**Policy as Code Documentation**

**Networking Sentinel Policies**

3.6 Ensure That SSH Access Is Restricted from the Internet.

***Sentinel Policy Name:***

3.6 Ensure That SSH Access Is Restricted from the Internet.

***Category:***

* + Networking

***Description of Policy:***

* + In this policy, we are creating a rule in firewall to ensure 2 things:
    - The Port number in a firewall rule is not equal to 22 (SSH) and Action is not set to Allow
    - The IP Ranges is not equal to 0.0.0.0/0 (or other public CIDR range) under Source filters.

***Sentinel Policy Restriction:***

* + - If port 22 (SSH) is mentioned in the “allowed” then the source range CAN NOT be from the public (“0.0.0.0/0”, “::0”, “/0”, “0.0.0.0”).

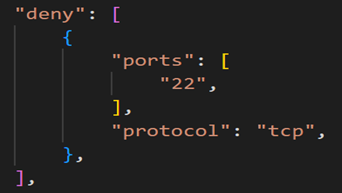
***Terraform attributes:***

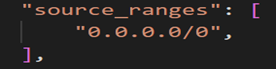
* + Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/compute_firewall>
  + Filter: 1. source\_ranges 2. Ports

* + ***Test cases:***

**Pass cases**

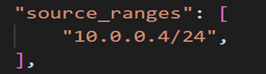
1. SSH Port 22 is denied, when the traffic is coming from the public internet (0.0.0.0/0)





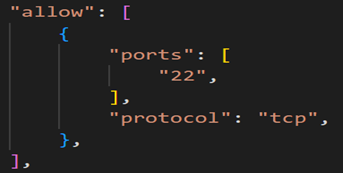
2. When the inbound traffic is not coming from the public internet, port 22 is allowed.

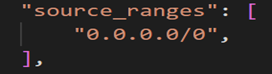




**Fail case:**

1. Allowing the port 22, when the inbound traffic is coming from the public internet





3.9 Ensure no https or SSL proxy load balances permit SSL policies with weak ciphers

***Sentinel Policy Name:***

* + - 3.9 Ensure no https or SSL proxy load balances permit ssl policies with weak ciphers

***Category:***

* + - Networking

***Description of Policy:***

* + Secure Sockets Layer (SSL) policies determine what port Transport Layer Security (TLS) features clients are permitted to use when connecting to load balancers. To prevent usage of insecure features, SSL policies should use (a) at least TLS 1.2 with the MODERN profile; or (b) the RESTRICTED profile, because it effectively requires clients to use TLS 1.2 regardless of the chosen minimum TLS version; or (3) a CUSTOM profile that does not support any of the following features:
    - TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256
    - TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384
    - TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
    - TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA
    - TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

***Sentinel Policy Restriction:***

The policy checks whether target HTTPS proxy or SSL proxy are created and if ssl policies named for either of the proxy are created with the following conditions:

* + at least TLS 1.2 with the MODERN profile
  + the RESTRICTED profile, because it effectively requires clients to use TLS 1.2 regardless of the chosen minimum TLS version
  + a CUSTOM profile that does not support any of the following features:
    - TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256
    - TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384
    - TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
    - TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA
    - TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

***Terraform attributes:***

* + - Provider Ref:[google\_compute\_ssl\_policy | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/compute_ssl_policy)
      * [google\_compute\_target\_https\_proxy | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/compute_target_https_proxy)
      * [google\_compute\_target\_ssl\_proxy | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/compute_target_ssl_proxy)
    - Terraform attribute:
      * target\_ssl\_proxy resource
        + ssl\_policy
      * target\_https\_proxy resource
        + ssl\_policy
      * ssl\_policy resource
        + min\_tls\_version
        + Profile
        + custom\_features
    - There is also a check if the SSL policy is call in target\_https\_proxy or target\_ssl\_proxy resource for the test to run properly.

