

Instantly share code, notes, and snippets.

harshavardhana / leak.go

Created 6 months ago

☆ Star

<> Code  Revisions 1  Stars 2

 leak.go

```
1 package main
2
3 import (
4     "crypto/tls"
5     "fmt"
6     "net"
7     "net/http"
8     "os"
9     "strconv"
10    "time"
11 )
12
13 func getHealthCheckTransport() func() *http.Transport {
14     // Keep TLS config.
15     tlsConfig := &tls.Config{
16         // Can't use SSLv3 because of POODLE and BEAST
17         // Can't use TLSv1.0 because of POODLE and BEAST using CBC cipher
18         // Can't use TLSv1.1 because of RC4 cipher usage
19         MinVersion:      tls.VersionTLS12,
20         InsecureSkipVerify: true, // FIXME: use trusted CA
21     }
22     tr := &http.Transport{
23         Proxy: http.ProxyFromEnvironment,
24         DialContext: (&net.Dialer{
25             Timeout:  2 * time.Second,
26             KeepAlive: 10 * time.Second,
27         }).DialContext,
28         ResponseHeaderTimeout: 5 * time.Second,
29         TLSHandshakeTimeout:   5 * time.Second,
30         ExpectContinueTimeout: 5 * time.Second,
31         TLSClientConfig:       tlsConfig,
32         // Go net/http automatically unzip if content-type is
33         // gzip disable this feature, as we are always interested
```

```

34         // in raw stream.
35         DisableCompression: true,
36     }
37     return func() *http.Transport {
38         return tr
39     }
40 }
41
42 func doHealth(svcURL string) error {
43     endpoint := fmt.Sprintf("%s%s", svcURL, "/minio/health/cluster")
44     req, err := http.NewRequest(http.MethodGet, endpoint, nil)
45     if err != nil {
46         return err
47     }
48
49     httpClient := &http.Client{
50         Transport: getHealthCheckTransport(),
51     }
52
53     resp, err := httpClient.Do(req)
54     if err != nil {
55         return err
56     }
57
58     drivesHealing := 0
59     if v := resp.Header.Get("X-Minio-Healing-Drives"); v != "" {
60         val, err := strconv.Atoi(v)
61         if err == nil {
62             drivesHealing = val
63         }
64     }
65     minDriveWrites := 0
66     if v := resp.Header.Get("X-Minio-Write-Quorum"); v != "" {
67         val, err := strconv.Atoi(v)
68         if err == nil {
69             minDriveWrites = val
70         }
71     }
72     fmt.Printf("drivesHealing: %d, writeQuorum: %d\n", drivesHealing, minDriveWrites)
73     return nil
74 }
75
76 func main() {
77     u := os.Args[1]
78     count := 0
79     maxCount := 1000
80     for {
81         doHealth(u)
82         time.Sleep(500*time.Millisecond)

```

```
83         count++
84         if count > maxCount {
85             time.Sleep(10*time.Hour)
86         }
87     }
88 }
```