OpenLDAP Baseline Security Analyzer

OpenLDAP Baseline Security Analyzer

Inmaculada Bravo García, Reyes Hernández Rodríguez, Mª Teresa Calvo Moya

Resumen

OpenLDAP-BSA proporciona un listado de criterios que guiarán el plan inicial de securización del directorio LDAP. A partir del análisis detallado de las amenazas de seguridad que pueden afectar a este servicio y de las contramedidas correspondientes, se ha implementado además una herramienta funcional que puede ayudar a la evaluación del cumplimiento de esos criterios.

En el proyecto se utiliza la herramienta gráfica OCIL Interpeter (software libre escrito en Java) para interrogar al administrador sobre el cumplimiento de cada uno de los criterios de seguridad propuestos.

El proyecto completo está accesible en la forja de RedIRIS. *(http://openldap-bsa.forja.rediris.es)*[1].

Palabras clave: Seguridad, LDAP, directorio, OCILInterpeter.

Summary

OpenLDAP-BSA provides a list of criteria that will serve as guidance for the initial plan for securing the LDAP directory. Based on the detailed analysis of security threats that may affect this service and the corresponding countermeasures, a functional tool has also been implemented that could help evaluate the extent to which these criteria are fulfilled.

In the project, the graphic tool OCIL Interpreter (javascript freeware) is used to question the administrator on the fulfilment of each of the proposed security criteria.

The full project can be accessed through the RedIRIS Forge. *(http://openldap-bsa.forja.rediris.es)*[1].

Keywords: Security, LDAP, address directory, OCIL Interpreter.

OpenLDAP-BSA utiliza la herramienta gráfica OCIL Interpeter

1. Introducción

El ciclo de vida del servicio de directorio LDAP en nuestras instituciones comienza típicamente como un servicio auxiliar asociado únicamente a la autenticación de usuarios de correo electrónico. En este punto, satisface una demanda interna muy particular y controlada, por lo que su securización no plantea problemas relevantes.

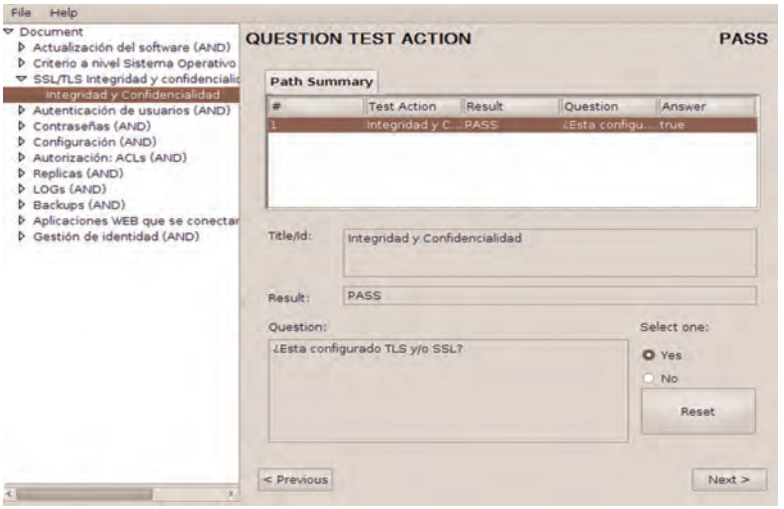
Una de las fases es analizar las amenazas de seguridad que puedan afectar al LDAP

Sin embargo, a medida que el servicio de directorio es usado como referencia para la autenticación requerida por nuevas aplicaciones, los accesos a este servicio comienzan a hacerse más numerosos, variados y complejos, con lo que aparecen un número significativo de vulnerabilidades que se deben abordar en el marco de una política de seguridad integrada.

Hasta ahora no se disponía de una herramienta específica que ayudara a seguir unas pautas normalizadas que guíen el proceso de securización de este servicio. El proyecto OpenLDAP-BSA aborda este problema en las siguientes fases:

1. Un análisis de las amenazas de seguridad que pueden afectar al directorio LDAP
2. Listado de recomendaciones, dirigidas al diseño, configuración y gestión del servicio, con la finalidad de minimizar el impacto de esas amenazas.
3. Implementación de una herramienta funcional para evaluar el cumplimiento de los criterios de seguridad.

