HashiCorp
Vault

Vault Self-Managed Lunch and Learn - HCDiag

<INSERT DATE>

HCDiag

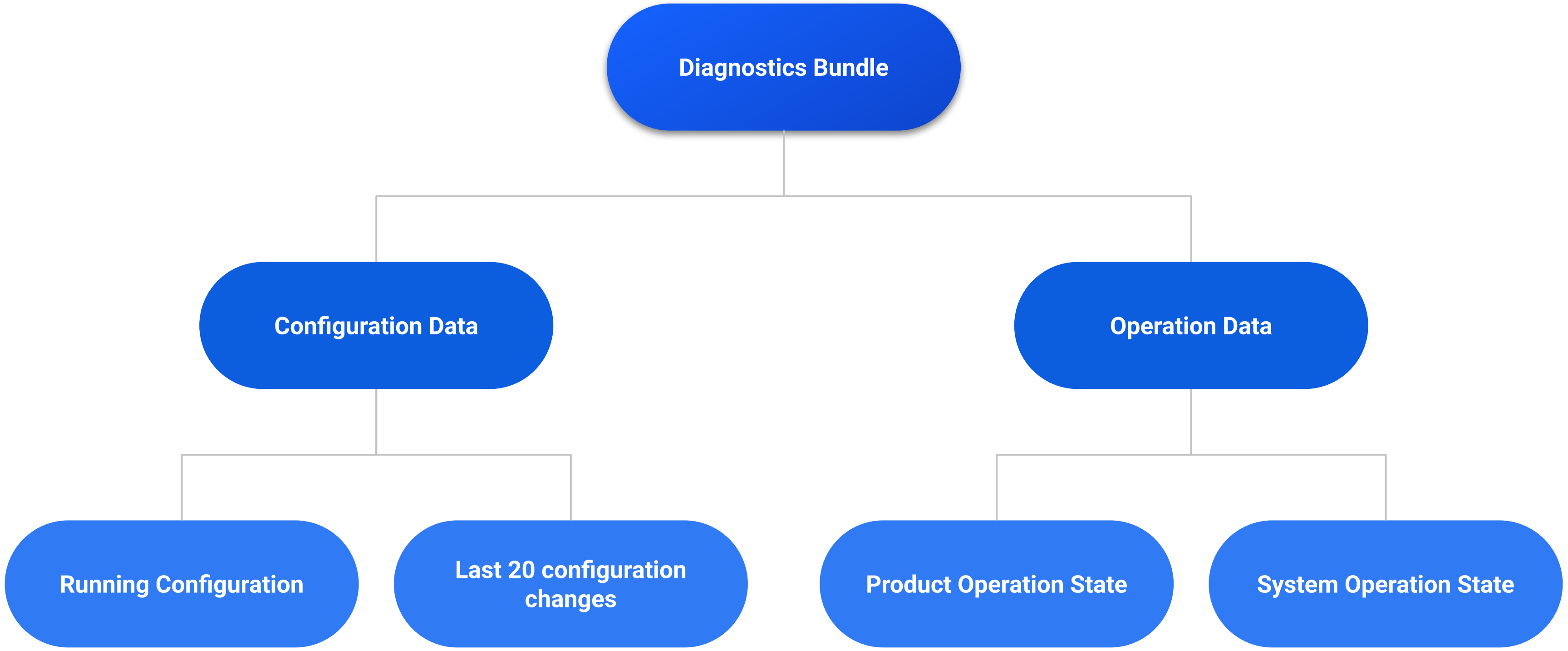
What is it?

HCDiag is a CLI based tool that can be used to simplify the collection of data during the troubleshooting process. When HCDiag is run it will collect system and product information, once complete it will create a compressed bundle of all the data captured.

