Test de penetración y gestión de vulnerabilidades, estrategia clave para evaluar la seguridad de red

Por los Profesores: Dr. Javier Areitio Bertolín y Dra. Ana Areitio Bertolín

- El Dr. Javier Areitio Bertolín es Catedrático de la Facultad de Ingeniería. ESIDE y Director del Grupo de Investigación Redes y Sistemas de la Universidad de Deusto.
- Dra. Ana Areitio Bertolín. Laboratorio de Informática Aplicada. Universidad del País Vasco (UPV/EHU).

En este artículo se examina el resultado de combinar los test de penetración en conjunción con la gestión de vulnerabilidades dando lugar a una estrategia moderna esencial para que toda organización pueda conocer con rigor y precisión como se encuentra su nivel de seguridad de la información, así mismo permite valorar el grado de seguridad existente, justificar cambios o la inclusión de nuevas medidas adicionales de seguridad y en última instancia mejorar la seguridad desde un punto de vista real, de modo que se puedan cumplir los niveles de riesgo seleccionados por los responsables para dicha organización con el mínimo de recursos utilizados. Una cuestión muy importante actualmente es poder saber la calidad de la seguridad que se incorpora en las empresas. En el presente artículo se describe como determinar-evaluar de forma profesional, a nivel operativo, el nivel de seguridad de una organización y en su caso poner remedios.

Introducción

Un test de penetración (también denominado hacking ético o pentest) permite identificar las vulnerabilidades de un sistema, aplicación, red o proceso de negocio concreto, es decir posibilita medir el riesgo y valorar los mecanismos de seguridad existentes; muestra donde falla la seguridad o es escasa y se puede utilizar para justificar la necesidad de una actualización, un mayor presupuesto para seguridad o para validar la valoración de riesgos. Se basa en explotar las vulnerabilidades para demostrar que los mecanismos de seguridad pueden fallar y de hecho fallarán. Muy importante antes de comenzar los tests de penetración, las personas que lo realizarán deben contar con un documento del responsable del negocio o CIO que les autorice. El documento incluye quienes realizarán el test, cuando se efectuará, por que razones se realiza y que tipos de actividades se llevaran a cabo, incluye sistemas y ubicaciones, clientes contactados para verificación y puede incluir razones para concluir prematuramente el test (por ejemplo impacto negativo en la red o sistemas). La seguridad en general es un deseo de estar "libre" de peligros, el objetivo es que no sucedan "cosas malas". La seguridad de computador se esfuerza en crear una plataforma de computación que sea lo suficientemente segura, diseñada de una forma que usuarios o programas sólo puedan realizar acciones que se les haya permitido. Esto implica especificar e implementar una política de seguridad. En el contexto de los sistemas de computación que comparten información, se trata de reducir sólo a las entidades autorizadas todo acceso a la información apropiada. La provisión o revelación de información es el elemento clave.

La seguridad de la información engloba todos los conceptos, técnicas, medidas tecnológicas y administrativas utilizadas para proteger los activos de información de todo tipo de acciones deliberadas o no intencionadas de adquisición, robo, daño, revelación, manipulación, modificación, pérdida o uso no autorizado. Actualmente la seguridad de la información presenta una nueva perspectiva, ha pasado de ser un problema técnico-tecnológico a ser un problema de negocios-institucional, de ser una cuestión u objetivo propiedad del departamento de TIC (Tecnologías de la Información y la Comunicación) a ser un asunto propiedad de la organización / del negocio, de ser tratado intermitentemente

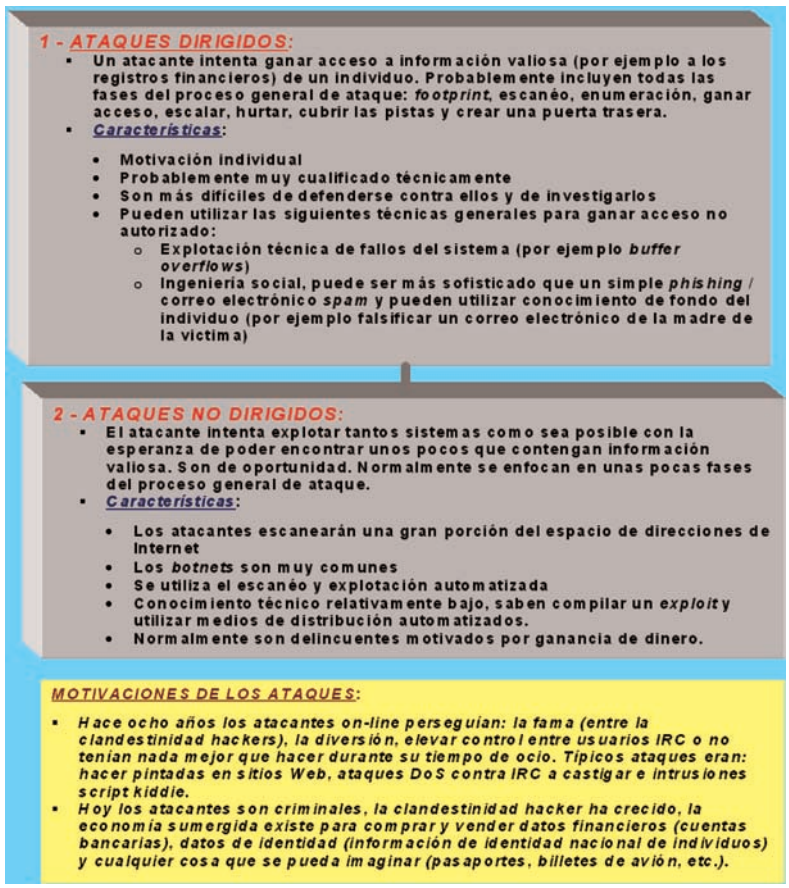


Figura 1. Tipos de ataques de red

a ser tratado de forma integral, de verse como algo que conduce sólo a gastos a verse como una inversión, de verse que se centra en aplicación/plataforma/práctica a verse que se centra en procesos, de verse como un enfoque de seguridad y supervivencia a verse con un enfoque de ventaja competitiva facilitando la continuidad de negocios y la resiliciencia en la empresa.

