hashicorp / **vault-ssh-helper**

Vault SSH Agent is used to enable one time keys and passwords

138 commits **3** branches **5** releases **8** contributors MPL-2.0

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

jefferai Bump travis.yml Latest commit `f429a92` on Sep 12

.github	Add issue template	a year ago
helper	Added failure logs	a year ago
scripts	Update Dockerfile	a year ago
test-fixtures	vsh: Tests to ensure that OTPs should be of UUID format	2 years ago
vendor	Update deps, switch to dep	a year ago
.gitignore	Add build and dist scripts	3 years ago
.travis.yml	Bump travis.yml	2 months ago
CHANGELOG.md	changelog++	a year ago
Gopkg.lock	Update deps, switch to dep	a year ago
Gopkg.toml	Update deps, switch to dep	a year ago
LICENSE	Initial commit	3 years ago
Makefile	removing go get dependencies (#24)	2 years ago
README.md	Update readme	a year ago
main.go	make help more safe for pam exec module by exiting with 1 when help i…	a year ago
main_test.go	vsh: Tests to ensure that OTPs should be of UUID format	2 years ago
version.go	Bump version	a year ago

README.md

vault-ssh-helper build passing

Please note: We take Vault's security and our users' trust very seriously. If you believe you have found a security issue in Vault, *please responsibly disclose* by contacting us at security@hashicorp.com.

`vault-ssh-helper` is a counterpart to HashiCorp Vault's SSH backend. It allows a machine to consume One-Time-Passwords (OTP) created by Vault servers by allowing them to be used as client authentication credentials at SSH connection time.

All of the remote hosts that belong to the SSH backend's OTP-type roles will need this helper installed. In addition, each host must have its SSH configuration changed to enable keyboard-interactive authentication and redirect its client authentication responsibility to `vault-ssh-helper`.

Vault-authenticated users contact the Vault server and retrieve an OTP issued for a specific username and IP address. While establishing an SSH connection to the host, the `vault-ssh-helper` binary reads the OTP from the password prompt and sends it to the Vault server for verification. Client authentication is successful (and the SSH connection allowed) only if the Vault server verifies the OTP. True to its name, once the OTP has been used a single time for authentication, it is removed from Vault and cannot be used again.

`vault-ssh-helper` is not a PAM module, but it does the job of one. `vault-ssh-helper` 's binary is run as an external command using `pam_exec.so` with access to the entered password (in this case, the issued OTP). Successful execution and exit of this command is a PAM 'requisite' for authentication to be successful. If the OTP is not validated, the binary exits with a non-zero status and authentication fails.

PAM modules are generally shared object files; rather than writing and maintaining a PAM module in C, `vault-ssh-helper` is written in Go and invoked as an external binary. This allows `vault-ssh-helper` to be contained within one code base with known, testable behavior. It also allows other authentication systems that are not PAM-based to invoke `vault-ssh-helper` and take advantage of its capabilities.

Usage

```
vault-ssh-helper [options]
```

Options

Option	Description
`verify-only`	Verifies that `vault-ssh-helper` is installed correctly and is able to communicate with Vault.
`config`	The path to the configuration file. Configuration options are detailed below.
`dev`	`vault-ssh-helper` communicates with Vault with TLS disabled. This is NOT recommended for production use. Use with caution.

Download vault-ssh-helper

Download the latest version of `vault-ssh-helper` at releases.hashicorp.com.

Build and Install

You'll first need Go installed on your machine (version 1.8+ is required).

Install `Go` on your machine and set `GOPATH` accordingly. Clone this repository into $GOPATH/src/github.com/hashicorp/vault-ssh-helper. Install all of the dependant binaries like `godep` , `gox` , `vet` , etc. by bootstrapping the environment:

```
$ make updatedeps
```

Build and install `vault-ssh-helper` :

```
$ make
$ make install
```

Follow the instructions below to modify your SSH server configuration, PAM configuration and `vault-ssh-helper` configuration. Check if `vault-ssh-helper` is installed and configured correctly and also is able to communicate with Vault server properly. Before verifying `vault-ssh-helper` , make sure that the Vault server is up and running and it has mounted the SSH backend. Also, make sure that the mount path of the SSH backend is properly updated in `vault-ssh-helper` 's config file:

```
$ vault-ssh-helper -verify-only -config=<path-to-config-file>
Using SSH Mount point: ssh
vault-ssh-helper verification successful!
```

If you intend to contribute to this project, compile a development version of `vault-ssh-helper` using `make dev` . This will put the binary in the `bin` and `$GOPATH/bin` folders.

```
$ make dev
```

If you're developing a specific package, you can run tests for just that package by specifying the `TEST` variable. For example below, only `helper` package tests will be run.

```
$ make test TEST=./helper
...
```

If you intend to cross compile the binary, run `make bin` .