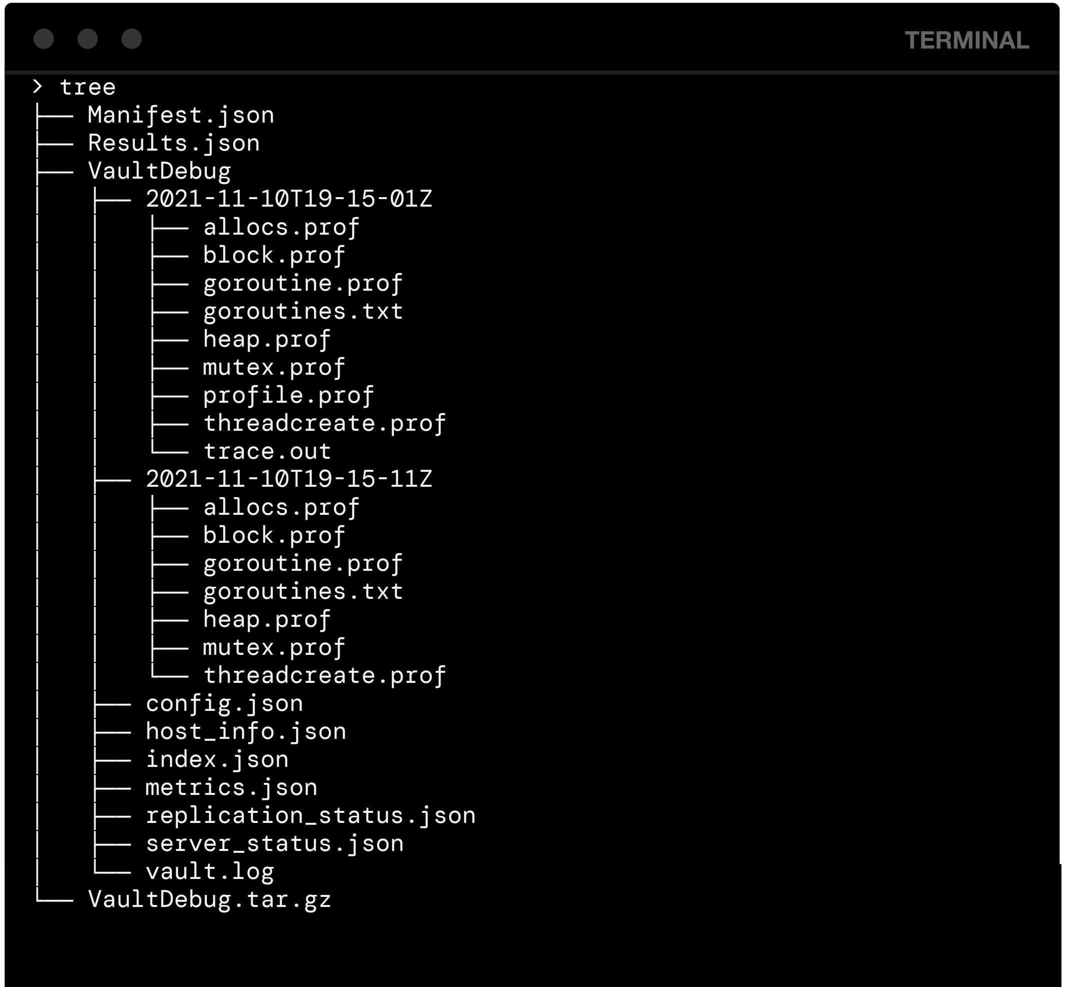
Supported Products:

