Section 5

HashiCorp Sentinel for Vault

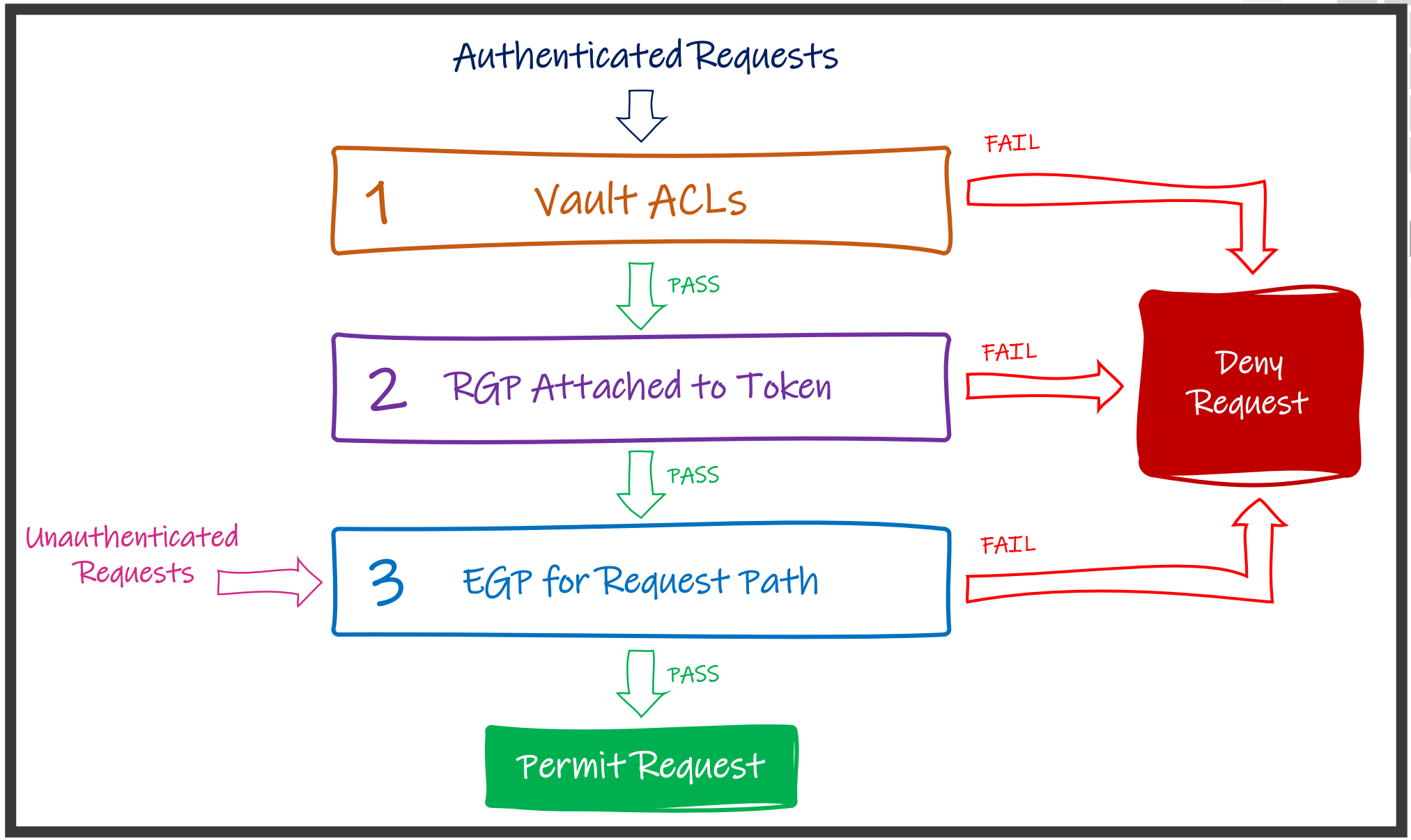
What is Sentinel?

- Policy as Code providing additional access control to Vault
- Featured in all of HashiCorp Enterprise offerings:
  - Vault, Terraform, Consul, & Nomad
- Expands Vault policies from only ACLs to also include:
  - **RGPs – Role Governing Policies**
    - policies tied to specific tokens, entities, or groups
  - **EGPs - Endpoint Governing Policies**
    - policies associated with particular paths
- Similar to ACLs, root tokens are not subject to Sentinel policy checks

What is Sentinel?

- Set using the sys/policies/rgp and sys/policies/egp
  - ***Note***: When moving from open-source to Enterprise, the path for policies change
  - **Open-source & Enterprise** = `sys/policy`
  - **Enterprise Governance & Policy** = `sys/policies` (because we now have acl, rgp, and egp)
- ACL Names ≠ RGP Names – both are assigned to tokens
- Enforcement levels for Sentinel policies can be set as:
  - Advisory
  - Soft Mandatory
  - Hard Mandatory
- Policy-overrides can be used with soft mandatory enforcement

Policy Evaluation



Sentinel Use Cases for Vault

- Validation of input data
- Limiting access from specific CIDR or IP address
- Disallow certain configurations in Vault
- Ensure access only during business hours and/or workdays
- Deny all previously created tokens

Examples

Restrict Request by IP Address

```
import "sockaddr"
import "strings"

# Expect requests to come only from our IP range or address
cidrcheck = rule {
    sockaddr.is_contained(request.connection.remote_addr, "10.10.10.10/32")
}

main = rule {
    cidrcheck
}
```

Examples

Restrict Read Request to Specific Path by IP Address

```
import "sockaddr"
import "strings"

# Only care about create, update, and delete operations against secret path
precond = rule {
    request.operation in [“write"] and
    strings.has_prefix(request.path, “aws/sts/production")
}

# Requests to come only from our private IP range
cidrcheck = rule {
    sockaddr.is_contained(request.connection.remote_addr, "10.10.0.0/16")
}

# Check the precondition before execute the cidrcheck
main = rule when precond {
    cidrcheck
}
```

Examples

Restrict Request by Time and Day

```
import "time"

# Expect requests to only happen during workdays (Monday through Friday)
workdays = rule {
   time.now.weekday > 0 and time.now.weekday < 6
}

# Expect requests to only happen during work hours (7:00 am - 6:00 pm)
workhours = rule {
   time.now.hour > 7 and time.now.hour < 18
}

precond = rule {
   request.operation in ["read"] and
   strings.has_prefix(request.path, "secret/")
}

main = rule when precond {
   workdays and workhours
}
```

Examples

Permit Only Authenticated Requests with Tokens Created in the Year 2020

```
import "time"

main = rule when not request.unauthenticated {
        time.load(token.creation_time).unix >
            time.load("2020-01-01T0:00:00Z").unix
}
```

Demo

HashiCorp Sentinel for Vault

Create a Sentinel Policy

Apply a Policy to a Vault Prefix (path)

Test the Policy