IMAGEN 1. IMAGEN DE LA IMPLEMENTACIÓN OPENLDAP-BSA. USA COMO INTERFAZ GRÁFICA LIBRE OCIL INTERPRETER A LA QUE SE LE PROPORCIONA UNA CODIFICACIÓN XML DE LOS CRITERIOS DE SEGURIDAD PROPUESTOS, SIGUIENDO EL SCHEMA OCIL[2]



La comunicación entre el cliente y el servidor puede ser interceptada por un intruso

2. Amenazas y ataques a un servidor LDAP

En esta sección se enumeran brevemente algunas de las amenazas que debe anticipar un administrador de LDAP, independientemente de la tecnología de directorio que se use.

Defectos de programación (bugs). Como en todo programa suficientemente complejo, pueden aparecer vulnerabilidades causadas por un análisis insuficiente de los escenarios de uso o por una evolución no prevista de ese contexto.

Accesos indebidos al sistema de archivos, provocados por una deficiente política de privilegios del sistema.

Cuando no se validan bien los parámetros introducidos, pueden usarse varias técnicas de ataques que permiten manipular las consultas y obtener accesos indebidos

Interceptación de la comunicación (man-in-the-middle). La comunicación entre el cliente y el servidor puede ser interceptada por un intruso de modo transparente mediante diversas técnicas de "envenenamiento de ARP". Como consecuencia, las credenciales pueden ser usadas por terceros indebidamente o usurpar la identidad[3].

Acceso a los datos transmitidos. Si los datos no están cifrados, pueden ser capturados durante la transmisión mediante un rastreador (sniffer).

Consecución de credenciales por "fuerza bruta". Cualquier servicio que requiere autentificación de usuarios puede verse sometido a intentos de validación automatizada mediante combinación de caracteres o palabras de listados predefinidos ("diccionarios").

Inyecciones de código en aplicaciones web. Cuando el servicio de autenticación de LDAP es requerido por aplicaciones web que no validan adecuadamente los parámetros introducidos, pueden usarse varias técnicas de ataque que permiten manipular las consultas y obtener accesos indebidos. (LDAPInjection, Blind LDAP Injection)[4].

Modificación de datos. Excesivos privilegios en algunos usuarios (ya sean legítimos o usurpados) pueden permitir la modificación de datos clave.

Denegación de Servicio. Como consecuencia de una incapacidad para atender un elevado número de peticiones, ya sean producidas por un ataque intencionado o por aplicaciones internas mal configuradas, puede producirse una caída del sistema.

Google y ficheros olvidados en un servidor web. Una deficiente organización del sistema de almacenado puede dar acceso público a archivos que comprometen la seguridad.

3. Criterios de seguridad

A partir de las amenazas descritas anteriormente, de otras consideraciones basadas en la experiencia en nuestras instituciones y de diversas fuentes bibliográficas (se referencia tres de ellas[5][6][7]), se ha elaborado la siguiente lista de recomendaciones para llevar a cabo una política de seguridad adecuada en relación al servicio de directorio.

Uno de los criterios a tener en cuenta es que para mantener la seguridad en el directorio hay que actualizar el software

(Nota: Puede encontrarse una explicación más detallada acerca del impacto, valoración e implementación de cada criterio en la página web del proyecto: http://openldap-bsa.forja.rediris.es. Además está disponible para su descarga una herramienta de auditoría basada en estas recomendaciones que advierte de los errores más frecuentes).

1. **Software actualizado.** Comprobar que el software está actualizado a su última versión estable evitará fallos ya corregidos.

