Departamento de Arquitectura y Tecnología de Computadores
UNIVERSIDAD DE GRANADA

TEMA 3. SEGURIDAD (2ª PARTE)

Centro de procesamiento de datos

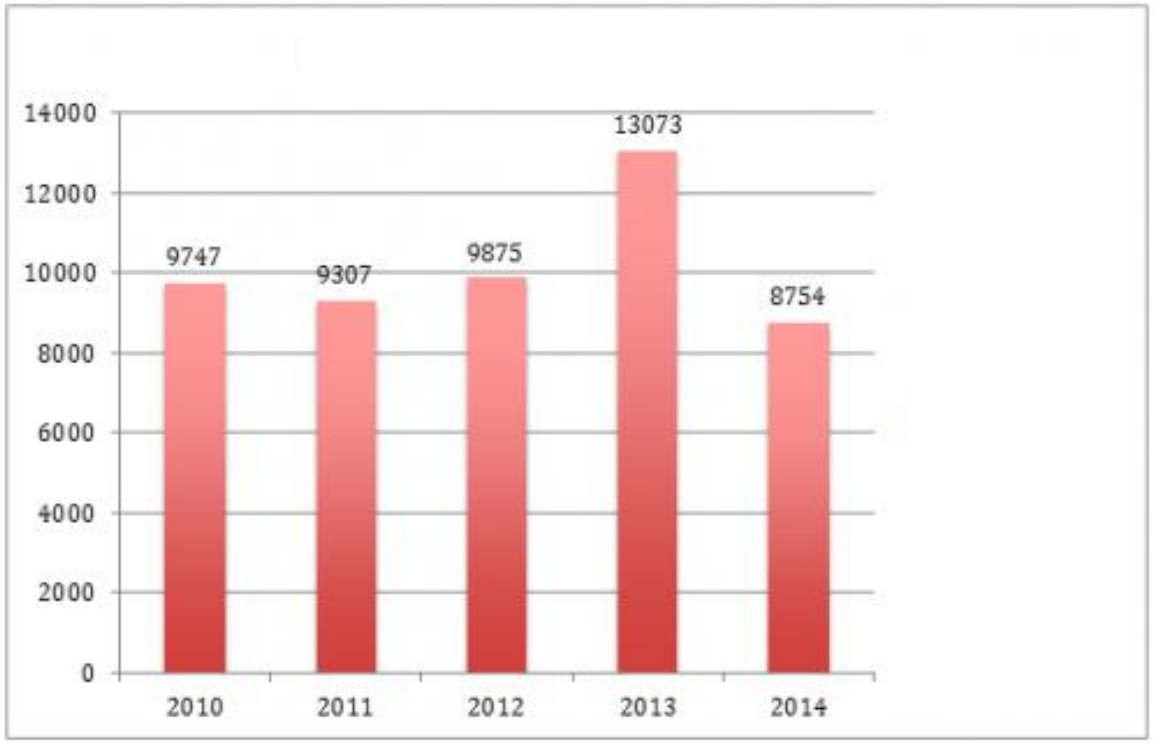
Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada

¿Por qué las aplicaciones son inseguras?

- Pueden recibir múltiples ataques
  - Ataques generales
  - Ataques dirigidos
  - Disponibilidad: DOS / DDOS…
  - Integridad: Defacement…
  - Confidencialidad: Evesdropping…
- Reconocimiento:
  - Port scanning, OS Fingerprint…
  - Ataques fuzzing: Testeo funcional del protocolo
- Exploits: es un programa malicioso, que trata de forzar alguna deficiencia o vulnerabilidad de otro programa
  - Desbordamiento de búfer (Buffer overflow
  - Desbordamiento de pila (Stack overflow)
  - Race conditions
  - Escalada de privilegios
  - Pobre verificación de identidad

Vulnerabilidades: BBDD

- **CVE:**
  - Common Vulnerabilities and Exposures.
  - http://cve.mitre.org
  - Ej: CVE-2011-2300



Difusión de vulnerabilidades

- **US-CERT:**
  - United States Computer Emergency Readiness Team
  - http://www.uscert.gov/
- **NSD:**
  - National Vulnerability Database
  - http://web.nvd.nist.gov/view/vuln/search
- **INTECO-CERT:**
  - Centro de respuestas a incidentes de seguridad TIC
  - http://cert.inteco.es
- **Portales de empresas sobre seguridad:**
  - http://securitytracker.com
  - https://www.redhat.com/security/data/metrics/
  - http://www.ubuntu.com/usn/
  - http://technet.microsoft.com/es-es/security
  - http://www.securityfocus.com/
  - http://vigilance.fr/

Desbordamiento de búfer y desbordamiento de pila

- **Desbordamiento de buffer:** es un error de software que se produce cuando se copia una cantidad de datos sobre un área que no es lo suficientemente grande para contenerlos
  - Ej: C: con función strcpy (solución strncpy), gets

- **Desbordamiento de pila:** La pila es una zona de memoria donde se almacenan variables locales y direcciones de retorno de las llamadas a función. Si la aplicación no controla del todo las variables que maneja en la pila se puede insertar código y forzar su ejecución

- Cuando se dispone del código fuente de las aplicaciones se puede saber más fácilmente si es posible aprovechar estos fallos de seguridad.

