Title:

Subhead: In this blog post, we will discuss how to use SQL and the tool Steampipe to find and enumerate the IP addresses, hostnames, and URLs from a customer's Amazon account. We will provide step-by-step instructions for using SQL queries and Steampipe to gather this information, and will explain how this information can be used to help define and manage the customer's cloud-network perimeter. Whether you are new to cloud computing or an experienced user, this post will provide valuable insights and information that can help you better understand and manage your Amazon cloud environment.

Post

In today's cloud-based world, it is essential for organizations to define and manage their cloud-network perimeter. This boundary defines the customer's external boundary in the cloud and is crucial for defining your External Attack Surface.

In this blog post, we will provide an overview of the importance of the cloud-network perimeter and will provide step-by-step instructions for enumerating the customer's cloud-network perimeter using Steampipe.

Resources that comprise a cloud-network perimeter can be broadly grouped into three categories: IP addresses, hostnames, and URLs.

1. IP addresses are numerical labels that are assigned to devices on a computer network. These labels are used to identify and locate devices on the network.
2. Hostnames are human-readable labels that are assigned to devices on a computer network. These labels are often used instead of IP addresses, as they are easier for people to remember and use. Hostnames are typically mapped to the corresponding IP address of a device using the Domain Name System (DNS). Hostnames can point to IP addresses inside the cloud or to on-premises networks.
3. URLs are web addresses that are used to access resources on the internet. These addresses typically consist of a combination of the hostname and the path to the specific resource being accessed. For example, the URL "https://steampipe.io/index.html" includes the hostname "steampipe.io" and the path to the resource "/index.html".

Each of these types of resources can be included in a cloud-network perimeter, and can be used to help identify and manage the boundary of a customer's cloud environment.

Enumerating IP Addresses using Steampipe

For the purposes of defining the customer-managed IP addresses that comprise the customer's cloud-network perimeter, we will primarily look at the Elastic Network Interfaces. While other resources like CloudFront, S3, and API Gateway also leverage IP Addresses, the network security protections of those services are the cloud provider's responsibility. This query will download a list of all of the public IP addresses that are tied to the customer's VPC.

```sql
select
	eni.association_public_ip AS public_ip
from
	aws_ec2_network_interface AS eni
where
	eni.association_public_ip is not Null;
```

Enumerating Hostnames using Steampipe

A DNS hostname is a human-readable label that is assigned to a device on a computer network. This label is used instead of the numerical IP address of the device, and is typically mapped to the corresponding IP address using the Domain Name System (DNS). Hostnames are often easier for people to remember and use than IP addresses, and can be used to access resources on the network using common internet protocols, such as HTTP and FTP. Examples of DNS hostnames include "www.steampipe.io" and "mail.steampipe.io".

In order to determine the DNS Hostnames used as part of your cloud perimeter, Steampipe can query all of the A records in your Route 53 Hosted Zones.

An A record is a type of DNS record that is used to map a hostname to the corresponding IP address of a device on a computer network. The A in A record stands for "address". This record is used to determine the IP address that is associated with a particular hostname, and allows devices on the network to communicate with each other using the hostname instead of the IP address. For example, if a DNS server has an A record that maps the hostname "www.steampipe.io" to the IP address "192.0.2.1", then a device on the network can access the resources at that IP address by using the hostname "www.steampipe.io" instead of the IP address.

To query all of the hostnames that point to A Records in your Route 53 Hosted Zones, use this SQL query:

```sql
select
  r.name as hostname,
  type,
  jsonb_array_elements_text(records) as resource_record
from
  aws_route53_zone as z,
  aws_route53_record as r
where r.zone_id = z.id
  and (type LIKE 'A' OR type LIKE 'CNAME')
  and z.private_zone=false
```

```
  and jsonb_pretty(records) not like '%dkim%'
  and jsonb_pretty(records) not like '%acm-validations.aws.%';
```

Much like with IP Addresses, you would want to review the list of IP addresses returned, and if permitted by the terms of service, scan these hostnames for exposed ports and services using nmap or Nessus. IP addresses belonging to a cloud provider should be assessed to ensure the cloud provider is performing proper security measures.

Note: The above query excludes private DNS for VPCs `z.private_zone=false` and excludes common CNAMEs needed for ACM and email validation.

Enumeration of the URLs using Steampipe

A URL, or uniform resource locator, is a web address that is used to access resources on the internet. URLs typically consist of a combination of the hostname and the path to the specific resource being accessed. For example, the URL "https://www.steampipe.io/index.html" includes the hostname "www.steampipe.io" and the path to the resource "/index.html".