2. **Consideraciones a nivel del Sistema Operativo.** En este apartado se analizan reglas básicas a nivel de sistema operativo, tales como la ejecución del demonio slapd, permisos sobre ficheros y directorios y la conveniencia de cifrar la base de datos. Estas reglas son:

   2.1. **Reglas básicas:** Mantener actualizado el sistema operativo; utilizar un sistema dedicado y redundante; instalar un firewall interno para filtrar las conexiones desde determinadas IPs y hacia determinados puertos.

   2.2. **Ejecución del demonio slapd.**

   - Crear un usuario y grupo específico no privilegiado con el que se arrancará el demonio.
   - Controlar en qué interfaces y puertos se ejecuta.
   - Se puede Ejecutar slapd en un entorno enjaulado CHROOT.
   - Evitar exponer directamente el servidor LDAP a internet.

   2.3. **Permisos sobre los directorios y archivos del OpenLDAP.** Los directorios y archivos deben tener los mínimos privilegios para evitar ser accedidos por usuarios no autorizados, con atención particular a ficheros de configuración, de schemas, de bases de datos, de bitácora (logs), de volcados (ldif).

   2.4. **Cifrar la base de datos bdb.** En entornos con servidores no dedicados donde el acceso al sistema de archivos no se puede garantizar, una opción para securizar los datos del directorio es cifrar la base de datos, teniendo en cuenta la penalización en la velocidad de respuesta.

Una opción para securizar los datos del directorio es cifrar la base de datos

3. **SSL/TLS Integridad y confidencialidad.** SSL/TLS provee dos elementos importantes de seguridad: por un lado garantiza al cliente la autenticidad del servidor, y por otro, permite la confidencialidad en la comunicación al cifrar la transmisión de datos. Se pueden configurar ambas opciones simultáneamente. SSL requiere un puerto diferente para el tráfico cifrado que suele ser el 636; TLS es un refinamiento de SSL más flexible y permite que los clientes que se conecten al puerto estándar 389 de LDAP puedan escoger entre transmisiones en claro o cifradas. Se negociará StartTLS al inicio de la comunicación entre el servidor y el cliente. Para obligar que ciertas operaciones o conexiones se realicen con TLS, se utiliza SSF.

El SSL/TLS garantiza la autenticidad del servidor y la confidencialidad en la comunicación

4. **Autenticación de usuarios.** Uno de los principales usos de un directorio es la autenticación de usuarios desde diferentes aplicativos.

   4.1. **Modo correcto de realizar la autenticación de usuarios.** El objetivo es evitar malos diseños de aplicaciones, que se conviertan en aplicaciones incompatibles, que no se pueda controlar los login fallidos, que se generen problemas de seguridad, etc. Estos 6 pasos estándar se podrán utilizar contra cualquier LDAP, con cualquier estructura, cualquier tecnología de cifrado o cualquier otro criterio:

   - Solicitar alias y contraseña del usuario.
   - Enlazar con el LDAP (hacer un bind) con la cuenta de usuario privilegiado creada para la aplicación. También se podría utilizar el usuario anónimo pero es mejor deshabilitar los bind anónimos.
   - Buscar en el directorio el DN asociado al alias del usuario.
   - Si devuelve una entrada y solo una, éste es el DN del usuario buscado, si hay cero o más de una entrada se devolverá usuario no encontrado.
   - Rebind al LDAP con el DN devuelto en el paso anterior y la contraseña del usuario obtenida en el primer paso.
   - Si LDAP permite el BIND el login ha tenido éxito, si no devuelve "contraseña no válida".

   4.2. **Crear cuentas administrativas y grupos.** Crear cuentas administrativas para cada aplicación o para delegar funciones, controlando tanto a los que tengan el mínimo acceso posible, como a las partes del directorio como desde qué IPs o dominios se permite su conexión. Crear grupos, para definir distintos roles y simplificar la lista de control de accesos definiendo permisos por grupos.

Ppolicy ofrece la posibilidad de realizar bloqueos de cuentas de manera temporal o permanente

5. **Contraseñas.** Se analiza la importancia del esquema de almacenamiento de las contraseñas y cómo implementar y mantener una política de contraseñas, exigiendo complejidad y cambios periódicos, bloqueando temporalmente cuentas tras varias conexiones fallidas.