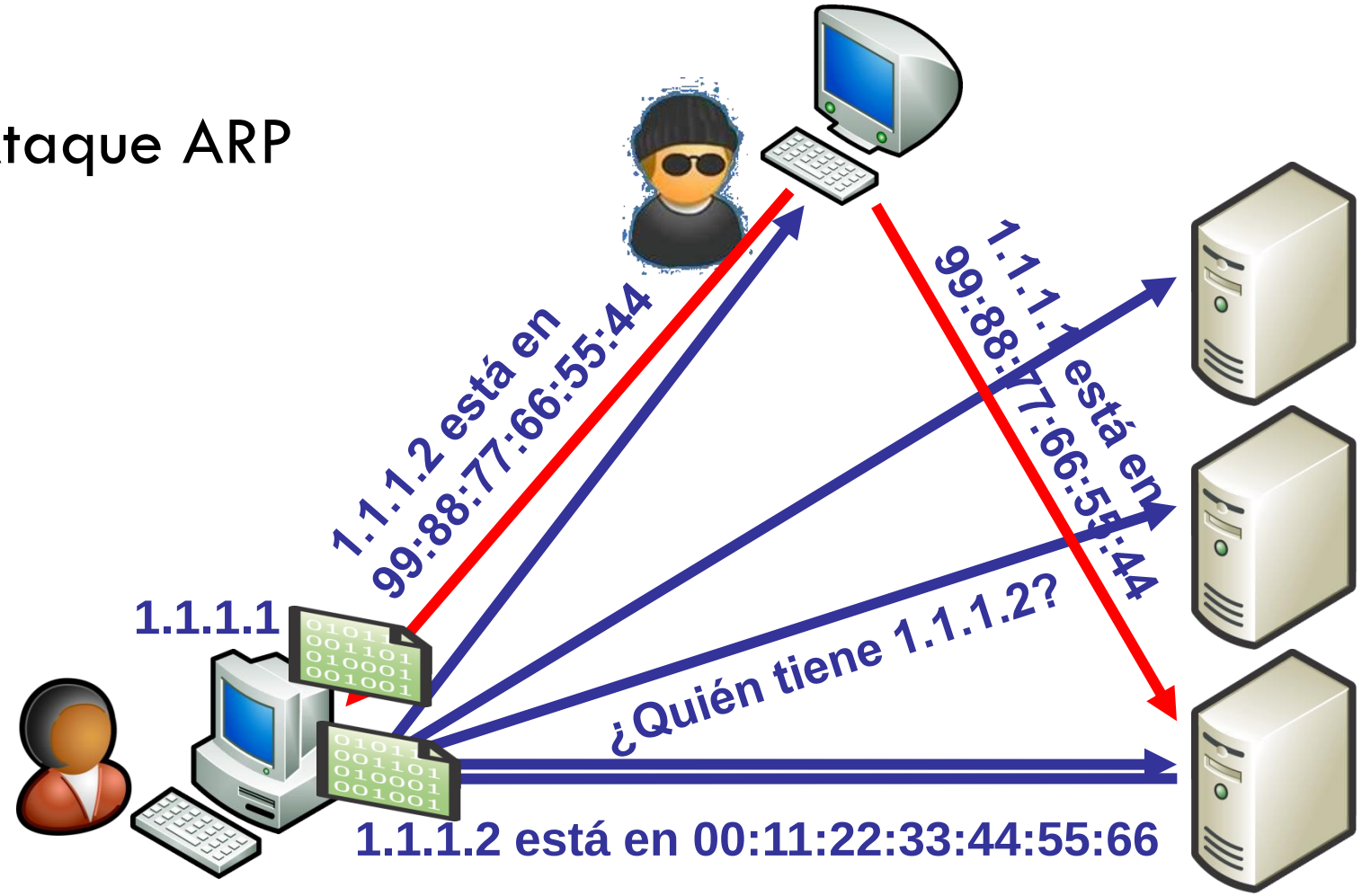
Spoofing

Uso de técnicas de suplantación de identidad

- **IP SPOOFING:** Suplantación de IP. Sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar
- **ARP SPOOFING:** Falsificación de tabla ARP. Construcción de tramas de solicitud y respuesta ARP modificadas con el objetivo de falsear la tabla ARP (relación IP-MAC) de una víctima y forzarla a que envíe los paquetes a un host atacante en lugar de hacerlo a su destino legítimo.
- **DNS SPOOFING:** Suplantación de identidad por nombre de dominio
- **WEB SPOOFING:** Suplantación de una página web real
- **MAIL SPOOFING:** Suplantación en correo electrónico de la dirección e-mail de otras personas o entidades

Ataques MitM:"Man in the Middle"

