A TLS Extension for Service Indication

draft-zhang-tls-service-indication-extension-00

Dacheng Zhang
Dapeng Liu
Alibaba Group
IETF 95
Motivation Scenario (1)

• To attract potential consumers and gain advantages in the market competition, ICPs seek to provide customers with discount for their traffics accessing their services.
• To achieve this, an ICP needs to cooperate with its ISPs and enable the charging gateways of ISPs to distinguish the traffic flows accessing to certain content/services from other traffics.
• In order to achieve this objective, additional Service Indication Information needs to be provided for a charging gateway so that the gateway can find the associated charging policies for the traffic flow.
• Such information should not be provided at the application layer when TLS has been widely used in practice.
Motivation Scenario (2)

- On 11 Nov. 2016, Alibaba attracted over 115 million buyers to its marketplaces and enabled RMB91.2 billion (US$14 billion) in GMV settled through Alipay on Alibaba’s platforms.
- Alibaba’s platform supported 467 million delivery orders during a 24-hour period and enabled about 140,000 peak transactions processed per second.
- So, security is a big concern, but we need a light solution.
Why don’t we use SNI?

- SNI is not use for service indication
- SNI has a length limitation
- No protection is provided for SNI. Moreover, SNI is relatively static, when a SNI of ICP A is know by ICP B, the APP of ICP B can use it to gain benefit
What do we need?

• We need to transfer the Service Indication information in a secure way so that ICP B cannot use the SNI of ICP A without being detected
Our Solution (1)

• We define an extension to carry the SI information and transfer it in the client_hello packet

```c
struct {
    opaque ServiceName;
    uint64 timestamp;
    KeyID key_identifier;
    opaque
        Message_authentication_data;
} ServiceIndicatingInfo;
enum {
    key_id(0)
} KeyID;
```
Our Solution (2)

• We use timestamp and HMAC to guarantee the freshness of the SI information
• In the current solution, the digest only covers the extension, so that an attacker can re-use the token when the timestamp is still valid
• In the future version we consider to have the digest cover the whole packet, which will make the attacks more difficult.
Why don’t we do this work at the TCP or IP layer

- No space for IPv4 header
- There are limits on the lengths TCP options
- TCP and IP are implemented in kernel mode, which makes the deployment of such change more difficult
- ...
Comments?