LURK TLS/DTLS Use Cases

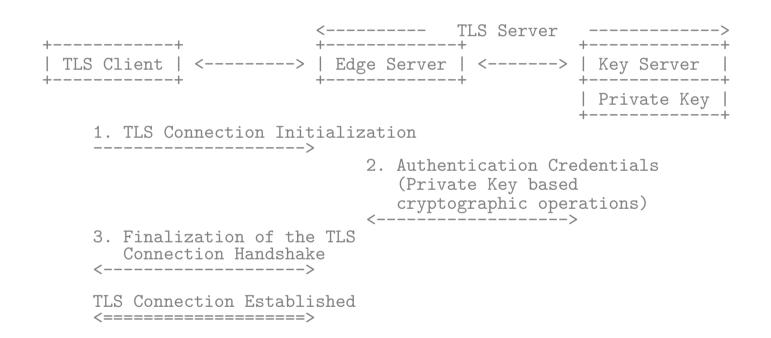
draft-mglt-lurk-use-cases-02

D. Migault, K. Ma, R. Salz, S. Mishra, O. Gonzales de Dios

16/07/2016- IETF96- Berlin

Arch Use Cases Requirements

LURK Architecture



Containers and Virtual Machines Use Case

Problem: No control of the Private Key

- Private Keys are spread all over the data center through:
 - Running instances of VMs / containers
 - Persistent images of VMs / containers
- Isolation may not sufficiently prevent access to the private keys

- Protects the Private Key against leakage by:
 - Outsourcing the Private Key from VMs / containers to the Key Server
 - Preventing direct access to the Private Key by the VMs / containers
 - Restricting Private Keys operations to those authorized by the Key Server
 - In case of priviledge escalation, virtualization isolation breach, etc...
- Provides inter-operability for different OS / applications

Content Provider Use Case

Problem: Edge Servers are exposed to OS-to-applications vulnerabilities

- Risk exposure increases with implementation diversity
- One leakage affects the whole service

- Protects the Private Key against leakage by:
 - Outsourcing the Private Key from Edge Servers to the Key Server
 - Preventing direct access to the Private Key by the Edge Servers
 - Restricting Private Keys operations to those authorized by the Key Server
 - In case a Edge Server become corrupted, etc...

Content Owner / Content Provider Use Case

Problem: Content Owner (URL) wants a CDN to operate without providing the Private Key

- Private key may present more value than the content itself:
 - Content accessed by devices configured with the public credentials need to be replaced/reconfigured in case of private key leakage
 - Content with ephemeral value presents acceptable content leakage risks
 - Content may be encrypted with DRM

- Enables the Private Key to operate without being provided.
- Enables interoperability between independent administrative domains Content Owner, Content Provider

CDN Interconnection Use Case

Problem: Providing the Private Key prevents CDNs to collaborate

- The company with which the Content Owner has contracted may further delegate delivery to another CDN with which the Content Owner has no official business relationship
- The delegating CDN may not even host the Key Server, in which case, it may proxy the communications to the upstream CDN or the Content Owner.

- Enables the Private Key to operate without being provided.
- Enables interoperability between independent administrative domains Content Owner, Content Provider

LURK Requirements

LURK Requirements

- R1: LURK MUST be standardized at the IETF.
- R2: LURK MUST NOT impact the TLS Client.

Key Server Requirements

- R3: The Key Server MUST be able to provide the necessary authentication credentials so the TLS Client and the Edge Server can set an authenticate TLS Connection with the Private Key.
- R4: The Key Server MUST NOT leak any information associated to the Private Key. In particular the Key Server MUST NOT provide a generic signing/encryption oracle.
- R5: The Key Server SHOULD NOT perform any operation outside the authentication of a TLS Connection.
- R6: The Key Server MUST provide confidential information to the Edge Sever only over an authenticated and encrypted channel.

LURK Requirements

Edge Server

The Edge Server SHOULD be provisioned with the public authentication credentials. Note: Public certificate provisioning is outside of LURK. Thank you for your attention