- Ataque ARP



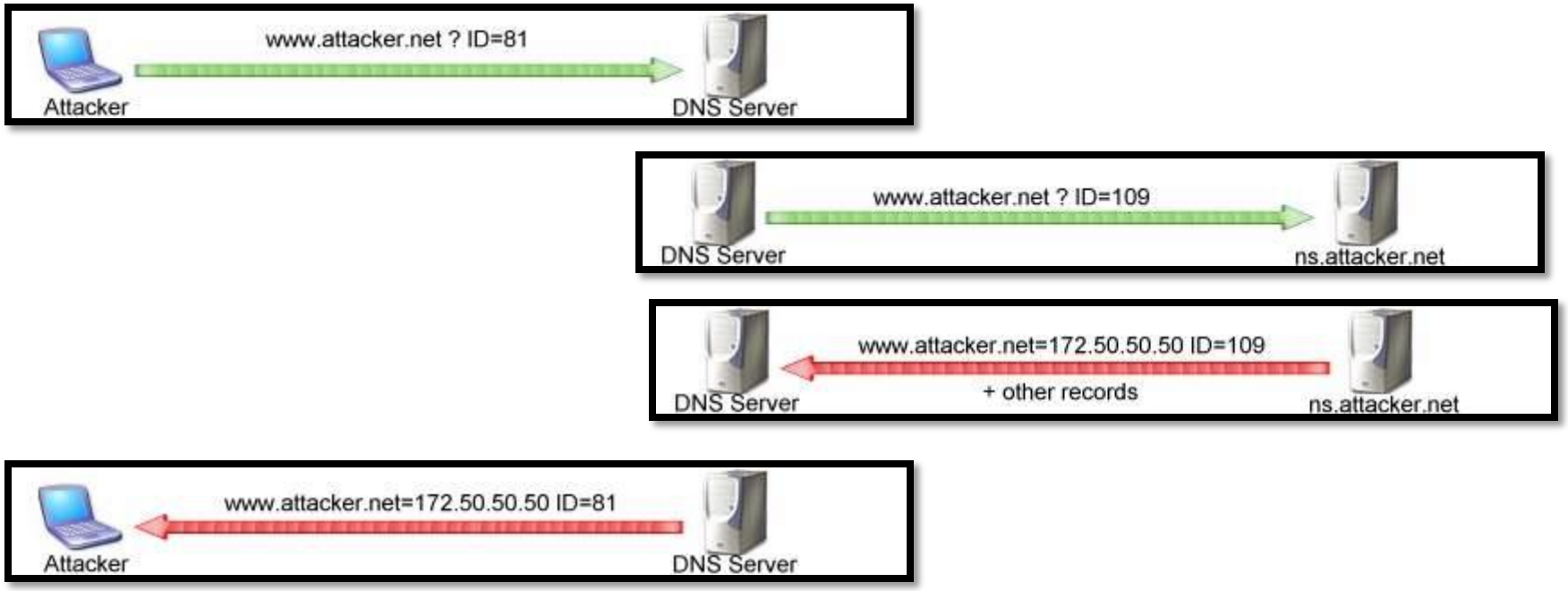
Cain & Abel

- http://www.oxid.it/index.html
- Herramienta Windows para:
  - Captura paquetes
  - Análisis MAC
  - Envenamiento ARP
  - Ataques DNS
  - Manipulación de certificados
  - Captura y ataque de claves
  - Grabación VoIP
  - Análisis protocolos enrutamiento
  - Abel: captura remota
- Winrtgen generador de tablas Rainbow:

LM, FastLM, NTLM, LMCHALL, HalfLMCHALL, NTLMCHALL, MSCACHE, MD2, MD4, MD5, SHA1, RIPEMD160, MySQL323, MySQLSHA1, CiscoPIX, ORACLE, SHA-2 (256), hash SHA-2 (384) y hash SHA-2 (512).

Ataques DNS Spoofing

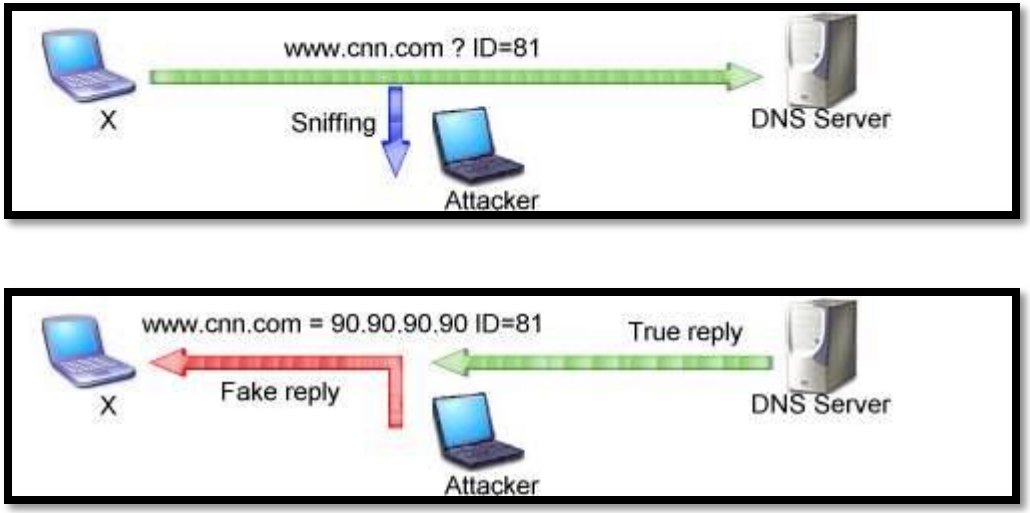
Envenenamiento cache DNS:



