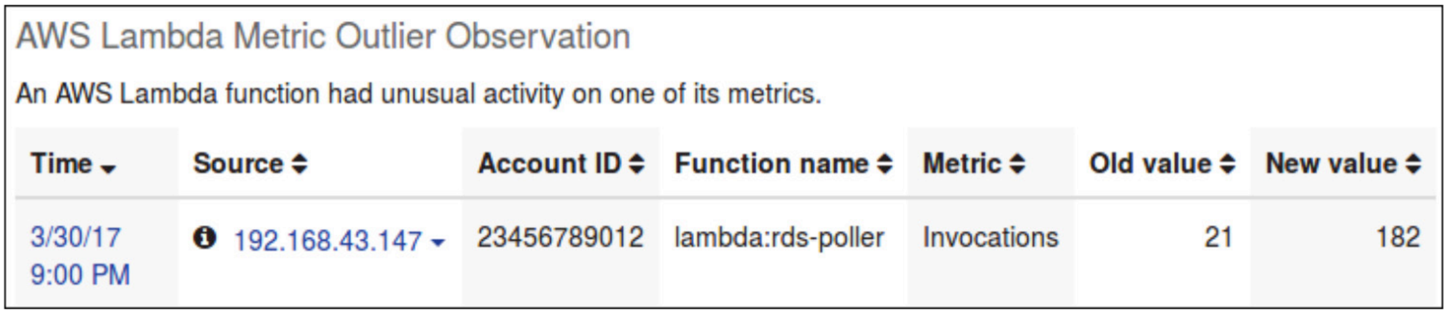


AWS CloudTrail Event Observation

AWS CloudTrail event reported for the device.

Time	Source	Account ID	User	Source IP	Event
3/28/17 8:23 AM	Network	757972810156	awslambda_963_20170328112232282	54.91.191.63	DeleteNetworkInterface
3/26/17 12:44 PM	Network	757972810156	awslambda_346_20170326162935979	54.91.191.63	DeleteNetworkInterface

Additionally, Stealthwatch Cloud polls CloudWatch metrics, so if the function is invoked, many more times than normal (due to abuse from the outside), that would generate an alert.



AWS Lambda Metric Outlier Observation

An AWS Lambda function had unusual activity on one of its metrics.

Time	Source	Account ID	Function name	Metric	Old value	New value
3/30/17 9:00 PM	192.168.43.147	23456789012	lambda:rds-poller	Invocations	21	182

The normal level of invocations is determined automatically, so a user doesn't need to specify what the expected invocation rate is.

Conclusions and recommendations

In this white paper, we have:

- Described the standard tools for auditing and monitoring Lambda function security– the Amazon Lambda and IAM APIs, AWS CloudTrail, and Amazon CloudWatch.
- Explored details of how Amazon Lambda functions access internal resources to discover another way to monitor behavior (VPC Flow Logs).
- Explained how Stealthwatch Cloud uses these tools to help enable entity modeling-based security alerts.

There are some simple steps Lambda users can take to improve their security stance with standard tools:

- **Auditing:** At the very least, use the AWS Console to look through the list of the Lambda function's execution roles and triggers. Check whether any seem overly broad (for instance, allowing Full Access to some service like Amazon S3) and set a reminder to do the same again in three months.
- **Monitoring:** Set up some Amazon CloudWatch metrics alarms for any Amazon Lambda functions that react to outside (such as untrusted) input. This will let you know if resources are being wasted, or if there was some latent behavior in the function's code that's now being expressed. Set a reminder to do this for new functions.

Amazon Lambda is a serverless compute service that runs code in response to events and automatically manages the underlying compute resources.

Monitoring Amazon Lambda functions is included in the Stealthwatch Cloud entity modeling service. If you're using Stealthwatch Cloud already, you can make sure that the Stealthwatch Cloud policy for your account is up-to-date since this will let it use the AWS APIs for Amazon Lambda monitoring. If you're new to Stealthwatch Cloud, you'll be able to check on Amazon Lambda function activity shortly after you initiate monitoring.

For more information, please view Stealthwatch Cloud in the AWS Marketplace, or visit us at http://cisco.com/go/stealthwatch-cloud