   5.1. **Esquemas de almacenamiento de las contraseñas.** El atributo que se suele utilizar es userPassword, es multivaluado y cada valor puede estar almacenado con un formato diferente (SSHA, SMD5, MD5, SHA, CRYPT). Se recomienda SSHA.

   5.2. **Política de contraseñas: overlay PPOLICY.**
   - 5.2.1. **Complejidad de las contraseñas.** Controlar la fortaleza de las contraseñas
   - 5.2.2. Obligar a los usuarios a realizar **cambios periódicos.**
   - 5.2.3. **Bloqueos de Cuentas.** Ppolicy ofrece la posibilidad de realizar bloqueos después de un número de intentos de bind fallidos dentro de un intervalo de tiempo dado y que este bloqueo sea temporal o permanente.
   - 5.2.4. **Comprobar qué cuentas se bloquean:** desde qué IPs se realizan los binds fallidos para crear un listado de IPs peligrosas y bloquearlas en el firewall.

6. **Configuración sladp.conf.** Existen dos bloques de directivas: globales (al principio del documento) y de backend o bases de datos. Las directivas especificadas en los backend sobreescribirán las globales.

    6.1. **Evitar ldapv2.** Por defecto ldapv2 no está habilitado. Evitarlo siempre que sea posible ya que perdemos las capacidades y mejoras que aporta ldapv3.

    6.2. **Evitar Accesos Anónimos.** Evitar bind anónimos y sesiones no autenticadas.

    6.3. **Límites.** Especificar determinados rangos en la configuración evitará determinados problemas.

    - sizelimit: número de entradas que será devuelto al realizar una consulta. Por defecto son 500. Limitarlo, p.e. a 50 incrementa el trabajo para los atacantes con ldap injection y se evita cargar al servidor.
    - idletimeout: tiempo de espera para forzar la desconexión de sesiones desocupadas.
    - timelimit: tiempo de ejecución de una consulta, si no se limita al hacer búsquedas por atributos no indexados se hace un recorrido secuencial por todo el directorio, lo cual sucede por un error en la aplicación o por un intento de ataque con ldap injection.

    6.4. **SSF (Security Strength Factor, fortaleza de la securización).** Con esta directiva se pueden controlar los criterios de integridad y confidencialidad que se requerirán para realizar ciertas tareas.

    6.5. **Password del rootdn.** La directiva rootdnpw sirve para especificar la password del rootdn (cuenta de administración del directorio). En lugar de especificarla en el archivo de configuración se deberá crear una entrada en el directorio para este usuario.

7. **Autorización: ACLs.** En OpenLDAP el mecanismo para autorizar o denegar accesos a ciertas partes del directorio son las ACLs (Listas de Control de Accesos).

    7.1. **ACLs globales.** Las ACLS en el bloque de las directivas globales afectarán a todos los backend. Proteger en este bloque sólo las partes del directorio: rootDSE y cn=Subschema.

    7.2. **ACLs en los Backend.** Sobreescribirán a las globales.
        7.2.1. **Origen de la conexión:** en primer lugar se debe limitar desde qué IPs o dominios se pueden conectar ciertas cuentas administrativas o grupos.
        7.2.2. **Réplicas:** el usuario utilizado para mantener las réplicas, debe tener acceso a todas las ramas que se van a replicar, o a todo el directorio si la réplica es completa. Además no debe ser afectado por los límites explicados en el punto 6.3.
        7.2.3. **Attributo userPassword:** restringirlo lo más posible, de hecho ningún usuario debería poder leerlo. Con el tipo de acceso =xw sólo se permite autenticar contra el atributo y actualizarlo, pero no leerlo.
        7.2.4. **objectClass posixAccount:** Clase que permite usar LDAP con NIS. No se debe permitir a los propios usuarios modificar la mayoría de sus atributos.
        7.2.5. **Accesos a grupos:** se pueden dar permisos a todos los usuarios que pertenezcan a un grupo.