Solución: Los servidores DNS pueden utilizar certificados para comprobar identidad del DNS que responde

Ataques DNS Spoofing

- DNS ID spoofing:

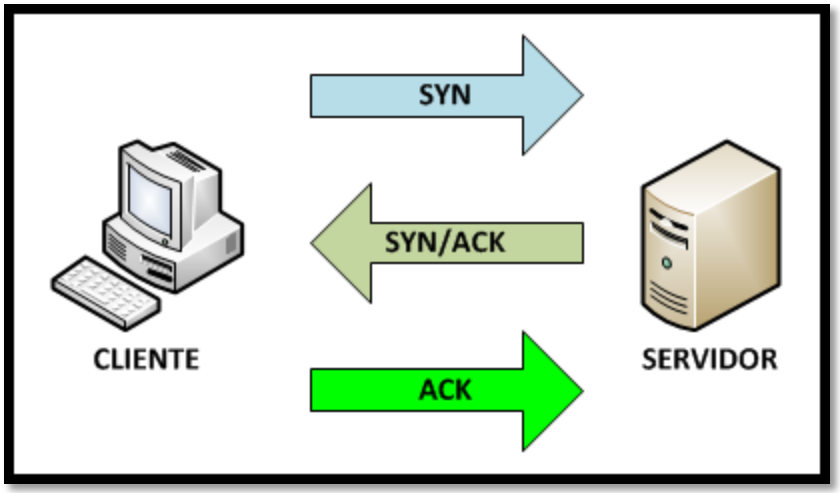


- El DNS falso debe responder antes que el verdadero

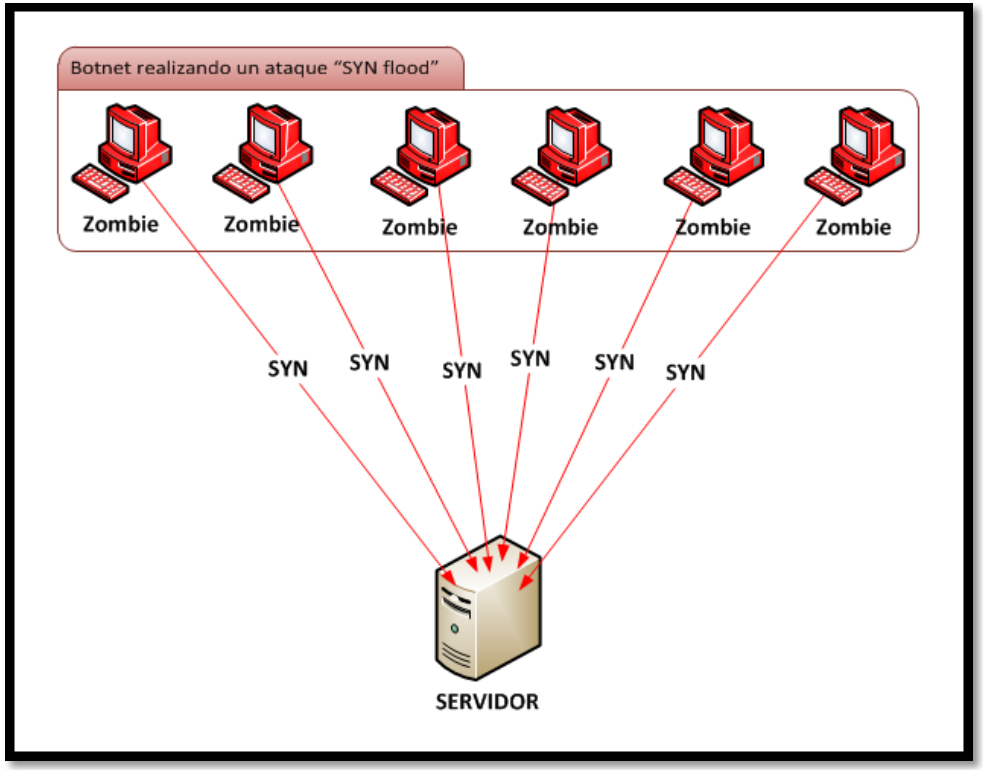
DoS: Denegación de Servicio

- Saturar a un ordenador enviándole masivamente peticiones de algún tipo de servicio, lo que le requiere un tiempo de CPU que acaba ralentizando o bloqueando el resto de servicios/procesos.
- Para conseguir un efecto mayor a veces se utiliza DDoS(Distributed Denial of Service) donde un conjunto de máquinas (zombies) ejecutan el código atacante y consiguen además agotar el ancho de banda. Para aumentar el ataque se utilizan ***botnets***.
  - **TCP SYN Flood**
  - TCP SYN-ACK Reflection Flood (DRDoS)
  - TCP Spoofed SYN Flood
  - TCP ACK Flood
  - TCP IP Fragmented Attack
  - HTTP and HTTPS Flood Attacks
  - INTELLIGENT HTTP and HTTPS Attacks
  - ICMP Echo Request Flood
  - UDP Flood Attack
  - DNS Amplification Attacks
- ddossim : para probar "en laboratorio" un ataque DDoS

TCP SYN Flood



Secuencia normal



Ataque SYN

Ataques DHCP

- Envío paquetes RAW DHCP
- Envío DoS paquetes DISCOVER (agotamiento pool IP)
  - DHCP snooping en los conmutadores lo detectan
- Servidores falsos de DHCP
  - El conmutador puede filtrar puertos de servidor válidos
- Envío DoS paquetes RELEASE (liberación IP asignadas)

NAT Pinning

- http://samy.pl/natpin/
- Si un usuario accede a una página web maliciosa ésta puede devolverle un Javascript al cliente que le indique a su enrutador que debe reencaminar puertos
- El javascript utiliza una petición IRC con DCC CHAT .