Hoy en día la seguridad de información soporta los objetivos de negocios clave como: gestión de riesgos, asegurar el gobierno corporativo, reducir costos, mejorar la experiencia del cliente y mejorar los ingresos. La seguridad de la información hoy en día pretende reducir el nivel de riesgos (donde se conjugan amenazas, vulnerabilidades, medidas de seguridad existentes, impactos) a los activos a un valor seleccionado por los responsables del negocio. Es decir se trata de invertir en tecnología y procesos que ayuden a reducir el riesgo más altos no deseados por los responsables de la organización con la mínima cantidad de recursos necesarios.

Según el GTISC (Georgia Tech Information Security Center) en su informe de Octubre del 2008 titulado Emerging Cyber Threats Report for 2009, las cinco principales amenazas a la seguridad que pueden esperarse para el 2009 son: el malware, los botnets, el cyber-warfare, las amenazas a VoIP y dispositivos móviles y la evolución de la economía del ciber-crimen. En Octubre del 2007 en un informe similar las cinco principales amenazas que pronosticó para el 2008 fueron: ataques a Web 2.0 y del lado del cliente, ataques dirigidos a mensajería, botnets, amenazas dirigidas a la convergencia móvil y las amenazas a los sistemas RFID (Radio Frequency IDentification).

BENEFICIOS DE LOS TEST DE PENETRACIÓN	▪ Evalúa la efectividad del programa de seguridad global ▪ Permite claridad de resultados, posibilita demostrar fácilmente a otros los riesgos de seguridad ▪ Se enfoca en el remedio priorizando lo primero que se necesita ▪ Puede validar las acciones de remedio más allá de los *parches*.
Evaluar la efectividad de un sistema de seguridad como un todo es una tarea muy compleja.	

COMPARATIVA ENTRE TEST DE PENETRACIÓN Y ESCANEO DE VULNERABILIDADES	
TEST DE PENETRACIÓN	▪ Utiliza ataques reales para evaluar las defensas y para complementar la gestión de vulnerabilidades ▪ Es muy bueno para priorizar las vulnerabilidades y la única opción para poder evaluar todas las defensas.
ESCANEO DE VULNERABILIDADES	▪ Es muy eficiente, pero no representa realmente las amenazas y sufre de falsos positivos. ▪ Es muy bueno como base para conocer lo que se tiene en la red o sistema

Figura 2. Beneficios de los test de penetración. Comparativa test de penetración contra escaneo de vulnerabilidades.

Razones para los test de penetración.

Algunas razones del por qué de los test de penetración son:
(i) Para medir la seguridad de un sistema, red o proceso de negocios por parte de una tercera parte.
(ii) Para valorar los posibles riesgos.
(iii) Para hacer a la alta dirección "consciente de la seguridad". La seguridad de la información se debe ver actualmente como un requisito no negociable para la organización.

Las instituciones financieras deben proteger sus redes para mantener la seguridad de todo el sistema financiero. Pero sin la capacidad de valorar el riesgo las organizaciones están volando a ciegas. Las valoraciones de seguridad de las tecnologías de la información se realizan actualmente con una mezcla de exploración de vulnerabilidades y test de penetración.

Un test de penetración es un intento de comprometer la seguridad utilizando las mismas técnicas empleadas por un atacante. La pregunta que se formula todo comprobador de penetración es si fuera un atacante hasta donde podría ir en su ataque. Es decir que facilidad existe para comprometer la seguridad de un computador o red o aplicación o sistema. La exploración de vulnerabilidades busca evidencias en cuanto a versiones de software vulnerables, a la presencia de carencia de parches, a la existencia de configuraciones incorrectas, etc. Los atacantes malos no ejecutan herramientas del tipo valoración de vulnerabilidades o Security Scanner como Nessus (http://www.nessus.org/). En cambio los exploradores de vulnerabilidades autorizados cada vez más ejecutan este tipo de herramientas que muestran gráficamente incluso en forma de diagrama de porciones la valoración por ejemplo de los agujeros, los warnings y las notas de seguridad relativas a los computadores explorados.

La exploración de vulnerabilidades por si sola no es suficiente. Ya que no dice lo que un atacante puede hacer a una red actualmente, tampoco identifica las relaciones de confianza peligrosas entre componentes, produce muchos falsos positivos que deben ser manualmente verificados, solo los elementos medibles son identificados como carentes de parches. Las organizaciones deberían aprovecharse de la combinación de la exploración de vulnerabilidades y de los test de penetración. La exploración de vulnerabilidades proporciona una base a partir de la cual empezar a construir un perfil de riesgo. Un test de penetración muestra lo que las vulnerabilidades significan para la organización en la actualidad y pueden ayudar a verificar los esfuerzos de remedios. El sistema financiero no puede permitir a las instituciones no realizar tests periódicos de penetración.

Objetivos de un test de penetración. Identificación de elementos a comprobar en el test.

