

Lines in Comic Sans are prompts Chris wrote to ChatGPT

Lines in Courier are the results from ChatGPT

Lines in Arial are edits and additions written directly by Chris

Write a summary for a blog post where we discuss how to find and enumerate the IP Address, HostNames and URLs from a customer's amazon account using SQL and the tool Steampipe.

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 network cloud 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.

Let's define the customer's cloud network perimeter as all the resources that are accessible via the public internet that do not require authentication using cloud provider API access keys. This would include resources that are managed by the cloud provider and exist on the internet along with resources that are managed by the customer, have a public IP address and exist in the customer's VPC.

Write the introduction to a blog post about how to enumerate the customer's cloud network perimeter. The introduction should be cloud-provider agnostic, however the remaining blog post will focus on AWS technologies.

In today's cloud-based world, it is essential for organizations to properly define and manage their network cloud perimeter. This boundary, which separates a customer's network from the cloud provider's network, is crucial for ensuring the security and integrity of the customer's data and applications, and helps to prevent unauthorized access to the customer's network resources. In this blog post, we will provide an overview of the importance of the network cloud perimeter, and will provide step-by-step instructions for enumerating the customer's cloud network perimeter using ~~AWS~~

~~technologies~~ **Steampipe**. Whether you are new to cloud computing or an experienced user, this post will provide valuable insights and information that can help you to better understand and manage your cloud environment.

**Resources that comprise the cloud network perimeter fall into three categories: 1) IP Addresses, 2) Hostnames and 3) URLs. Please describe each of these.**

Resources that comprise a cloud network perimeter can be 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, and can be either static (permanently assigned to a device) or dynamic (temporarily assigned to a device).
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).
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://www.example.com/index.html"](https://www.example.com/index.html) includes the hostname ["www.example.com"](https://www.example.com) 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**

**Describe the activities a cloud customer should perform in order to secure public IP Addresses that exist in their cloud VPC.**

In order to secure public IP addresses that exist within a customer's cloud virtual private cloud (VPC), the customer should perform the following activities:

1. Identify the public IP addresses that are in use within the VPC. This can be done using **steampipe** and the queries we will show in a moment.

2. Implement network security measures, such as firewalls and virtual private networks (VPNs), to control and restrict access to the public IP addresses. These measures should be configured to allow only authorized traffic to pass through to the public IP addresses, and should be regularly reviewed and updated to ensure they remain effective.
3. Monitor the network traffic to and from the public IP addresses, and investigate any suspicious or unauthorized activity. This can be done using tools provided by the cloud provider, or by using third-party security monitoring solutions.
4. Regularly review and update the security measures that are in place to protect the public IP addresses, to ensure that they remain effective in preventing unauthorized access.

By following these steps, a cloud customer can help to ensure the security of their public IP addresses, and protect their data and applications from potential threats.

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 APIGateway also leverage IP Addresses, the network security protections are the responsibility of the cloud provider and not the customer.

```
```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

### Define a DNS Hostname

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.example.com" and "mail.example.com".

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.

**Explain what an A record is in DNS.**

In the Domain Name System (DNS), 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.example.com" 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.example.com" instead of the IP address.

To query all of the A Records from your Route 53 Hosted Zones, use this SQL Query:

```
```sql
select
  r.name as hostname
from
  aws_route53_zone as z,
  aws_route53_record as r
where
  r.zone_id = z.id
  and type LIKE 'A'
  and z.private_zone=false
```
```

**## Enumeration of the URLs using Steampipe**

**What is a URL, and how is a URL used as part of an HTTP Request?**

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.example.com/index.html" includes the hostname "www.example.com" 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.

**Briefly describe each of the above services.**

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 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;
'''

```

### What are Lambda URLs?

~~Lambda URLs are web addresses that are used to access AWS Lambda functions over the internet. These URLs are automatically generated by the Amazon API Gateway service when a Lambda function is associated with an API Gateway API, and allow the Lambda function to be invoked by sending an HTTP request to the URL. Lambda 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 ~~Lambda~~ URL

["https://abc123.execute-api.us-east-1.amazonaws.com/prod/hello"](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, and 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
```
```

Write a summary for a blog post where we discuss how to find and enumerate the IP Address, HostNames and URLs from a customer's amazon account using SQL and the tool Steampipe.

<results become the header/description>

**Make the above the conclusion to the blog post.**

In 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. By following the steps outlined in this blog post, you can gather this information and use it to help define and manage the customer's network cloud 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.



### **What are CIS Critical Security Controls one and two**

The CIS Critical Security Controls (CSC) are a set of 20 recommended actions that organizations can take to protect their systems and data from cyber attacks. CSC 1 and CSC 2 are the first two controls in the list, and are as follows:

1. CSC 1: Inventory and Control of Hardware Assets

This control recommends that organizations maintain an up-to-date inventory of all hardware assets, including laptops, desktops, servers, and mobile devices. This inventory should include information such as the make, model, and operating system of each device, as well as its current location and assigned owner. The inventory should be regularly reviewed and updated, and any changes to the hardware assets should be recorded and tracked.

2. CSC 2: Inventory and Control of Software Assets This control recommends that organizations maintain an up-to-date inventory of all software assets, including applications, operating systems, and patches. This inventory should include information such as the name, version, and licensing details of each software asset, as well as its current location and assigned owner. The inventory should be regularly reviewed and updated, and any changes to the software assets should be recorded and tracked.

Together, these two controls help organizations to gain visibility and control over their hardware and software assets, which can be important for ensuring the security and compliance of their systems and data.