Seguridad Web

OWASP:(Open Web Application Security Project)

- https://www.owasp.org
- Top10
  - http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf
  - https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf
- Herramientas:
  - **Nikto**: http://www.cirt.net/nikto2 (GPL)
  - Acunetix ($$)
  - Otros analizadores: Paros proxy, WebScarab, WebInspect

OWASP Top 10 (2010)

OWASP Top 10 – 2010 (Previo)	OWASP Top 10 – 2013 (Nuevo)
A1 – Inyección	A1 – Inyección
A3 – Pérdida de Autenticación y Gestión de Sesiones	A2 – Pérdida de Autenticación y Gestión de Sesiones
A2 – Secuencia de Comandos en Sitios Cruzados (XSS)	A3 – Secuencia de Comandos en Sitios Cruzados (XSS)
A4 – Referencia Directa Insegura a Objetos	A4 – Referencia Directa Insegura a Objetos
A6 – Defectuosa Configuración de Seguridad	A5 – Configuración de Seguridad Incorrecta
A7 – Almacenamiento Criptográfico Inseguro – Fusionada A9→	A6 – Exposición de Datos Sensibles
A8 – Falla de Restricción de Acceso a URL – Ampliada en →	A7 – Ausencia de Control de Acceso a las Funciones
A5 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)
<dentro de A6: – Defectuosa Configuración de Seguridad>	A9 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Redirecciones y reenvíos no validados	A10 – Redirecciones y reenvíos no validados
A9 – Protección Insuficiente en la Capa de Transporte	Fusionada con 2010-A7 en la nueva 2013-A6

Inyección SQL

- Se “inyecta" código SQL "invasor" dentro de otro código SQL para alterar su funcionamiento normal, y hacer que se ejecute maliciosamente el código "invasor" en la base de datos.
- Ej: código susceptible de ser atacado:

```
consulta := "SELECT * FROM usuarios WHERE nombre = '" + nombreUsuario +
   "';"
```

- Texto a insertar en el campo del formulario:

"Alicia'; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE '%“

Los lenguajes incorporan funciones para filtrar el contenido mailicioso: Ej.PHP:

```
$query_result = mysql_query (
 "SELECT * FROM usuarios WHERE nombre = \"" .
   mysql_real_escape_string($nombre_usuario) . "\""

);
```

SQLMAP

- http://sqlmap.org/
- sqlmap es una herramienta de pruebas de penetración de código abierto que automatiza el proceso de detectar y explotar los errores de inyección SQL en servidores de bases de datos.
  - MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB and HSQLDB
- Puede identificar a la BBDD, extraer información útil para identificar vulnerabilidades, acceder al sistema de archivos y ejecutar órdenes remotas.
- Ej: https://www.youtube.com/watch?v=RsQ52eCcTi4

Inyección SQL: Contramedidas

- Fortificación de Servidor Web:
  - Códigos de error.
  - Restricción de verbos, longitudes, etc..
  - Filtrado de contenido HTTP en Firewall.

- Fortificación de SGBD:
  - Restricción de privilegios de motor/usuario de acceso desde web.
  - Aislamiento de bases de datos.

Cross-Site Scripting (XSS)
Robo de sesiones

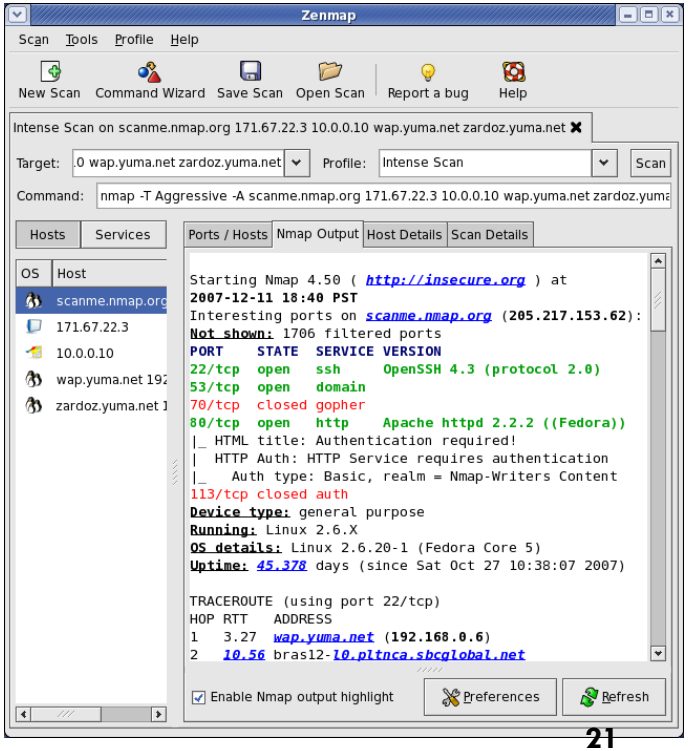
- Realizar un script que llamase a una página alojada en nuestro servidor pasándole la cookie
- Este Script se colaría en el servidor de la victima aprovechando un punto vulnerable a XSS
- Cuando un usuario esté accediendo en el servidor y ejecute el script se enviará a nuestro servidor el contenido de la cookie
- Una vez que la página obtiene la cookie (almacenandola por ejemplo en un fichero) mediante programas como Odysseus se puede hacer una llamada al servidor pasándole la cookie original
- Esta cookie es válida para robar la sesión sólo mientras el usuario no cierre la sesión