Posibles objetivos de un test de penetración o hacking ético son:

(i) Conocer cuanta información de nuestra red esta disponible públicamente.

(ii) Conocer si es posible comprometer este o ese sistema.

(iii) Conocer si es posible perturbar un determinado proceso de negocios.

(iv) Conocer el nivel de efectividad de los controles de seguridad que tengamos implantados: firewall, antivirus-spam-filtrado de contenidos URL, sistemas de detección y prevención de intrusiones, sistemas IAM (Identity and Access Management), etc.

(v) Conocer si la política de seguridad de la información aplicada en una organización es correcta.

(vi) Saber si los empleados pueden comprometer la seguridad de sus estaciones de trabajo y PDAs. (vii) Conocer si estamos seguros en una organización.

Algunos de los elementos que pueden ser comprobados son:

(i) Servidores y estaciones de trabajo: servidor Web, servidor de base de datos, servidor DNS, controlador de dominio (PDC, Principal Domain Controler), estaciones de trabajo, PDAs, etc.

(ii) Infraestructura: dispositivos de red (routers, switch, etc.), VPNs, acceso dial-in (RAS), redes inalámbricas (puntos de acceso Wi-Fi, etc). Herramientas de hacking de redes IEEE 802.11a/b/g son NetStumbler (http://www.netstumbler.com), el sniffer-disector de red WiFi Kismet (http://www.kismetwireless.net), AirSnort, WEPCrack, THC-RUT, etc.

(iii) Aplicaciones.

(iv) Empleados utilizando ingeniería social.

Algunas vulnerabilidades del software del lado del cliente son: IE (Internet Explorer), Firefox, Outlook, Thunderbird, MSM Messenger, AOL IM, ICQ, Media Players, procesadores/lectores de imágenes y documentos. Algunos ejemplos: Vulnerabilidad de objeto COM devenum.dll del IE (MS05-038); Vulnerabilidad de procesamiento PNG del MSM Messenger (MS05-009); Vulnerabilidad WMF de Windows (KB912840). Las vulnerabilidades pueden clasificarse en locales o remotas y de severidad alta, media o baja.

Para realizar los test de penetración se debe simular a los atacantes. Los principales modos de test de penetración son:

(i) Ataque desde fuera. Al igual que los posibles adversarios externos: competidores, terroristas, script kiddies, periodistas deshonestos.

(ii) Ataque desde dentro. Al igual que lo harían: empleados deshonestos, empleados descontentos, consultores, contratados.

(iii) Ataque a equipamiento robado. Se trata de sustraer portátiles, servidores, PDAs, teléfonos móviles inteligentes, etc.

(iv) Ataque para valorar la entrada física.

(v) Ataque saltándose la autenticación, por ejemplo un punto de acceso WiFi.

(vi) Ataque al componente humano utilizando ingeniería social. Se trata de engañar y manipular las mentes de las personas de algún modo sutil para obtener cierto tipo de información.

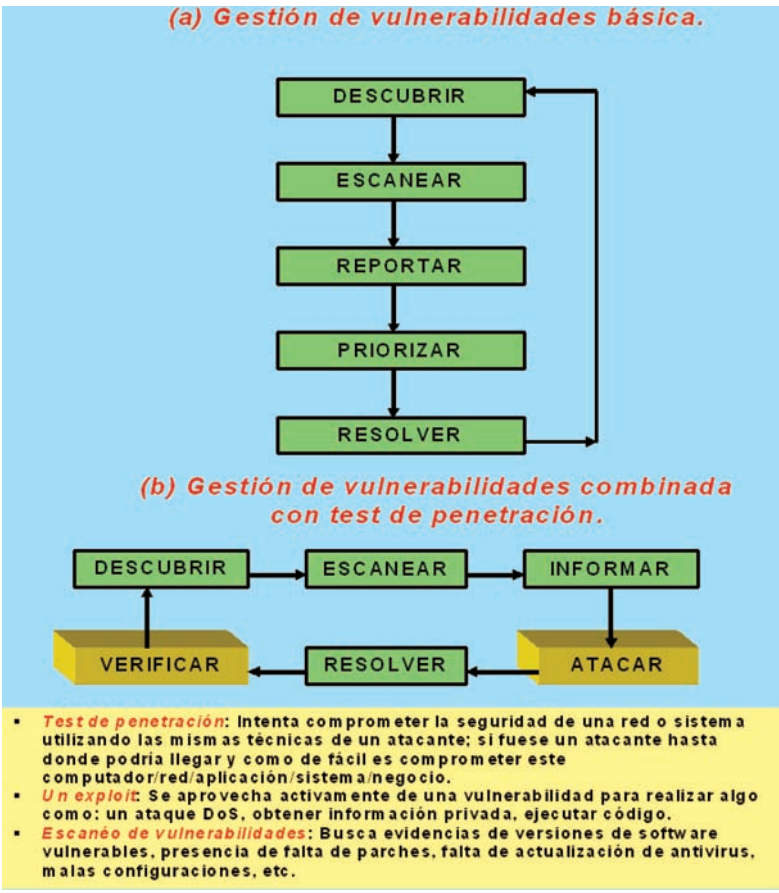


Figura 3. Gestión de vulnerabilidades básica mejorada con test de penetración

Fases de un test de penetración.

Las fases de un test de penetración son:

(1) Recoger información.