* + ***Test cases:***

**Pass cases**

1. In pass case, if the ssl policy as one of the 3 below, then the pass case will always pass:

SSL policies should be one of the following:

* + at least TLS 1.2 with the MODERN profile

Screenshot of argument in mock file

" 1 _ policy. mod 
" address" : " google_compute 
"change" • 
"actions" : 
create" 
" after 
"custom features": 
"description' . 
min t1s version 
name : 
"profile" • 
"timeouts" : 
SSI -policyl" 
ssl_policy. mod_ssl-policyl " , 
null, 
null, 
"TLS 1 2", 
"nonprod - ssl-policy" , 
"MODERN" , 
null, 

* + the RESTRICTED profile, because it effectively requires clients to use TLS 1.2 regardless of the chosen minimum TLS version

Screenshot of argument in mock file

" google_compute_ssl_policy. res 
" address" . " google_compute 
"change" • 
"actions" : 
create" 
" after 
"custom features": 
"description' . 
min t1s version 
name : 
"profile" • 
"timeouts" : 
SSI -policy2" 
ssl_policy. res_ss1-p01icy2" 
null, 
null, 
"TLS 1 0", 
'I production-ssl-policy" 
" RESTRICTED" , 
null, 

* + a CUSTOM profile that does not support any of the following features:
    - TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256
    - TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384
    - TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
    - TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA
    - TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

Screenshot of argument in mock file

" google_compute_ssl_policy. custom_ss1-p01icy3" • 
"address" . 
" google_compute_ssl_policy. custom_ssl -policy3 " , 
"change" • 
actions" : 
create" 
" after 
"custom_features" : [ 
"TLS ECDHE ECDSA WITH AES 256 SHA384", 
"TLS ECDHE RSA WITH AES 256 GCM SHA384", 
"description" 
min t1s version 
name : 
"profile" : 
"timeouts" : 
null, 
"TLS 1 2", 
" custom-ssl-policy" 
"CUSTOM" , 
null, 

**Fail case:**

1. In pass case, if the ssl policy is one of the 3 below, then the pass case will always fail:

SSL policies will be one of the following:

* + Not at least TLS 1.2 with the MODERN profile

Screenshot of argument in mock file

" 1 _ policy. mod 
SSI -policyl" 
"address" . 
" google_compute 
ssl_policy. mod_ssl-policyl " , 
"change" • 
actions" : 
create" 
" after 
"custom features": 
null, 
"description' . 
null, 
min t1s version 
"TLS 1 0", 
"nonprod - ssl-policy" , 
name : 
"profile" : 
"MODERN" , 
"timeouts" : 
null, 
"after_unknown" : { 
"creation_timestamp" : true, 
"enabled features" : 
true, 
"fingerprint" • 
true, 
'lid" : 
true, 
"project" • 
true, 
"self link": 
true, 

* + NOT the RESTRICTED profile, because it effectively requires clients to use TLS 1.2 regardless of the chosen minimum TLS version

Screenshot of argument in mock file

"google_compute_ssl_policy. res 
SSI -policy2" 
" address" : " google_compute 
ssl_policy. res_ss1-p01icy2" 
"change" • 
"actions" : 
create" 
" after 
"custom features": 
null, 
"description' . 
null, 
min t1s version 
"TLS 1 0", 
'I production-ssl-policy" 
name : 
"profile" • 
"COMPATIBLE" 
"timeouts" : 
null, 
"after_unknown" : { 
"creation_timestamp" : true, 
"enabled features" : 
true, 
"fingerprint" • 
true, 
'lid" : 
true, 
"project" • 
true, 
"self link": 
true, 

* + a CUSTOM profile that does support any of the following features:
    - TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256
    - TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384
    - TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
    - TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA
    - TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

Screenshot of argument in mock file

" google_compute_ssl_policy. custom_ss1-p01icy3" • 
"address" : " custom_ss1-p01icy3" , 
"change" • 
"actions" : 
create" 
" after 
custom 
features" : 
"TLS 
"TLS 
"TLS 
"TLS 
"TLS 
RSA WITH 
RSA WITH 
RSA WITH 
RSA WITH 
RSA WITH 
AES 128 GCM SHA256", 
AES 256 GCM SHA384", 
AES 128 CBC SHA", 
AES 256 CBC SHA", 
3DES EDE CBC SHA" , 
"description " . 
min t1s version 
name : 
"profile" • 
"timeouts" : 
null, 
"TLS 1 2", 
" custom-ssl-policy" 
"CUSTOM" , 
null, 