Ataques HTTP específicos a CMS

- Muchas plataformas Web usan CMS
- Joomscan
- WPScan

Herramientas de análisis de (in)seguridad. NMAP

- **Nmap:** Identificación de SO y puertos. http://nmap.org/
- Zenmap: GUI para Nmap. http://nmap.org/zenmap/
- Soporta múltiples análisis: conexión TCP, TCP SYN (medio abierto), ICMP, FIN, Barrido ACK, FIN, Null scan,…
- detección del S.O. remoto, análisis indetectable, análisis paralelos, fragmentación, análisis RPC directo, …
- Análisis: nmap –T5 -sV –O localhost
- Análisis SYN: nmap –sS (computador)
- Análisis TCP: nnamp –sT (computador)
- Análisis SCTP INIT: nmap –sY
- Análisis servicios / versión: nmap –sV
- Evasión de cortafuegos/IDS



Herramientas de análisis de (in)seguridad (II)

- **OSSIM: (Open Source Security Information Management)**
- http://communities.alienvault.com/
- Incluye:
  - **Arpwatch**: detección de anomalías mac
  - **P0f**: detección pasiva de OS
  - **Pads** : detección de anomalías en servicios
  - **Nessus**:análisis de vulnerabilidad correlacionando (IDS y escáner seguridad)
  - **Snort** : IDS (también utilizado con nessus)
  - **Spade** : detector de anomalía en estadísticas de tráfico de paquetes.
  - **Tcptrack** : Extrae información de sesiones TCP, ayuda a confirmar ataques.
  - **Ntop** : genera una base de datos con información de red para detectar anomalías
  - **Nagios** : Monitoriza ordenadores y servicios
  - **Osiris** : Un HIDS
  - Metasploit:
  - nexpose
- Open Threat Exchange

  http://www.alienvault.com/alienvault-labs/open-threat-exchange

Yersinia

- http://www.yersinia.net
- Herramienta que aprovecha debilidades en diversos protocolos de red:
  - Spanning Tree Protocol (STP)
  - Cisco Discovery Protocol (CDP)
  - Dynamic Trunking Protocol (DTP)
  - Dynamic Host Configuration Protocol (DHCP)
  - Hot Standby Router Protocol (HSRP)
  - IEEE 802.1Q
  - IEEE 802.1X
  - Inter-Switch Link Protocol (ISL)
  - VLAN Trunking Protocol (VTP)