- Vault
- Terraform
- Consul
- Nomad

Bundle Contents

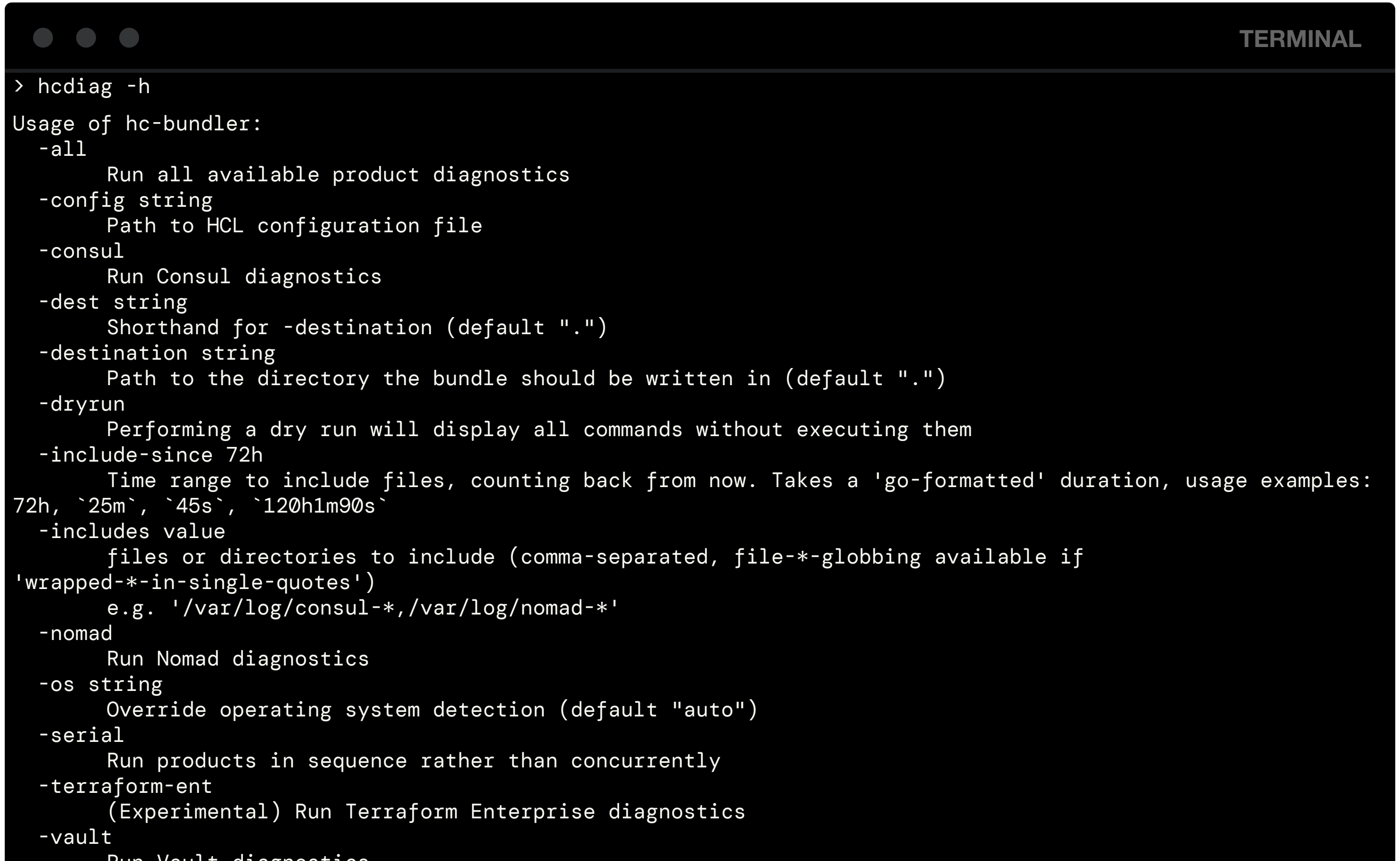


Bundle Contents


TERMINAL

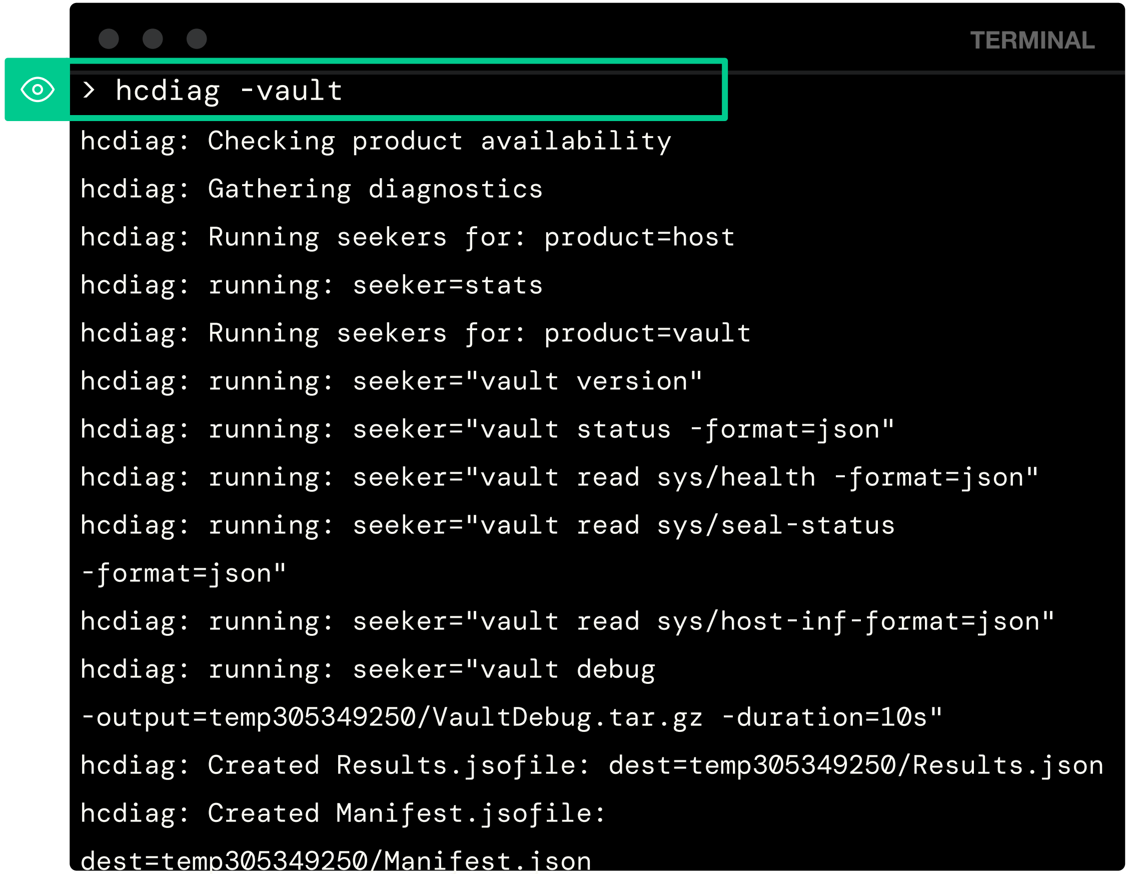
```
> tree
├── Manifest.json
├── Results.json
├── VaultDebug
│   ├── 2021-11-10T19-15-01Z
│   │   ├── allocs.prof
│   │   ├── block.prof
│   │   ├── goroutine.prof
│   │   ├── goroutines.txt
│   │   ├── heap.prof
│   │   ├── mutex.prof
│   │   ├── profile.prof
│   │   ├── threadcreate.prof
│   │   └── trace.out
│   ├── 2021-11-10T19-15-11Z
│   │   ├── allocs.prof
│   │   ├── block.prof
│   │   ├── goroutine.prof
│   │   ├── goroutines.txt
│   │   ├── heap.prof
│   │   ├── mutex.prof
│   │   └── threadcreate.prof
│   ├── config.json
│   ├── host_info.json
│   ├── index.json
│   ├── metrics.json
│   ├── replication_status.json
│   ├── server_status.json
│   └── vault.log
└── VaultDebug.tar.gz
```

HCDiag Usage


TERMINAL

```
> hcdiag -h
Usage of hc-bundler:
  -all
        Run all available product diagnostics
  -config string
        Path to HCL configuration file
  -consul
        Run Consul diagnostics
  -dest string
        Shorthand for -destination (default ".")
  -destination string
        Path to the directory the bundle should be written in (default ".")
  -dryrun
        Performing a dry run will display all commands without executing them
  -include-since 72h
