Sliver es un framework adversarial C2 de código abierto desarrollado por **Bishop Fox**. Está escrito en Go, soporta multiplataforma (Windows, Linux y macOS) y permite establecer canales de comunicación seguros entre implantes y el servidor central usando protocolos como mTLS, HTTP(S) y DNS. Ofrece un conjunto moderno de capacidades para gestionar comunicaciones con máquinas comprometidas, generar implantes únicos y operar de forma colaborativa durante campañas de evaluación de seguridad.

A diferencia de otras soluciones comerciales, Sliver ofrece:

- Generación dinámica de implantes, cada uno con claves únicas que dificultan la detección por firmas.
- Comunicación cifrada por defecto.
- Soporte para múltiples transports y escenarios de red.
- Modo colaborativo (“multiplayer”) para compartir sesiones y agentes entre operadores.

## ARQUITECTURA BÁSICA

Sliver se compone de dos componentes principales:

- **Servidor C2**: controla y coordina las conexiones entrantes de los agentes.
- **Implantes (“slivers”)**: ejecutables generados por Sliver que se entregan a los objetivos y que se conectan de vuelta al servidor.

Los implantes se generan con el comando **generate**, y el proceso de compilación con ofuscación integrada hace que cada binario sea único, lo que ayuda a evadir mecanismos de seguridad basados en firmas.

## INSTALACIÓN Y PRIMER USO

El primer paso es descargar Sliver desde su repositorio oficial o compilarlo desde código fuente. Se puede instalar mediante una única línea de terminal:

```
curl -sL https://sliver.sh/install | sudo bash
```

Una vez instalado, lo lanzamos rápidamente desde la línea de comandos:

```
sliver
```

Después de iniciar el servidor, desde la interfaz de Sliver podemos generar implantes y listeners. Por ejemplo:

```
 generate --mtls mtls sessions interact
```

Con esto levantamos un listener mTLS, generamos el implante y nos preparamos para interactuar con sesiones desde los agentes cuando se conecten.

## OPERATIVA HABITUAL

Una vez que un implante se conecta, Sliver permite tareas típicas de post-explotación como:

- Enumeración del sistema.
- Subir/descargar archivos.
- Pivotar y redirigir tráfico.

- Ejecutar shellcodes y MSI/reflective DLLs.

## MODO «MULTIPLAYER»

Una de las ventajas diferenciales de Sliver es su modo colaborativo o “multiplayer”. Esto permite que varios operadores trabajen con la misma infraestructura compartida:

- Generar configuración para nuevos operadores con **new-player**.
- Levantar Sliver en modo “multiplayer”.
- Importar configuraciones en otros clientes para acceder a las mismas sesiones.

Con este modo podemos **compartir agentes y sesiones** entre diferentes participantes de una campaña de Red Team, lo cual mejora la coordinación y eficiencia del equipo.

## TÉCNICAS AVANZADAS

Además del modo colaborativo, Sliver permite:

- Pivotes TCP para encadenar conexiones internas.
- Generación y gestión de stagers en diferentes formatos.
- Integración de herramientas externas como módulos o extensiones para enriquecer la funcionalidad de los agentes.

## COMPARATIVA CON OTRAS SOLUCIONES

Sliver compite con frameworks como Cobalt Strike, Covenant o Havoc. Su principal ventaja respecto a algunas alternativas comerciales es que es **open source** y no depende de licencias costosas, además de ofrecer dinamismo en la generación de implantes y comunicaciones cifradas por defecto.

Aunque su interfaz es más básica porque es basada en texto, su flexibilidad y arquitectura modular lo convierten en una herramienta muy útil en campañas de evaluación de seguridad y entornos de laboratorio.

Si queremos profundizar más es sus capacidades podemos revisar exhaustivamente su correspondiente web de documentación aquí.