Accessibility

The SPIFFE Workload Endpoint often serves as the mechanism for initial identity bootstrapping, including the delivery and management of roots of trust. Since a workload in its early stages may have no prior knowledge of its identity or whom it should trust, it is very difficult to secure access to the endpoint. As a result, the SPIFFE Workload Endpoint SHOULD be exposed through a local endpoint, and implementers SHOULD NOT expose the same endpoint instance to more than one host. Keeping the endpoint and related traffic confined to a single host mitigates bootstrap problems as they relate to initial authentication and issuance security. Please see the Transport and Authentication sections for more detail.

- **Workload** 跟 **Agent** 在同一裝置**(**本機**)**上**,** 所以可以用像 **putty** 那樣直接找不同進程、**PID**辨識

Transport

The SPIFFE Workload Endpoint MUST be served over gRPC, and compliant clients MUST support gRPC. It may be exposed as either a Unix Domain Socket (UDS) or a TCP listen socket. Implementations SHOULD prefer Unix Domain Socket transport, however TCP is supported for implementations in which Unix Domain Sockets are impractical or impossible. TCP transport MUST NOT be used unless the underlying network allows the Workload Endpoint server to strongly authenticate the workload based on source IP address (e.g., over a localhost or link-local network), or other strong network-level assertions (e.g., via an SDN policy).

As a hardening measure against Server Side Request Forgery (SSRF) attacks, every client request to the SPIFFE Workload Endpoint MUST include the static gRPC metadata key workload.spiffe.io with a value of true (case sensitive). Requests not including this metadata key/value MUST be rejected by the SPIFFE Workload Endpoint (see the Error Codes section for more information). This prevents an attacker from exploiting an SSRF vulnerability to access the SPIFFE Workload Endpoint unless the vulnerability also gives the attacker control over outgoing gRPC metadata.

- **gRPC metadata key :** 在以內網通訊的情況下額外加上的驗證用來防**SSRF**

- 強烈建議用**Unix Domain Socket (UDS)**做單一機台傳輸**,** 因安全性考量非必要勿用**TCP**做傳輸**,** 以減少連線的範圍

- **Workload**初次連線不應要求用**TLS,** 因為**Workload**還沒拿到憑證跟憑證鏈

Locating the Endpoint

Clients may be explicitly configured with the socket location, or may utilize the well-known environment variable SPIFFE_ENDPOINT_SOCKET. If not explicitly configured, conforming clients MUST fall back to the environment variable.

The value of the SPIFFE_ENDPOINT_SOCKET environment variable is structured as an RFC 3986 URI. The scheme MUST be set to either unix or tcp, which indicates that the endpoint is served over a Unix Domain Socket or a TCP listen socket, respectively.

If the scheme is set to unix, then the authority component MUST NOT be set, and the path component MUST be set to the absolute path of the SPIFFE Workload Endpoint Unix Domain Socket (e.g. unix:///path/to/endpoint.sock). The scheme and path components are mandatory, and no other component may be set.

If the scheme is set to tcp, then the host component of the authority MUST be set to an IP address, and the port component of the authority MUST be set to the TCP port number of the SPIFFE Workload Endpoint TCP listen socket. The scheme, host, and port components are mandatory, and no other component may be set. As an example, tcp://127.0.0.1:8000 is valid, and tcp://127.0.0.1:8000/foo is not.

- **在Client 寫死設置 Workload Endpoint 的 socket 位置不然就要在 SPIFFE_ENDPOINT_SOCKET 這項環境變數中設置 socket 位置**

- **URI在 RFC 3986標準下的格式**
  -> scheme : [// **authority**] **path** [? **query**] [# **fragment**]

  scheme : 使用的協議

  authority : 資料來源/供給者

  path : 資源路徑

  query : 網頁傳遞間紀錄的參數

  fragment : 紀錄跳轉到某個頁數或某個部份

- **Unix Domain Socket (UDS)模式下除scheme跟path必填, 其餘欄位皆不可填, 且 path要是絕對路徑 (範例中**unix:///path/to/endpoint.sock **紅色這槓是指絕對路徑)**