En esta fase se intenta recopilar la mayor cantidad posible de información pública. Para ello se utiliza Internic, IANA/RIPE, Whois (herramienta para footprinting y reconocimiento: http://www.allwhois.com/), Google/Usenet, páginas privadas de empleados, direcciones de correo electrónico, números de teléfono. Por ejemplo para conocer información sobre el URL xxx.com se utiliza: http://reports.internic.net/cgi/whois?whois_nic=xxx.com&type=domain. Así mismo para buscar información acerca de la dirección IP 194.34.99.145 se busca utilizando: http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=194.34.99.145.

La sintaxis de búsqueda en Google es:

(i) allintitle:"Index of /etc"
(ii) site:gov site:mil site:ztarget.com
(iii) filetype:doc filetype:pdf filetype:xls
(iv) intitle:, inurl:, allinurl:
(v) allinurl:mssql, allinurl:gw …… (vi) inurl:".aspx?ReturnUrl="
(vii) "+www.ernw.+de"
(viii) related:www.deusto.es
(ix) login site:www.deusto.es
(x) [cached]. Listas de correo /forums /Usenet: Algunos fabricantes soportan preguntas de correo electrónico interno a grupos de news.

Una herramienta valiosa es Google hacking, se trata de utilizar el buscador Google para hacer hacking, es decir para encontrar información acerca del objetivo a atacar-penetrar. Esta información se abre a la Web de forma inadvertida. Algunos ejemplos: ficheros de historia shell (intitle:index.of .bash_history); portales intranet mal configurados ("Bienvenido a la Intranet de la empresa XXX"); cámaras de red panasonic (inurl:"ViewerFrame?Mode="); los resultados de un test de penetración ("la empresa XXX realizó una valoración de vulnerabilidades"); software vulnerable: conocidas vulnerabilidades cross-site scripting ("PHP xxxx:Admin.php"), conocidas vulnerabilidades PHP ("Powerred by: xxxxx Version 1.1.5") permite ejecución de código remoto.

Algunas posibilidades son:
(i) Listar directorios: "intitle:index.of site: <mi-dominio.com>" (Servidor Apache).
(ii) Mensajes de error y warning: "error | warning site: <mi-dominio.com>".
(iii) Recolección de dirección de correo electrónico. ¿De qué forma obtuvieron mi dirección de correo electrónico?: "[a-z]*@[a-z]*mi-dominio.com".
(iv) API Google. Permite realizar fácilmente interrogaciones de forma automatizada. Posibilita encontrar subdominios expuestos. Un atacante puede encontrar elementos críticos de tu red.

Posibles técnicas footprinting para fisgar son:
(i) Encontrar en el sitio Web <mi-compañía.com> comentarios en el código fuente, las direcciones de correo electrónico del desarrollador, los nombres de los administradores y quizás números de teléfono internos. (ii) Utilizar USENET y otros foros Web para preguntar acerca del personal de <mi-compañía.com> en relación al hardware/software

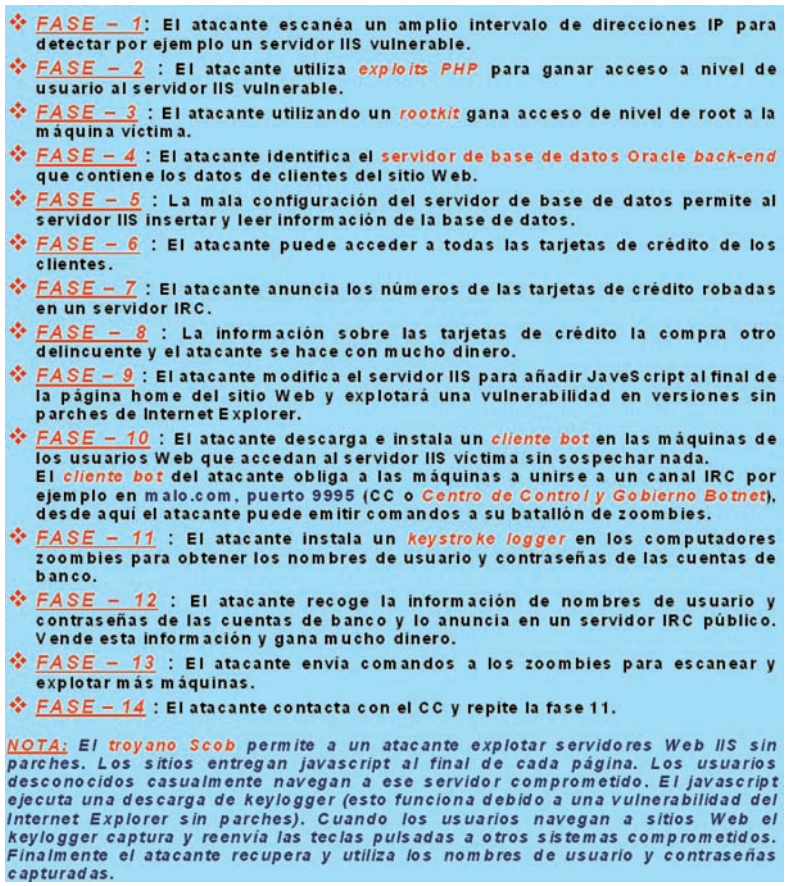


Figura 4. Fases de un ataque

que se utiliza, los nombres de los empleados, sus direcciones de correo electrónico, etc.

(2) Analizar la infraestructura y las máquinas.

Se utilizan técnicas como:

(i) Quering System e información de DNS. Se pueden utilizar:

(a) TraceRoute. Traza la ruta de red que da información acerca del proveedor y del tipo de conexión (simple, redundante o con carga balanceada). Permite saber en que salto o hop el ICMP se bloquea.

