Generating a strong pre-shared key

A pre-shared key (also called a shared secret or PSK) is used to authenticate the Cloud VPN tunnel to your peer (https://cloud.google.com/vpn/docs/concepts/overview#peer-definition) VPN gateway. As a security best practice, it's recommended that you generate a strong 32-character shared secret.

Generated for you

The random string below has been generated by your browser using the JavaScript snippet (#javascript) at the bottom of this page. It is 24 bytes from `Crypto.getRandomValues`, base64 encoded to create a 32 character PSK.

With this snippet, the private key stays securely in your browser. If you wish to generate it on your own system, use one of the Generation methods (#generation_methods) below.

The Regenerate button will generate a new, random PSK when clicked.

```
N+n4AIibkcXrNmhu7Mi0H2urRZIlJ5Kr
```

REGENERATE

Generation methods

Use the following methods to generate a strong 32-character shared secret.

Using OpenSSL to generate a shared secret

Run the following OpenSSL (https://www.openssl.org/) command on a Linux or macOS system to generate a shared secret:

```
openssl rand -base64 24
```

Using /dev/urandom to generate a shared secret

On Linux or macOS, you can also use `/dev/urandom` as a pseudorandom source to generate a shared secret:

- On Linux or macOS, you can send the random input to `base64`:

  ```
  head -c 24 /dev/urandom | base64
  ```

- You can pass the random input through a hashing function, like sha256:
  - On Linux:

    ```
    head -c 4096 /dev/urandom | sha256sum | cut -b1-32
    ```

  - On macOS:

    ```
    head -c 4096 /dev/urandom | openssl sha256 | cut -b1-32
    ```

Using JavaScript to generate a pre-shared key

You can also generate the pre-shared key directly in a doc page using JavaScript with the W3C Web Cryptography API (https://www.w3.org/TR/WebCryptoAPI/#Crypto-method-getRandomValues). This API uses the Crypto.getRandomValues() method (https://developer.mozilla.org/en-US/docs/Web/API/Crypto/getRandomValues), which provides a cryptographically sound way of generating a pre-shared key.

The code below will create an array of 24 random bytes, and then base64 encode those bytes to produce a random 32-character string.

```
var a = new Uint8Array(24);
```

```
window.crypto.getRandomValues(a);

console.log(btoa(String.fromCharCode.apply(null, a)));
```

What's next

- Learn about the basic concepts of Cloud VPN (https://cloud.google.com/vpn/docs/concepts/overview)
- See Advanced Configurations (https://cloud.google.com/vpn/docs/concepts/advanced) for information on high-availability, high-throughput scenarios, or multiple subnet scenarios.
- Create a custom Virtual Private Cloud network (https://cloud.google.com/vpc/docs/using-vpc#create-custom-network)
- Set up different types of Cloud VPN (https://cloud.google.com/vpn/docs/how-to/choosing-a-vpn)
- Maintain VPN tunnels and gateways (https://cloud.google.com/vpn/docs/how-to/maintaining-vpns)
- View logs and monitoring metrics (https://cloud.google.com/vpn/docs/how-to/viewing-logs-metrics)
- Get troubleshooting help (https://cloud.google.com/vpn/docs/support/troubleshooting)

Last updated December 4, 2019.