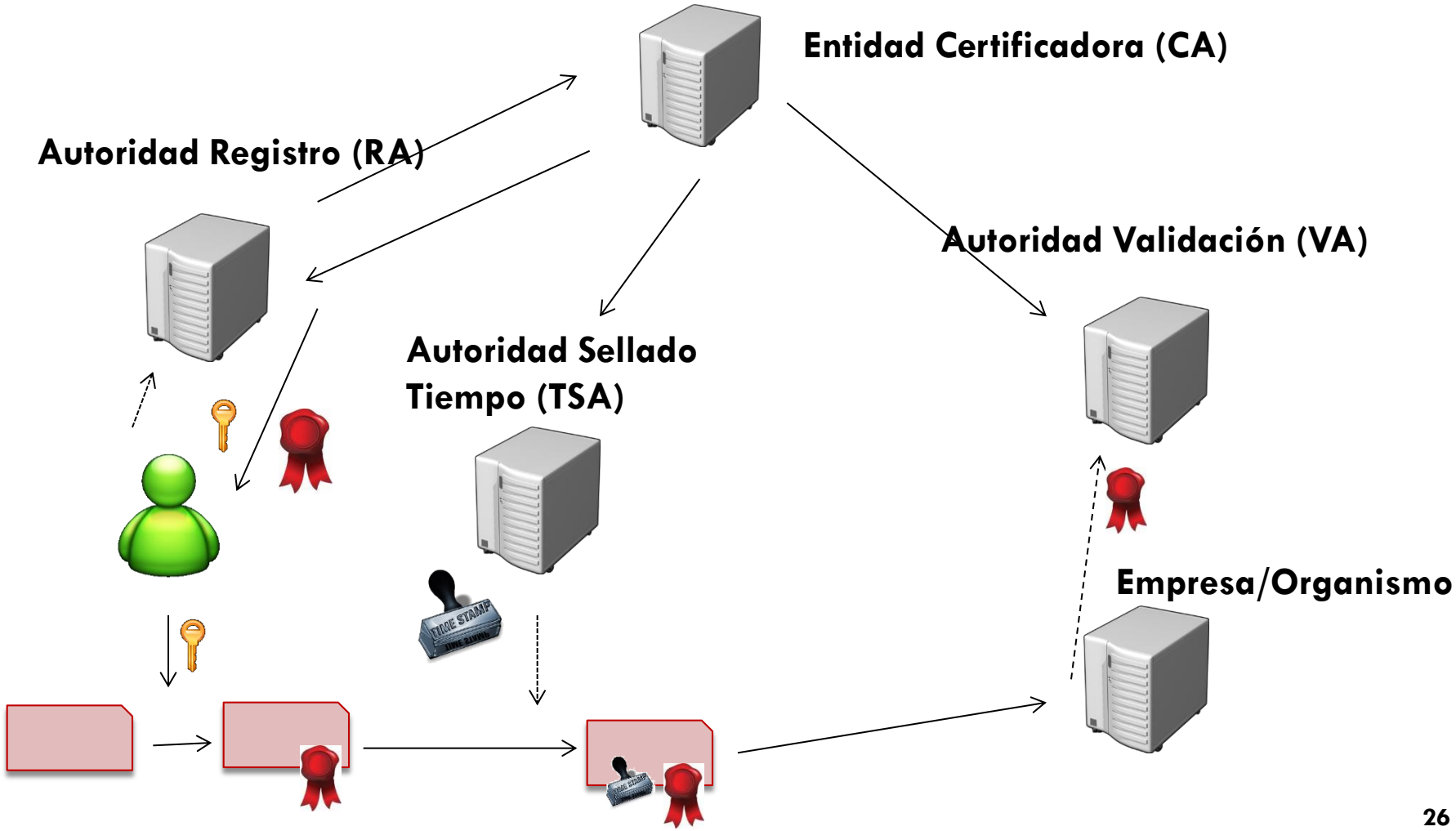
Ataques redes VoIP

- Descubrir contraseñas: SIPCrack
  - sipdump: captura hashes de la autenticación
  - Sipcrack: contraseña por fuerza bruta (diccionario)
- Suplantación identidad
- Desregistro usuarios, redirección de llamadas
- Herramientas: SiVUS, Cain (Rediccionamiento/análisis)
- Interceptar llamadas:
  - Oreka, ag, vomit,…
- Ataques DoS:
- Ataque terminales
- Ataques Fuzzing: crear paquetes o peticiones especialmente malformadas
  - Ohrwurm, Asteroid

PKI (Public Key Infrastructure)

- Garantiza la autentificación de usuarios basada en la "confianza" de la firma de una entidad.
- Permite el cifrado de información con llaves asimétricas.
- Estructura
  - Autoridad certificadora (CA)
    - Emite y revoca certificados. Es la entidad de confianza que da legitimidad a la relación de una clave pública con la identidad de un usuario o servicio.
  - Autoridad de registro (AR)
    - Verifica el enlace entre los certificados (entre la clave pública del certificado) y la identidad de sus titulares
  - Autoridad de validación (VA: Validation Authority)
    - Comprueba la validez de los certificados digitales.
  - Autoridad de sellado de tiempo (TSA: TimeStamp Authority)
    - Firma documentos para de probar que existían antes de un determinado instante de tiempo
  - Repositorios
    - Son las estructuras encargadas de almacenar la información relativa a la PKI.
    - Repositorio de certificados
    - Repositorio de listas de revocación de certificados.(CRL)

Flujo PKI


Entidad Certificadora (CA)
Autoridad Registro (RA)
Autoridad Validación (VA)
Autoridad Sellado Tiempo (TSA)
Empresa/Organismo
TIME STAMP

¿Por qué utilizar certificados en las comunicaciones y aplicaciones?

- Permite autentificar
  - Usuarios, nodos, recursos,…
  - Basado en clave pública. Robusto.

- Control individual
  - Mecanismos sencillos Claves propias, no comunes
  - Revocación
  - Basado en la confianza de la entidad certificadora.

Contenido de un certificado

- Formato X509v3
  - Ej: http://pki.cica.es/cacert/
- Información
  - Nombre, organización, dirección, …
- Validez
- Entidad certificadora (CA)
  - Clave pública , clave privada
- Autenticación
  - Clave púbica, clave privada
  - Estas claves pueden estar firmadas por una CA
- Firma electrónica
  - Clave pública , clave privada

Almacenamiento de certificados en ficheros

- Formato PEM (.pem, .crt, .cer, .key)
  - Común en entidades de certificación
  - Codificado en ASCII, Base64
  - Extensiones Varios certificados y la clave privada pueden ir en el mismo fichero, aunque algunas plataformas requieren en ficheros separados
  - Generalmente ficheros .key almacenan las claves privadas
- Formato DER (.cer, .der)
  - Forma binaria del forma PEM
  - Utilizado en entornos Java

Almacenamiento de certificados en ficheros (II)

- Formato P7B / PKCS#7 (.p7b, .p7c)
  - Sólo contiene certificados y cadenas, no claves
  - Codificado en ASCII, Base64
  - Windows, Tomcat
- Formato PFX /PKCS#12 (.pfx, .p12)
  - Almacenan certificados de servidor, intermedios y claves en un fichero que puede cifrarse
  - Formato binario.
  - Utilizado en Windows para importar y exportar certificados y claves privadas
- openssl permite fácilmente cambiar de formato

Almacenamiento de certificados (III)

- Tarjetas inteligentes (PKCS#11)
  - http://www.rsa.com/rsalabs/node.asp?id=2133
- Criptoki: API gestionar todos los recursos de forma transparente.
- OpenSC:
  - Biblioteca para utilizar smart cards.
  - http://www.opensc-project.org/pam_pkcs11/
  - Implementa PKCS#11 API.