A URL is used as part of an HTTP request in order to specify the location of the resource that the request is trying to access. When a client, such as a web browser, sends an HTTP request to a web server, the request includes the URL of the resource that the client is trying to access. The web server then uses the URL to determine the location of the requested resource, and sends a response back to the client with the content of that resource. This allows the client to access and display the requested resource, such as a web page or an image file.

URLs can be produced by a number of AWS services, including S3, CloudFront, API Gateway, and AWS Lambda.

Amazon Simple Storage Service (Amazon S3) is a cloud storage service that allows users to store and retrieve data from anywhere on the internet. It provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.

S3 Buckets exist as URLs on the public internet and can be accessed if the bucket is not properly secured. To get a list of all of the URLs for the *public* buckets in your cloud environment, you can use this query:

```sql
select
  'https://' || name || '.s3.' || region || '.amazonaws.com/' as url
from
  aws_s3_bucket
where
```

```
  bucket_policy_is_public is True;
```

Amazon CloudFront is a content delivery network (CDN) service that speeds up the delivery of static and dynamic web content, such as HTML, CSS, JavaScript, and images. It uses a global network of edge locations to cache content and deliver it to users with low latency, high transfer speeds, and high availability.

To request all of the URLs from CloudFront, this SQL query will return both the distribution name and any aliases that are part of that distribution:

```sql
select
  'https://' || domain_name as url
from
  aws_cloudfront_distribution
UNION ALL
select
  'https://' || jsonb_array_elements_text(aliases -> 'Items') as url
from
  aws_cloudfront_distribution;
```

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. It allows users to create APIs that access data and business logic from back-end services, such as Amazon S3 and AWS Lambda, and expose them to web and mobile applications.

API Gateways themselves do not have URLs, but when the customer creates a “stage” a URL is created. To get a list of all the API Gateway v2 URLs, this SQL query can be used:

```sql
select 'https://' || api_id || '.execute-api.' || region || '.amazonaws.com/' || stage_name as url
from aws_api_gatewayv2_stage;
```

API Gateway URLs typically have the following format:
https://[api-id].execute-api.[region].amazonaws.com/[stage]/[path]
where [api-id] is the ID of the API Gateway API, [region] is the AWS region where the API is deployed, [stage] is the stage of the API (such as "prod" or "dev"), and [path] is the path to the specific Lambda function being accessed. For example, the URL "https://abc123.execute-api.us-east-1.amazonaws.com/prod/hello" could be used to invoke a Lambda function named "hello" that is associated with the API Gateway API with ID "abc123", and is deployed in the "us-east-1" region.

AWS Lambda is a serverless computing platform that allows users to run code without provisioning or managing servers. It automatically scales and monitors the code, and provides a pay-as-you-go pricing model. This makes it easy for developers to build and deploy applications and microservices without worrying about infrastructure.

Lambda URLs are a new feature of AWS Lambda, released in April of 2022. AWS allows the creation of Lambda URLs that do not require authentication, therefore, these Lambda URLs should be considered part of your cloud-network perimeter. To find the Lambda URLs in your environment, you can use this Steampipe query:

```sql
select url_config ->> 'FunctionUrl' as url
from aws_lambda_function
where url_config is not Null;
```

Empowered with a list of all the exposed URLs in your organization, you can then setup a process to scan these using a number of web-focused Dynamic Application Security Testing (DAST) tools and scanners such as:

- URL Fuzzer
- Zed Attack Proxy
- StackHawk (commercial)
- dirsearch (Web path scanner)
- Aquatone
- Nikto2

Additionally the OWASP® Foundation maintains a full list of scanning tools that could be used.

If you are a larger organization that runs a bug bounty program, scanning your URLs for these “low-hanging fruit” is a quick and easy way to avoid payouts to researchers who use the same tools.

Steampipe Perimeter Mod

In addition to the above queries to extract the components of the cloud-network perimeter, Steampipe has the AWS Perimeter Mod that can scan your accounts for misconfigurations.

Conclusion

Using SQL and the tool Steampipe can be a powerful way to find and enumerate the IP addresses, hostnames, and URLs from a customer's Amazon account. Following the steps outlined in this blog post, you can gather this information to help define and manage the customer's cloud-network perimeter. This can help to ensure the security and integrity of your

data and applications, and prevent unauthorized access to your network resources. Whether you are new to cloud computing or an experienced user, this information can be valuable in helping you to better understand and manage your Amazon cloud environment.

The techniques we've shown here hopefully give you a better idea how you can enumerate your cloud perimeter, but everyone's situation is unique and you may find a solution that works better for you. If so, please let us know: we love to learn from our community!