        Time range to include files, counting back from now. Takes a 'go-formatted' duration, usage examples:
72h, `25m`, `45s`, `120h1m90s`
  -includes value
        files or directories to include (comma-separated, file-*-globbing available if
'wrapped-*-in-single-quotes')
        e.g. '/var/log/consul-*,/var/log/nomad-*'
  -nomad
        Run Nomad diagnostics
  -os string
        Override operating system detection (default "auto")
  -serial
        Run products in sequence rather than concurrently
  -terraform-ent
        (Experimental) Run Terraform Enterprise diagnostics
  -vault
        Run Vault diagnostics
```

Collecting a Vault Bundle



```
> hcdiag -vault
hcdiag: Checking product availability
hcdiag: Gathering diagnostics
hcdiag: Running seekers for: product=host
hcdiag: running: seeker=stats
hcdiag: Running seekers for: product=vault
hcdiag: running: seeker="vault version"
hcdiag: running: seeker="vault status -format=json"
hcdiag: running: seeker="vault read sys/health -format=json"
hcdiag: running: seeker="vault read sys/seal-status
-format=json"
hcdiag: running: seeker="vault read sys/host-inf-format=json"
hcdiag: running: seeker="vault debug
-output=temp305349250/VaultDebug.tar.gz -duration=10s"
hcdiag: Created Results.jsofile: dest=temp305349250/Results.json
hcdiag: Created Manifest.jsofile:
dest=temp305349250/Manifest.json
hcdiag: Compressed and archived output file: dest=.
```