(b) Transferencia de Zona DNS. El servidor DNS debería configurarse para no permitir Transferencias de Zona salvo a correspondientes específicos. Las Zonas DNS son una funte de información muy interesante, nos dicen qué máquinas existen en la Zona y se obtiene información acerca de la estructura de la red IP. Se utiliza nslookup.

(ii) Exploración de puertos y Fingerprinting. El escaneo de puertos da información acerca de que puertos escucha una máquina. Cada puerto abierto es potencialmente vulnerable. Escaners más avanzados descubren la clase de software del sistema operativo esta instalada (fabricante y versión). Los scanners más populares son SuperSacan y Nmap. También se puede conectar con NetCat (http://www.atstake.com/research/tools/network_utilities/nc11nt.zip) o Telnet a un servicio para obtener información detallada, esta técnica se denomina "banner grabbing".

(iii) Exploración y explotación de vulnerabilidades. Se pueden utilizar escaners automatizados que comprueban vulnerabilidades conocidas y dan más información para investigar las vulnerabilidades. Existen en Internet bases de datos de vulnerabilidades y exploits:

(a) SecurityFocus (http://www.securityfocus.com/bid/vendor).

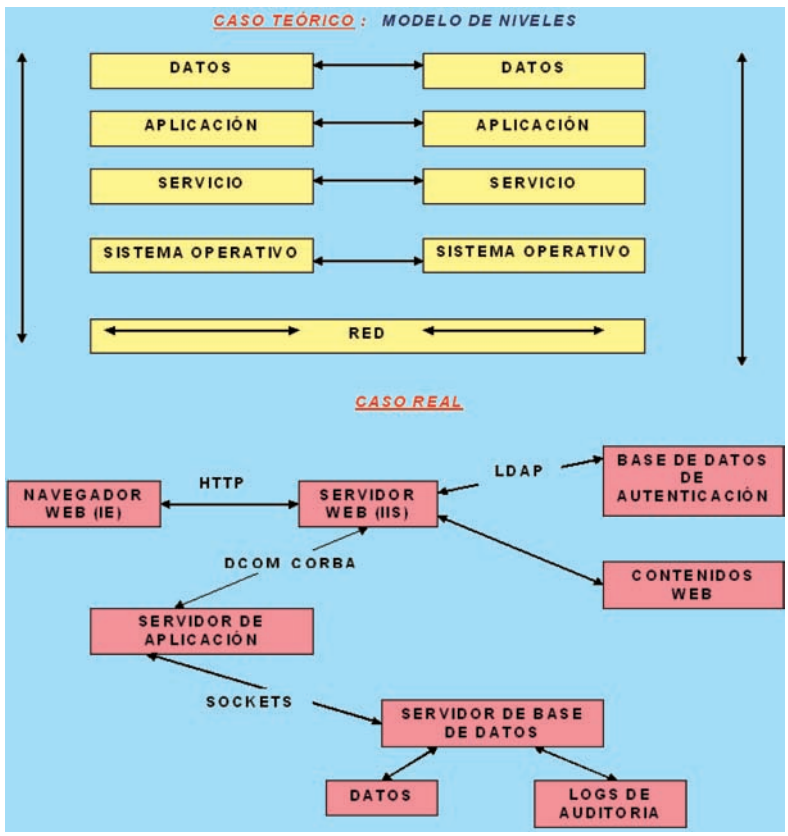
(b) Packet Store (http://packetstormsecurity.org). Algunos escaners de vulnerabilidades de sistemas/host son: Nessus (http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=194.34.99.145.), Retina (http://www.eeye.com), ISS Security Scanner (http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=194.34.99.145.) y MBSA de Microsoft (http://www.microsoft.com). Algunos escaner de base de datos son: MetaCoreTex (http://www.metacoretex.com), AppSecIncAppDetective (http://www.appsecinc.com) e ISSDatabaseScanner (http://www.iss.net). Uno de los escáner para servidores Web es Nikto (http://www.cirt.net). Otros escáner para Web son: WebInspect (http://www.spidynamics.com) y NStealth (http://www.nstalker.com).

(3) Analizar las máquinas.

Utilizando fingerprinting y escaneo de puertos/vulnerabilidades (con herramientas como el Open Source Snort), atacando el sistema y utilizando pruebas de concepto. El escanéo de puertos proporciona información acerca de que puertos en una máquina se encuentran escuchando. Cada puerto abierto es potencialmente vulnerable. Los más avanzados scanners intentan obtener que clase de software (sistema operativo y versión) se encuentra instalado. Los scanner de puertos más populares son: Nmap (http://www.insecure.org/nmap) y SuperScan (http://www.foundstone.com).

(4) Analizar las aplicaciones.

Utilizando análisis funcional/estructural, atacando la autenticación/autorización, atacando los datos y comunicación back-end y atacando los clientes. Se puede visualizar tráfico HTTP utilizando sniffers como Ethereal o Web proxies como: Fiddler (http://www.fiddlertool.com), Achilles (http://packetstormsecurity.nl/web/achilles-0-27.zip), WebProxy (http://www.foundstone.com). Para los test de penetración en aplicaciones Web se busca .css (cascading style sheets), .inc (ficheros include), (.xml, .xsl) datos xml / hojas de estilo xml, datos javacript (.js), datos de texto .txt, formularios (campos ocultos, campos de contraseña y atributos de longitud máxima. Se debe intentar inyectar código SQL en formularios, si el programador concatena la entrada de usuario con instrucciones SQL es posible comprometer una base de datos; se debe intentar generar errores insertando un carácter comilla ('). El cross site scripting permite a un atacante inyectar código script en páginas Web. Esto sucede cuando la aplicación directamente saca la entrada del cliente sin una codificación HTML correcta.