Evitar ldapv2 porque perdemos capacidades y mejoras aportadas por ldapv3

Restringir lo más posible el atributo user Password

7.2.6. **Expresiones Regulares:** se pueden especificar ACLs con expresiones regulares. Además se pueden agrupar dichas expresiones entre paréntesis para poder ser utilizadas como variables en las frases "by".

7.2.7. **Set:** se pueden diseñar ACLs complejas utilizando "set", que permite operadores booleanos y acceso a valores de atributos.

7.2.8. **Comando slapacl:** se usa para comprobar si las ACLs escritas tienen el comportamiento deseado.

8. **Réplicas.** Las ACLs no viajan con los datos, por lo que es importante mantener las mismas políticas de seguridad en todas las réplicas del directorio.

9. **Bitácoras (LOGs).** Se ofrecen diferentes niveles de log que se puede usar solos o en combinación con otros.

La mayor parte de los ataques a un directorio se producen a través de las aplicaciones web que lo utilizan

10. **Copias de respaldo (Backups).** Con el comando "slapcat" se puede realizar un volcado completo de la base de datos a un archivo "ldif". Es conveniente que las copias se guarden cifradas.

11. **Aplicaciones web que se conectan con el directorio.** La mayor parte de los ataques a un directorio se producen a través de las aplicaciones web que lo utilizan. Recomendaciones:

    - Validar siempre los datos recibidos del lado del cliente para asegurar que corresponden al tipo esperado.
    - Limitar los datos que se envían al cliente.
    - Utilizar el modo correcto de autenticación de usuarios.
    - HTTPs: los datos transmitidos entre el cliente y el servidor web deben estar cifrados.
    - Configurar el módulo del servidor Apache "modSecurity".
    - Ofrecer a los aplicativos una pasarela controlada para conectarse al LDAP. Así se evitarán las posibles inyecciones de código, búsquedas por atributos no indexados, etc.
    - Instalar un sistema de autenticación centralizada de usuarios para ofrecer un punto de validación de usuarios a cualquier aplicativo que lo solicite de un modo completamente controlado.
    - Configurar ACLs sobre los atributos y clases, dar siempre los mínimos privilegios posibles, sobre todo a las identidades que ejecutan servicios a través de la web.

Este proyecto ha elaborado una guía de recomendaciones de seguridad

12. **Gestión de IDENTIDAD.** La institución debe tener descrita una política de gestión de identidades que determine el ciclo de vida de cada usuario en el directorio. Es un importante agujero de seguridad mantener usuarios con ciertos privilegios después de que su relación con la misma haya terminado.

5. Conclusión

El directorio es un servicio clave en la institución y es un objetivo potencial de atacantes, ya que existen múltiples amenazas. Aunque cualquier modificación en el directorio afecta a muchas aplicaciones, es necesario reconducir malos hábitos.

Con este proyecto se ha elaborado una guía de recomendaciones de seguridad para desarrollar un plan inicial de securización, y proporciona una herramienta para evaluar la seguridad del directorio.

Referencias

[1] OpenLDAP Baseline Security Analyzer (*http://openldap-bsa.forja.rediris.es/*)

[2] OCIL: The Open Checklist Interactive Language (*http://scap.nist.gov/specifications/ocil/index.html*)

[3] Ataques a LDAP *(http://elladodelmal.blogspot.com/2008/05/ataques-ldap-i-de-iv.html)*

[4] Welcome to IEEE Xplore 2.0: LDAP injection techniques *(http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4737330)*

[5] OpenLDAP, Main Page *(http://www.openldap.org/)*

[6] *Mastering OpenLDAP: Configuring, Securing and Integrating Directory Services* by Matt Butches

[7] *LDAP Presentations* by Adam Tauno Williams

Inmaculada Bravo García
(inma@usal.es)
Reyes Hernández Rodríguez
(reyes@usal.es)
Mª Teresa Calvo Moya
(mcalvo@usal.es)
Universidad de Salamanca