Lab 1: Lab Setup

Duration: 10 minutes

Each student should have received the lab workstation log in information from the instructor. This lab ensures that everyone can connect to the workstation, and verify that a Vault server is running so that vault commands can run against it.

- Task 1: Connect to the Student Workstation
- Task 2: Getting Help
- Task 3: Enable Audit Logging

Task 1: Connect to the Student Workstation

Step 1.1.1

SSH into your workstation using the provided credentials.

```
$ ssh <username>@<workstation_IP_address>
password: <password>
```

Depending on your machine setting, you may need to explicitly set the `PubKeyAuthentication` to `false`:

```
$ ssh -o PubKeyAuthentication=false -l training <workstation_public_IP_address>
```

When you are prompted, enter "yes" to continue connecting.

On a Windows, use SSH client such as PuTTY. On a Linux or Mac, use the Terminal to SSH into your workstation.

Alternatively, launch a web browser and enter:

```
http://<workstation_IP_address>/wetty
```

When you are prompted, enter the username and password provided by your instructor.

Step 1.1.2

Run the following command to check the Vault server status:

```
$ vault status
```

```
Key             Value
---             -----
Seal Type       shamir
Sealed          false
Total Shares    1
Threshold       1
Version         0.11.3
Cluster Name    vault-cluster-c78ba77b
Cluster ID      3e9a12c3-19f2-f20f-7f65-46bc02f0ac19
HA Enabled      falsee
```

Notice that the server has been unsealed.

```
Sealed          false
```

The server has been started in *dev* mode. When you start a Vault server in dev mode, it automatically unseals the server.

Step 1.1.3

Authenticate with Vault using the root token:

```
$ vault login root
```

Expected output:

```
Success! You are now authenticated. The token information displayed below is
already stored in the token helper. You do NOT need to run "vault login" again.
Future Vault requests will automatically use this token.

Key                  Value
---                  -----
token                root
token_accessor       7993db51-1c35-ecc7-6293-6c4279230299
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
```

NOTE: For the purpose of training, we will start slightly insecure and login using the root token. Also, the Vault server is running in *dev* mode.

Task 2: Getting Help

Step 1.2.1

Execute the following command to display available commands:

```
$ vault help
```

Or, you can use short-hand:

```
$ vault -h
```

Step 1.2.2

Get help on vault server commands:

```
$ vault server -h
```

The help message explains how to start a server and its available options.

As you verified at Step 1.1.2, the Vault server is already running. The server was started using the command described in the help message: `vault server -dev -dev-root-token-id="root"`

Step 1.2.3

Get help on the `read` command:

```
$ vault read -h
```

This command reads a secret from a given path.

Step 1.2.4

To get help on the API, the help command becomes `path-help` instead:

```
$ vault path-help sys/policy
```

The key/value secret backend is mounted on `secret/` path.

Task 3: Enable Audit Logging

Audit backend keeps a detailed log of all requests and responses to Vault. Sensitive information is obfuscated by default (HMAC). Prioritizes safety over availability.

Step 1.3.1

Change directory into `/workstation/vault`

```
$ cd /workstation/vault
```

Step 1.3.2

Get help on the `audit enable` command:

```
$ vault audit enable -h
```

Step 1.3.3

Let's write audit log in current working directly so that you can inspected as you go through other labs.

Execute the following command to enable audit logging:

```
$ vault audit enable file \
    file_path=/workstation/vault/audit.log
```

Expected output:

```
Success! Enabled the file audit device at: file/
```

Step 1.3.4

You can verify that the audit log file is generated:

```
$ sudo cat audit.log
```

However, at this point, its content is hard to read. You can pipe the output with jq tool.

```
$ sudo cat audit.log | jq
...
  "request": {
    "id": "0f2fb5fd-6a74-f425-9537-2c6d4283b7b8",
    "operation": "read",
    "client_token": "hmac-sha256:85a4130cf4527b8bc5...",
    "client_token_accessor": "hmac-sha256:7dcfaabb1c...",
    "path": "secret/company",
    "data": null,
    "policy_override": false,
}
...
```

Sensitive information such as client token is obfuscated **by default** (HMAC).

Optional

Often times, the logged information can help you understand what is going on with each command. Invoke the following command to generate a raw log:

```
# Remove the old log
$ vault audit disable file
$ rm audit.log

$ vault audit enable file file_path=/workstation/vault/audit.log \
        log_raw=true
```

If you want to tail the log as you go through hands-on labs, you can open another terminal, and run the following command:

```
$ sudo tail -f audit.log | jq
```

End of Lab 1