Esquemas teórico y real de un análisis de la infraestructura y máquinas

Diferencias entre un test de penetración y un procedimiento de auditoria. Elementos clave de un test de penetración y errores comunes.

Los test de penetración simulan a un atacante motivado durante un cierto tiempo específico y permiten obtener una instantánea de la seguridad actual de un sistema o de un proceso de negocios. Se utilizan enfoques de caja negra/caja blanca. En cambio la auditoría analiza los ficheros de configuración, la arquitectura y el código fuente, así mismo analiza si se cumple con la política de seguridad (planes y procedimientos operacionales).

Los principales elementos de un test de penetración son:

(1) Descubrir y explotar las vulnerabilidades a lo largo de la red.

(2) Influencia de las relaciones de confianza entre componentes.

(3) Acceso a la información crítica. Consideremos el siguiente ejemplo: después de explotar una vul-

Figura 6. Tres metodologías o modelos de test de penetración

nerabilidad en un servidor Exchange podemos recoger una lista de usuarios de correo electrónico y contraseñas válidas. Entonces utilizamos este servidor para atacar el servidor de base de datos en la DMZ (que no es visible desde fuera de la red interna). Uno de las exploits tuvo éxito y se obtuvo acceso a la cuenta del administrador del servidor, incluyendo acceso completo a todas las tablas de la base de datos de los clientes.

Un buen test de penetración debe:

(i) Cubrir todos los vectores de ataque relevantes.

(ii) Mostrar claramente el nivel de vulnerabilidad de los activos que pueden ser comprometidos.

(iii) Comprobar el sistema como un todo, incluyendo los mecanismos de defensa que incluye.

(iv) Documentar todas las actividades realizadas.

Los errores comunes de las organizaciones a la hora de realizar un test de penetración son:

(a) Limitar el test sólo a ejecutar un escaneado de vulnerabilidades. (b) Comprobar los componentes de forma aislada.

(c) No tener en cuenta los cambios de entorno de la compañía mientras el test se realiza.

(d) Pasar por alto las relaciones críticas como suministradores, socios corporativos, empresas de externalización y offshoring.

Los test de penetración se han convertido en una parte esencial de la valoración y mejora de la seguridad de una red de una empresa u organización. El objetivo de un test de penetración es valorar la seguridad global de una red intentando comprometer al sistema utilizando las técnicas de los atacantes. A menudo existe una confusión entre lo que es un escanéo de vulnerabilidades y un test de intrusión, el primero identifica el problema que puede ya haber ocurrido mientras que el segundo evalúa el sistema o red contra un ataque real. El test de penetración es activo en el sentido de atacar un sistema y medir su grado de preparación o resistencia al ataque. El escanéo de vulnerabilidades es pasivo, no se implica en el éxito de la intrusión y sólo muestra las vulnerabilidades potenciales. Una test de penetración es un intento autorizado de romper la arquitectura de un sistema empleando técnicas de atacante. Los resultados de un test de penetración van más allá de obtener los datos generados por un escanéo de vulnerabilidades. Con un test de penetración de hecho se explotan las vulnerabilidades de la red evaluada para intentar replicar los tipos de acceso que puede hacer un atacante y poder identificar que recursos serán expuestos y determinar si las inversiones actuales en seguridad detectan y previenen ataques reales.

ETAPAS DE UN TEST DE PENETRACIÓN

RECONOCIMIENTO/FOOTPRINT: Objetivo conseguir direcciones IP y nombres de dominio. Técnicas: búsqueda en fuente abierta , whois, transferencias de zona DNS. Usa herramientas como: USENet, motores de búsqueda como Google, Yahoo, nslookup, dig, networksolutions.com, dnspredict, dnswalk.

ESCANEO DE PUERTOS: Objetivo identificar servicios que escuchan, se enfoca en encontrar caminos de entrada de buena pinta, determina el sistema operativo utilizado. Técnicas: Nmap, barrido de pings. Usa herramientas como: Nmap (con capacidades de escanéo sigiloso, predicción del número de secuencia, análisis de la pila fingerprinting TCP, versión y tipo SO, IPv6), ping, hping, Nessus, Metasploit, Core Impact, Canvas, BindView Hacker Shield, Internet Scanner de Internet Security Systems.

ENUMERACIÓN: Objetivo identificar cuentas válidas de usuario, encuentra recursos o partes poco protegidas, identifica aplicaciones vulnerables en computadores destino, extrae información sobre recursos o partes de la red, nombres de usuario o grupos asignados a la red, última vez que se logeó un usuario, contraseña de usuario. *Técnicas*: Lista de cuentas de usuario, listado de file shares, identifica versiones de aplicaciones por fingerprinting utilizando el método banner grabbing. Usa herramientas como: Banner grabbing (netcat, telnet, rpcinfo, nessus), de Microsoft (dumpacl, sid2user), de Unix (showmount), etc.

GANAR ACCESO: Objetivo penetrar en el computador y establecer un punto de apoyo. Técnicas: robar contraseñas o escucha clandestina MITM, acceso por fuerza bruta, buffer overflow (copiar bytes de una posición de memoria a otra sin comprobar adecuadamente los límites). Usa herramientas como: tcpdump, readsmb de L0phtCrack, tftp (graba /etc/passwd en host Unix), pwdump2 (graba los hash de contraseñas en Windows2000 y Windows2003), keyloggers, spyware, rootkits, LKMs, metasploit, nessus, canvas, impact, script dirigidas a vulnerabilidades conocidas.