Extending HCDiag

Config files

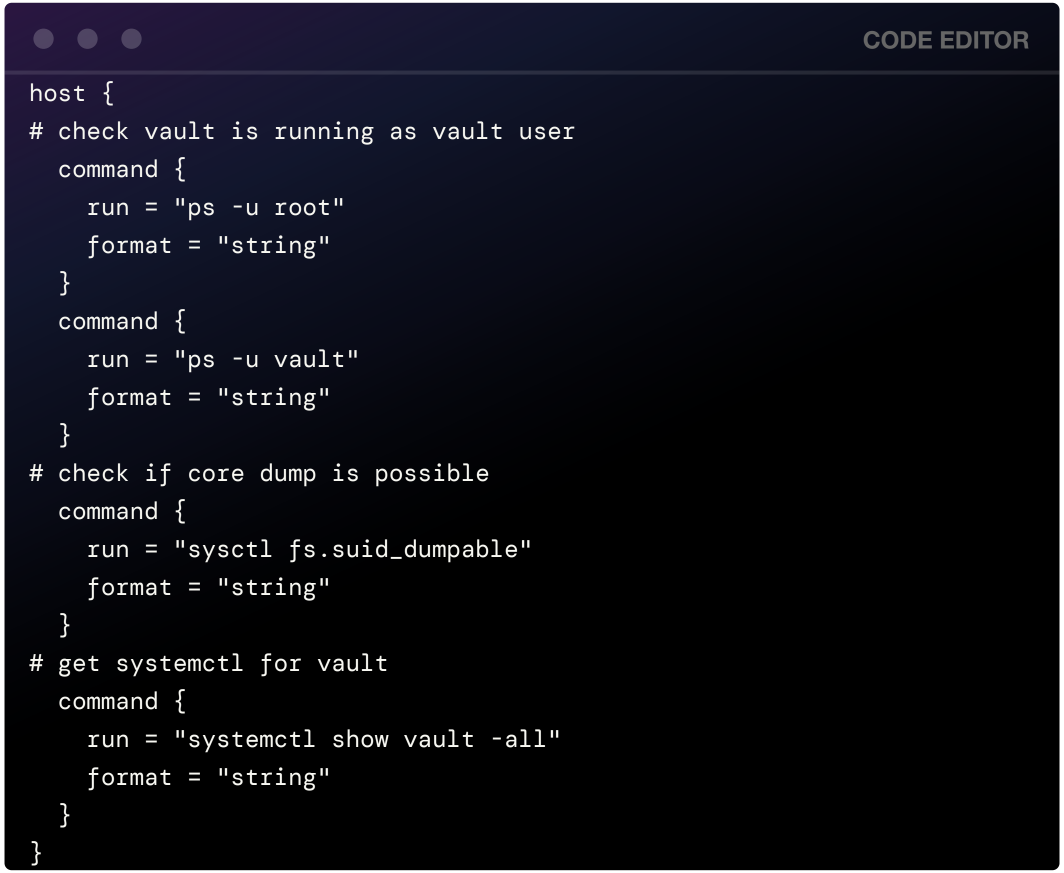
You can configure hcdiag behavior with a HashiCorp Configuration Language (HCL) formatted file. If you examine the Results.json file under the relevant product key, such as vault, you can find the commands that are executed.

Seekers

In your custom configuration file you can define seekers that can execute commands to gather additional information.

