1. Sistema de Escaneo de Puertos

Versión Básica (Inicio):

- Escribe un programa en Python que escanee los puertos abiertos en una dirección IP utilizando la biblioteca socket.
- Muestra los resultados en consola.

Escalabilidad (Avanzando):

- Añade opciones para escaneo rápido, detallado o personalizado.
- Implementa un reporte en formato HTML o CSV.
- Usa librerías como **Scapy** para escaneos más avanzados (detección de servicios).
- Crea una interfaz gráfica simple para que el usuario pueda interactuar con el programa.

---

2. Honeypot Básico

Versión Básica (Inicio):

- Crea un servidor en Python que simule un servicio (por ejemplo, un servidor web).
- Loguea todas las conexiones entrantes (IP, puerto, hora).

Escalabilidad (Avanzando):

- Añade más servicios simulados (FTP, SSH).
- Implementa detección de patrones sospechosos y genera alertas.
- Usa Docker para desplegar varios honeypots en diferentes puertos.

---

3. Sistema de Análisis de Tráfico de Red

Versión Básica (Inicio):

- Captura tráfico de red local utilizando Wireshark o pyshark.
- Filtra paquetes básicos (HTTP, DNS, etc.) y muestra estadísticas simples (número de paquetes por protocolo).

Escalabilidad (Avanzando):

- Implementa detección de paquetes sospechosos (como intentos de escaneo o DoS).
- Añade visualización gráfica de datos (con Matplotlib o Dash).
- Introduce alertas automáticas basadas en patrones anómalos.

---

4. Sistema de Gestión de Contraseñas

Versión Básica (Inicio):

- Crea un gestor de contraseñas en Python que almacene contraseñas en un archivo cifrado (usa cryptography o bcrypt).
- Añade una funcionalidad para generar contraseñas aleatorias.

Escalabilidad (Avanzando):

- Almacena las contraseñas en una base de datos segura (como SQLite).
- Añade autenticación de dos factores (2FA).
- Crea una interfaz gráfica o una API para interactuar con el gestor.

---

5. Auditoría Básica de Seguridad en un Sistema Local

Versión Básica (Inicio):

- Escribe un script en Python o Bash que:
    - Detecte archivos sin permisos seguros (como permisos 777 en Linux).
    - Liste los servicios activos en el sistema.

Escalabilidad (Avanzando):

- Implementa verificaciones de vulnerabilidades comunes (por ejemplo, servicios expuestos).
- Genera un informe detallado en PDF o HTML.
- Amplía el script para incluir análisis de logs del sistema.

---

6. Simulador de Ataques de Fuerza Bruta

Versión Básica (Inicio):

- Escribe un programa en Python que pruebe combinaciones de contraseñas contra un servidor SSH simulado o un archivo protegido.

Escalabilidad (Avanzando):

- Añade soporte para diferentes protocolos (FTP, HTTP Basic Auth).
- Implementa contramedidas para simular sistemas que bloqueen intentos repetidos.
- Crea un sistema para comparar la eficiencia de métodos de fuerza bruta versus diccionario.

---

7. Dashboard de Seguridad Personal

Versión Básica (Inicio):

- Crea un script que recopile datos de seguridad de tu sistema, como:
    - Estado del firewall.
    - Últimos inicios de sesión.
    - Espacio en disco.

Escalabilidad (Avanzando):

- Convierte el script en una webapp sencilla usando Flask o Django.
- Añade integración con APIs externas (por ejemplo, VirusTotal para análisis de archivos).
- Implementa notificaciones automáticas vía email o Telegram.

---

8. Simulador de Phishing

Versión Básica (Inicio):

- Diseña una página web básica que simule un ataque de phishing (solo para prácticas locales, no reales).
- Loguea cualquier intento de envío de credenciales.

Escalabilidad (Avanzando):

- Introduce simulaciones más avanzadas, como correos electrónicos falsos.
- Implementa una herramienta que evalúe qué tan vulnerable es un usuario a caer en phishing (usando métricas).