ESCALAR PRIVILEGIOS: Objetivo ganar control completo, obtener root o admin. Técnicas: craquear contraseñas, exploits publicados, telnet inverso, cron jobs, rebuscar información no protegida. Usa herramientas como: crack, L0phtcrack, john-the-ripper, rdist, getadmin, sechole, rootkits, scripts dirigidas a vulnerabilidades conocidas.

HURTAR: Objetivo recoger detalles de ficheros locales, usuarios, información oculta; obtener acceso a sistemas confiables, establecer un sitio pequeño para herramientas o aprovecharse de los ciclos de CPU. Técnicas: listar estructuras de directorio, shares, información del registro, buscar relaciones de confianza, buscar contraseñas en claro, revelar secretos LSA (Local Security Authority) via revelation. Usa herramientas como: revelation de snadboy soft., barok, rdist, rhosts, getadmin, sechole, scripts dirigidas a vulnerabilidades conocidas.

CUBRIR RASTROS: Objetivo ocultar la intrusión al administrador del sistema, destruir evidencias de cómo se gano el acceso, permanecer en sigilo para mantener la autorización de root o admin. Técnicas: limpiar logs, ocultar herramientas de hacking. Usa herramientas como: zap, invisible, cloak, stealth, rdist, rhosts, getadmin, sechole, scripts dirigidas a vulnerabilidades conocidas.

CREAR PUERTAS TRASERAS: Objetivo asegurar que el acceso pueda ser reobtenido, crear varias bachdoors en varias áreas del sistema. Técnicas: crear cuentas de usuario especiales, reemplazar aplicaciones con troyanos, modificar ficheros de startup, instalar monitores. Usa herramientas como: modificar Registry, Netcat, remote.exe, VNC (*Virtual Network Computing*), Sub7, NetBus 2.0 Pro, añadir cuentas a alias de mail, especialmente sysadmin.

Figura 7. Etapas de un test de penetración

Calidad de los test de seguridad. Enfoques para los test de seguridad.

La calidad de los test de seguridad se mide esencialmente por dos elementos:

1) Falsos negativos (algo no visto).

Es el problema más serio, expone al sistema a ataques. Las principales razones de los falsos negativos son: (i) Cobertura de los componentes comprobados: URL/parámetro/componente no visto; sección del código específico no alcanzado en el test (flujo, atributos, usuarios). (ii) Cobertura de los test: vulnerabilidad específica no comprobada, variante específica no comprobada. (iii) Calidad/capacidad de los test. Existen tres casos: (a) Test definido de forma deficiente: ataque no creado correctamente, resultados esperados no definidos adecuadamente. (b) Ejecución deficiente del test, carece de capacidad-competencia. (c) Tecnología cambiada / existen medidas de seguridad. El test requiere modificaciones, se necesitan técnicas de evasión, se requiere un análisis diferente de resultados.

Los principales problemas de cobertura son:

(i) Cobertura de datos de aplicación. Existen tres variantes:

(a) Problemas de crawling automáticos: enlaces prácticamente infinitos, enlaces del lado del cliente (JS/Ajax), adecuado flujo y datos de contexto.

(b) Disponibilidad: componentes de terceras partes no se pueden comprobar, código no disponible.

(c) Tamaño: demasiados URL/parámetros/código a cubrir, tiempo insuficiente.

(ii) Cobertura del test. Existen dos casos:

(a) Vulnerabilidad no comprobada: imposible comprobar (procedimiento lógico en vez de automatizado y fuerza bruta por manual), vulnerabilidad nueva descubierta (aún no actualizada), vulnerabilidad vista como insignificante, vulnerabilidad nunca vista antes, el test puede deteriorar la disponibilidad o la fiabilidad.

(b) Variante no comprobada. Existen dos casos: (a) Demasiadas variantes posibles (común con problemas de inyección). (b) Vulnerabilidad lógica muy dependiente de la aplicación actual.

2) Falsos positivos.

Crea problemas a las empresas, genera trabajo y esfuerzo redundante, crea desconfianza, no es necesariamente tecnológico (el fallo existe pero no posee amenaza real). Las principales razones de los falsos positivos son la calidad/capacidad de los test. Existen cuatro casos:

(a) Test definido de forma deficiente: resultado esperado no defi-

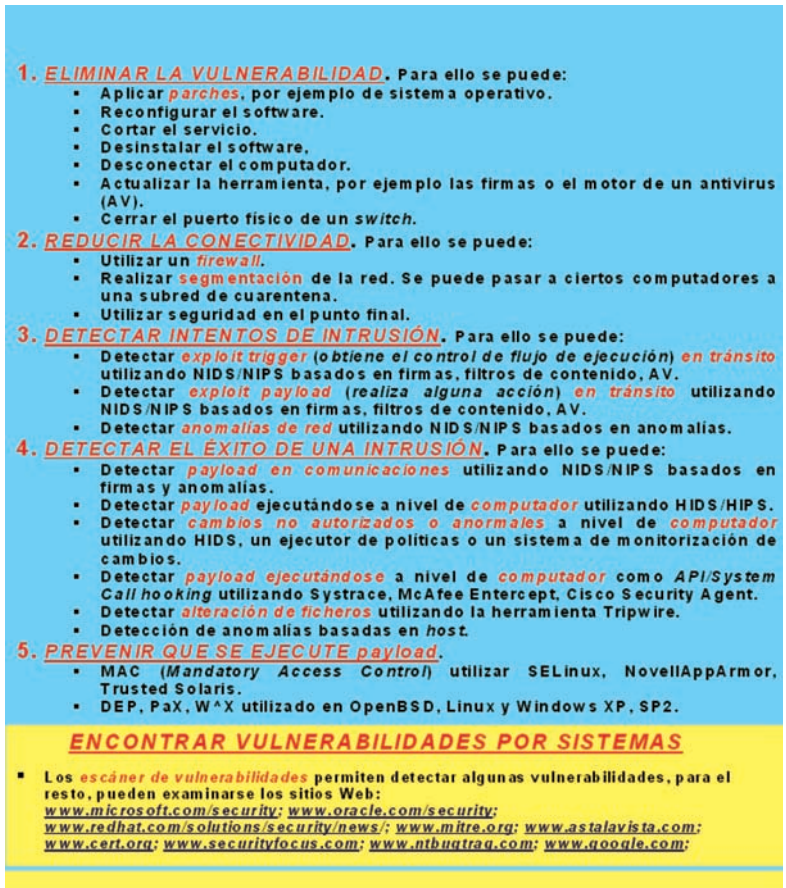


Figura 8. Técnicas para prevenir la explotación de una vulnerabilidad

nido correctamente, diferenciadores de validación definidos de forma defectuosa.

(b) Ejecución del test deficiente y carece de capacidad.

(c) Tecnología cambiada / existen medidas de seguridad: el resultado parece vulnerable, señuelos del tipo honeypots, soluciones de seguridad parcheadas, test incapaz de correlacionar a otros componentes (revisión del código).

(d) Vulnerabilidad tecnológica no amenaza. El test correlaciona al contexto del sistema

Las principales razones de los falsos negativos son:
(i) Cobertura de los componentes comprobados: URL/parámetro/componente no visto; sección del código específico no alcanzado en el test (flujo, atributos, usuarios).
(ii) Cobertura de los test: vulnerabilidad específica no comprobada, variante específica no comprobada.
(iii) Calidad/capacidad de los test. Existen tres casos:

(a) Test definido de forma deficiente: ataque no creado correctamente, resultados esperados no definidos adecuadamente.

(b) Ejecución deficiente del test, carece de capacidad-competencia.

(c) Tecnología cambiada / existen medidas de seguridad. El test requiere modificaciones, se necesitan técnicas de evasión, se requiere un análisis diferente de resultados.

Algunos enfoques para test de seguridad son:
(i) Caja negra/gris. Se utilizan escaners de vulnerabilidades de aplicación y test de penetración.
(ii) Caja blanca. Se utilizan analizadores de código estático y se revisa el código. Ω

Este artículo se enmarca en las actividades desarrolladas dentro del proyecto LEFIS-APTICE (financiado por Socrates 2005-2007. European Commission).

Bibliografia.

Areitio, J. and Areitio, A. "Tipificación de amenazas, identificación de contramedidas de seguridad en el ámbito de gestión de redes y sistemas". Revista Española de Electrónica. Nº 613. Diciembre 2005.

Areitio, J. "Análisis en torno a la auditoría de seguridad en tecnologías de la información y las comunicaciones". Revista Española de Electrónica. Nº 625. Diciembre 2006.

Areitio, J. "Identificación de la tecnología firewall para la protección de la seguridad de red". Revista Española de Electrónica. Nº 638. Enero 2008.

Areitio, J. "Seguridad de la Información: Redes, Informática y Sistemas de Información". Cengage Learning Paraninfo. 2008.

Kruse, W.G. and Heiser, J.G. "Computer Forensic: Incident Response Essentials". Addison Wesley. 2009.

Lang, D.T. "Introduction to Computer Forensic". CRC Press. 2005.

NIST: http://csrc.nist.gov/publications/drafts/security-testing.pdf.

Carr, H., Snyder, C. and Bailey, B. "Management of Network Security". Prentice-Hall. 2008.

NcNab, C. "Network Security Assessment". O'Reilly. 2007.

Foreman, P. "Vulnerability Management". Auerbach Publications. 2009.

Tiller, J.S. "The Ethical Hack: Testing Security Measures Through the Act of Exploitation". Auerbach Publishers, Inc. 2004.

Schiffman, M., Pennington, B., Pollino, D. and O'Donnell, A. "Hackers Challenge 2: Test Your Network Security and Forensic Skills". MCgraw-Hill. 2002.

Hoopes, J. "Virtualization for Security: Including Sandboxing, Disaster Recovery, High Availability, Forensic Analysis and Honeypotting". Syngress. 2008.

Singh, A. "Vulnerability Analysis and Defence for the Internet". Springer. 2007.

Sammes, A.J. and Jenkinson, B. "Forensic Computing". Springer. 2007.

Dowd, M., McDonald, J. and Schuh, J. "The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities". Addison-Wesley. 2006.

Buchanan, W.J. "Introduction to Network Forensic". Auerbach Publications. 2009.

Lee, W., Wang, C. and Dagon, D. "Botnet Detection: Countering the Largest Security Threat". Springer. 2007.

Basta, A. and Halton, W. "Computer Security and Penetration Testing". Delmar Thomson Learning. 2007.

Carrier, B. "File Systems Forensic Analysis". Addison-Wesley. 2005.

Kizza, J.M. "Computer Network Security". Springer-Verlag. New York, Inc. 2005.