Seeker Configuration

Custom seeker to check for Vault production hardening

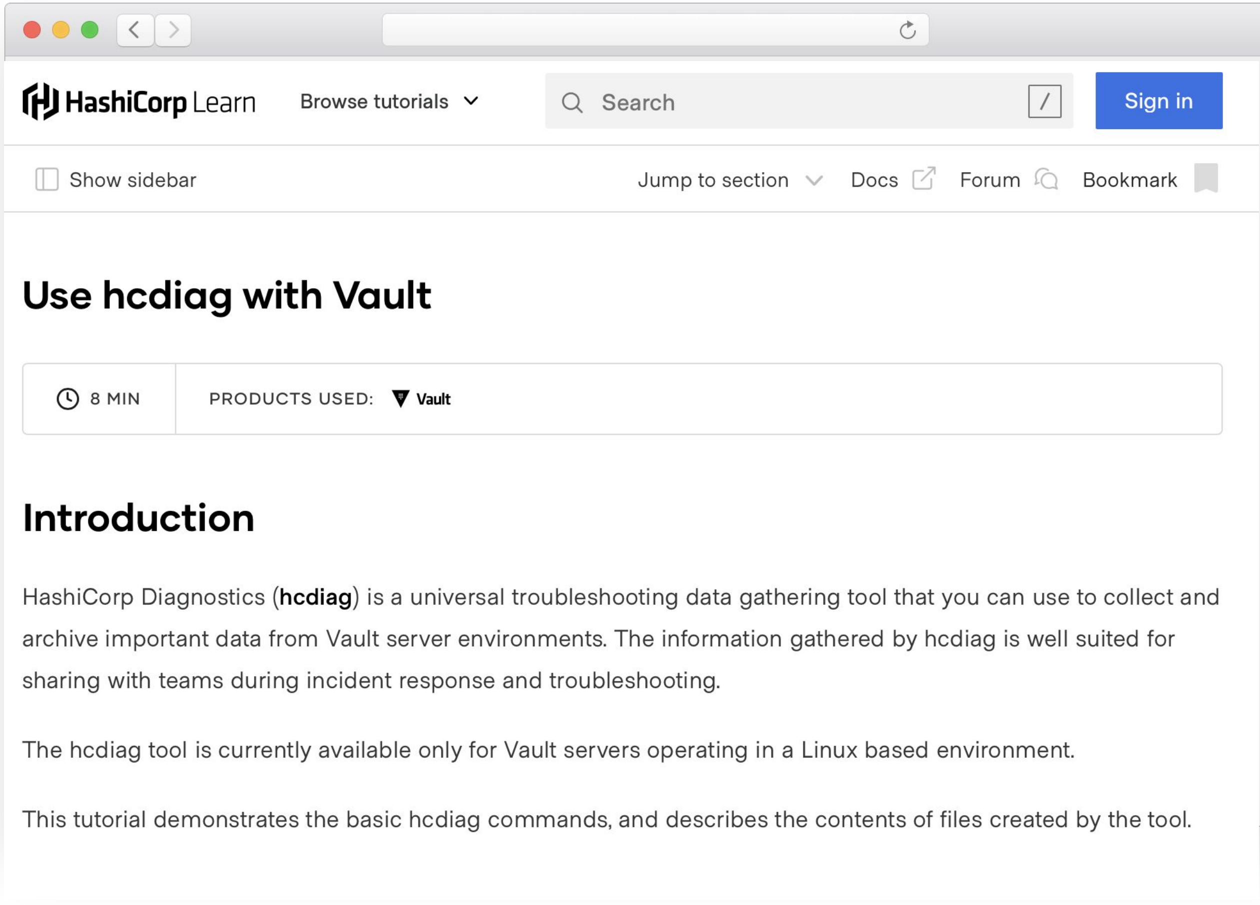


```
host {
# check vault is running as vault user
  command {
    run = "ps -u root"
    format = "string"
  }
  command {
    run = "ps -u vault"
    format = "string"
  }
# check if core dump is possible
  command {
    run = "sysctl fs.suid_dumpable"
    format = "string"
  }
# get systemctl for vault
  command {
    run = "systemctl show vault -all"
    format = "string"
  }
}
```

Learn Guide

Try hcdiag with Vault inside HashiCorp Docker image before testing in your environment.

[Use hcdiag with Vault]



Use hcdiag with Vault

8 MIN | PRODUCTS USED: Vault

Introduction

HashiCorp Diagnostics (**hcdiag**) is a universal troubleshooting data gathering tool that you can use to collect and archive important data from Vault server environments. The information gathered by hcdiag is well suited for sharing with teams during incident response and troubleshooting.

The hcdiag tool is currently available only for Vault servers operating in a Linux based environment.

This tutorial demonstrates the basic hcdiag commands, and describes the contents of files created by the tool.

02 HCDIAG

Demo

03 HCDIAG

Q & A

Thank You

customer.success@hashicorp.com
www.hashicorp.com