Final Project Design 1

Tuesday, November 17, 2020 5:48 PM

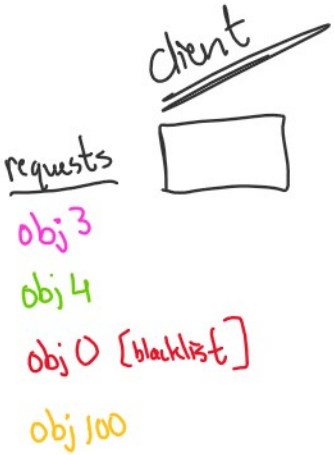
Topics

- rendezvous hashing
- bloom filter
- proxies
- TLS protocol
- parsing

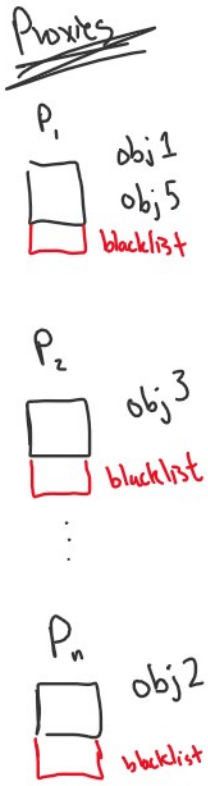
References

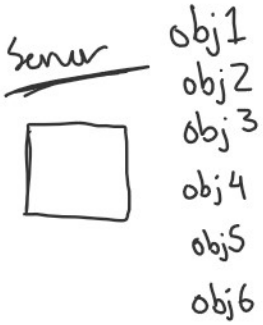
- Bob Beck's libTLS tutorial
- LinuxConf AU 2017 slides
- On Certificate Authorities
- Official libtls documentation

interactions



Proxies





Overview of server/proxy/client system

obj 0 : match on proxy

obj 4: no match on proxy

obj0: match on proxy

1) client sends request to appropriate proxy
2) proxy checks if obj0 on blacklist
3) proxy checks local cache for obj0
4) returns obj0 to client

obj4: no match on proxy

1) client sends request to appropriate proxy
2) proxy checks if obj4 on blacklist
3) proxy checks local cache for obj4
4) proxy sends request to server
5) server searches for obj4
6) server returns obj4 to proxy
7) proxy adds obj4 to cache
8) proxy returns obj4

obj0: item on blacklist

1) client sends request to appropriate proxy
2) proxy checks to see if obj0 on blacklist
3) proxy finds match on blacklist
4) proxy returns empty object to client

obj100: item doesn't exist

1) client sends request to appropriate proxy
2) proxy checks to see if obj100 on blacklist
3) proxy checks local cache for obj100
4) proxy sends request to server
5) server searches for obj100
6) server returns empty object to proxy
7) proxy adds empty object for obj100 to cache
8) proxy returns empty object to client

Hashing

- For a hash function, we will use the **polynomial rolling hash function**

Polynomial Rolling Hash Function

- given a string s of length n, prime p, and large prime m

$$\text{hash}(s) = s[0] + s[1] \cdot p + s[2] \cdot p^2 + \ldots + s[n-1] \cdot p^{n-1} \pmod{n}$$

- For proxy server selection, we use the **Rendezvous Hashing Schema**

Rendezvous Hashing

- we use the highest random weight scheme, where we select a proxy on the largest hash value $h(s)$

We develop a string S by concatenating object name O with proxy name P

$$S = OP \text{, then with } P_1, P_2, \ldots, P_m$$

the proxy used is
the proxy selected from: $\max\left(h(s_1), h(s_2), \ldots h(s_m)\right)$

Bloom Filters

- No more than 6,000 blacklisted objects per proxy
- Must retain less than 1%
- 5 hash functions
- at least 59093 bits (solved below)

Fake Positive Calculation

$$Pr[FP] = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{|R \times 1|}$$

$$\approx \left(1 - e^{-k\frac{n}{m}}\right)^{k}$$

where k hashes, m bits, n elements

→ solving for the minimum number of m-bits is:

$$\text{solve for } m,\ \left(1 - e^{-5 \cdot \frac{6000}{m}}\right)^5 = 0.01 \quad : \quad m = 59092.82192\ldots$$

for the bloom filter, we must pick an array of bits that is at least 59093 bits!

For our example, lets round up to 60,000 bits or 7,500 bytes