`vault-ssh-helper` Configuration

[Note]: This configuration is applicable for Ubuntu 14.04. SSH/PAM configurations differ with each platform and distribution.

`vault-ssh-helper` 's configuration is written in HashiCorp Configuration Language (HCL). By proxy, this means that `vault-ssh-helper` 's configuration is JSON-compatible. For more information, please see the HCL Specification.

Properties

Property	Description
`vault_addr`	[Required] Address of the Vault server.
`ssh_mount_point`	[Required] Mount point of SSH backend in Vault server.
`ca_cert`	Path of a PEM-encoded CA certificate file used to verify the Vault server's TLS certificate. `-dev` mode ignores this value.
`ca_path`	Path to directory of PEM-encoded CA certificate files used to verify the Vault server's TLS certiciate. `-dev` mode ignores this value.
`tls_skip_verify`	Skip TLS certificate verification. Use with caution.
`allowed_roles`	List of comma-separated Vault SSH roles. The OTP verification response from the server will contain the name of the role against which the OTP was issued. Specify which roles are allowed to login using this configuration. Set this to `*` to allow any role to perform a login.
`allowed_cidr_list`	List of comma-separated CIDR blocks. If the IP used by the user to connect to the host is different than the address(es) of the host's network interface(s) (for instance, if the address is NAT-ed), then `vault-ssh-helper` cannot authenticate the IP. In these cases, the IP returned by Vault will be matched with the CIDR blocks in this list. If it matches, the authentication succeeds. (Use with caution)

Sample `config.hcl` :

```
vault_addr = "https://vault.example.com:8200"
ssh_mount_point = "ssh"
ca_cert = "/etc/vault-ssh-helper.d/vault.crt"
tls_skip_verify = false
allowed_roles = "*"
```

PAM Configuration

Modify the `/etc/pam.d/sshd` file as follows; each option will be explained below.

```
#@include common-auth
auth requisite pam_exec.so quiet expose_authtok log=/tmp/vaultssh.log /usr/local/bin/vault-ssh-helper -
config=/etc/vault-ssh-helper.d/config.hcl
auth optional pam_unix.so not_set_pass use_first_pass nodelay
```

First, the previous authentication mechanism `common-auth` , which is the standard Linux authentication module, is commented out, in favor of using our custom configuration.

Next the authentication configuration for `vault-ssh-helper` is set.

Keyword	Description
`auth`	PAM type that the configuration applies to.
`requisite`	If the external command fails, the authentication should fail.

Keyword	Description
`pam_exec.so`	PAM module that runs an external command (`vault-ssh-helper`).
`quiet`	Suppress the exit status of `vault-ssh-helper` from being displayed.
`expose_authtok`	Binary can read the password from stdin.
`log`	Path to `vault-ssh-helper` 's log file.
`vault-ssh-helper`	Absolute path to `vault-ssh-helper` 's binary.
`config`	The path to `vault-ssh-helper` 's config file.

The third line works around a bug between some versions of `pam_exec.so` and `vault-ssh-helper` that causes a successful authentication from `vault-ssh-helper` to fail due to some resources not being properly released. Because it is marked as optional, it is essentially a no-op that ensures that PAM cleans up successfully, avoiding the bug.

Option	Description
`auth`	PAM type that the configuration applies to.
`optional`	If the module fails, authentication does not fail (but if the OTP was invalid, we will have already failed previously).
`pam_unix.so`	Linux's standard authentication module.
`not_set_pass`	Module should not be allowed to set or modify passwords.
`use_first_pass`	Do not display password prompt again. Use the password from the previous module.
`nodelay`	Avoids the induced delay after entering a wrong password.

SSHD Configuration

Modify the `/etc/ssh/sshd_config` file. Note that for many distributions these are the default options; you may not need to set them explicitly but should verify their values if not.

```
ChallengeResponseAuthentication yes
UsePAM yes
PasswordAuthentication no
```

Option	Description
`ChallengeResponseAuthentication yes`	[Required] Enable challenge response (keyboard-interactive) authentication.
`UsePAM yes`	[Required] Enable PAM authentication modules.
`PasswordAuthentication no`	Disable password authentication.