Firma digital ciega

- Permite firmar un mensaje sin conocer su contenido (notario ciego).
  - Se pueden revocar mensajes.
- Aplicaciones:
  - Dinero electrónico. Otros modelos: Bitcoin.
  - Voto electrónico. Otros modelos:
    - Sensus; Esquema propuesto por Lee y Lin; Esquema propuesto por Lin, Hwang-Chang. (Basado en el esquema de firma digital ElGamal)
- Problema:
  - Existencia de listas de revocación que podrían crecer enormemente.

OpenSSL

- http://www.openssl.org
- API que incluye funciones de cifrado
- Utilizado en HTTPS, e-Comercio, stunnel, y gran cantidad de programas que necesitan establecer conexiones seguras
- Implementa:
  - (SSL v2/v3)
  - Transport Layer Security (TLS v1)
- Incluye un programa (opensll)que permite:
  - Creación de parámetro de claves RSA, DH y DSA
  - Creación de certificados X.509, CSRs y CRLs
  - Cálculo de "Message Digests" MD4,MD5
  - Cifrado y descrifrado con "Ciphers"
  - Test Cliente y Servidor SSL/TLS
  - Procesamiento de correo S/MIME firmado o cifrado

OpenSSL

- Obtener el digest MD5
  - openssl dgst –md5 /mifichero
  - openssl md5 /mifichero
- Cifrar
  - openssl bf –e –in /fichero_orig –out /fiche_cif
  - openssl enc –base64 –e –aes128 –in /fichero_orig –out /fiche_dest
- Descifrar
  - openssl bf –d –in /fich_cif –out /fich_orig
- Generar contraseñas
  - echo clave | openssl passwd –stdin -1
- Generar secuencia aleatoria
  - openssl rand -out random 1000
- Medida de prestaciones
  - openssl speed

Creación de certificados con OpenSSL

- Crear la entidad certificadora (CA):
  - openssl req –nodes –new –x509 –keyout mi-ca.key –out mi-ca.crt –days 3650
  - Crea certificado: mi-ca.crt
  - Clave privada: mi-ca.key
- Crear los certificados en los extremos:
  - openssl req -nodes -new -keyout office.key -out office.csr
  - openssl ca -out office.crt -in office.csr
- Verificar integridad
  - Openssl verify –CAfile cacert.pem office.crt

- Ficheros:

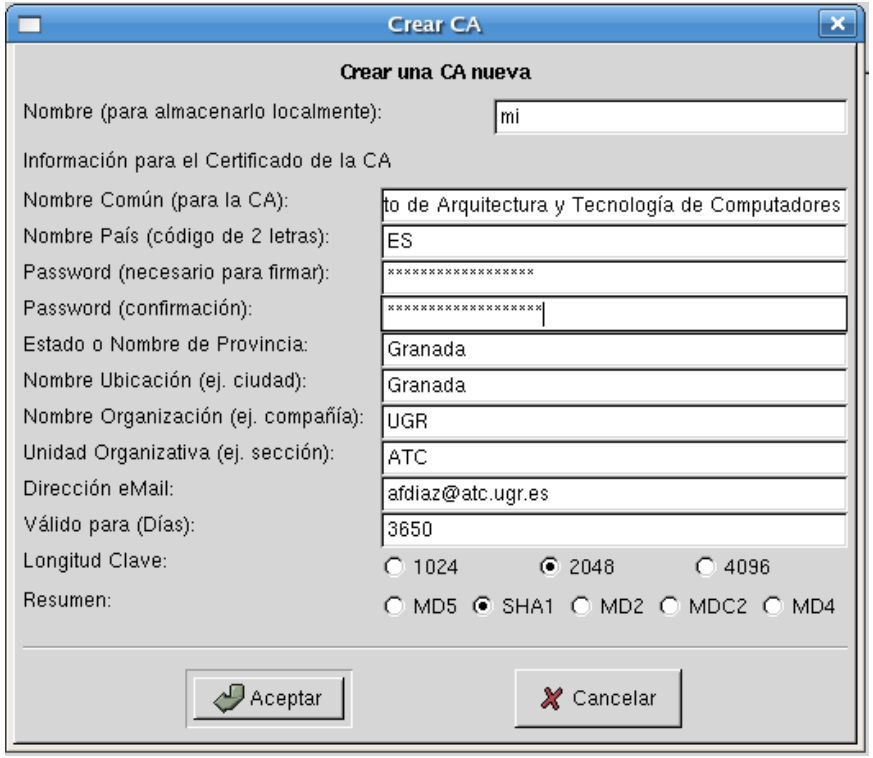
  .crt: Certificado

  .csr: (Certificate Sign Request) Certificado sin firmar por CA.

  .key: Clave privada

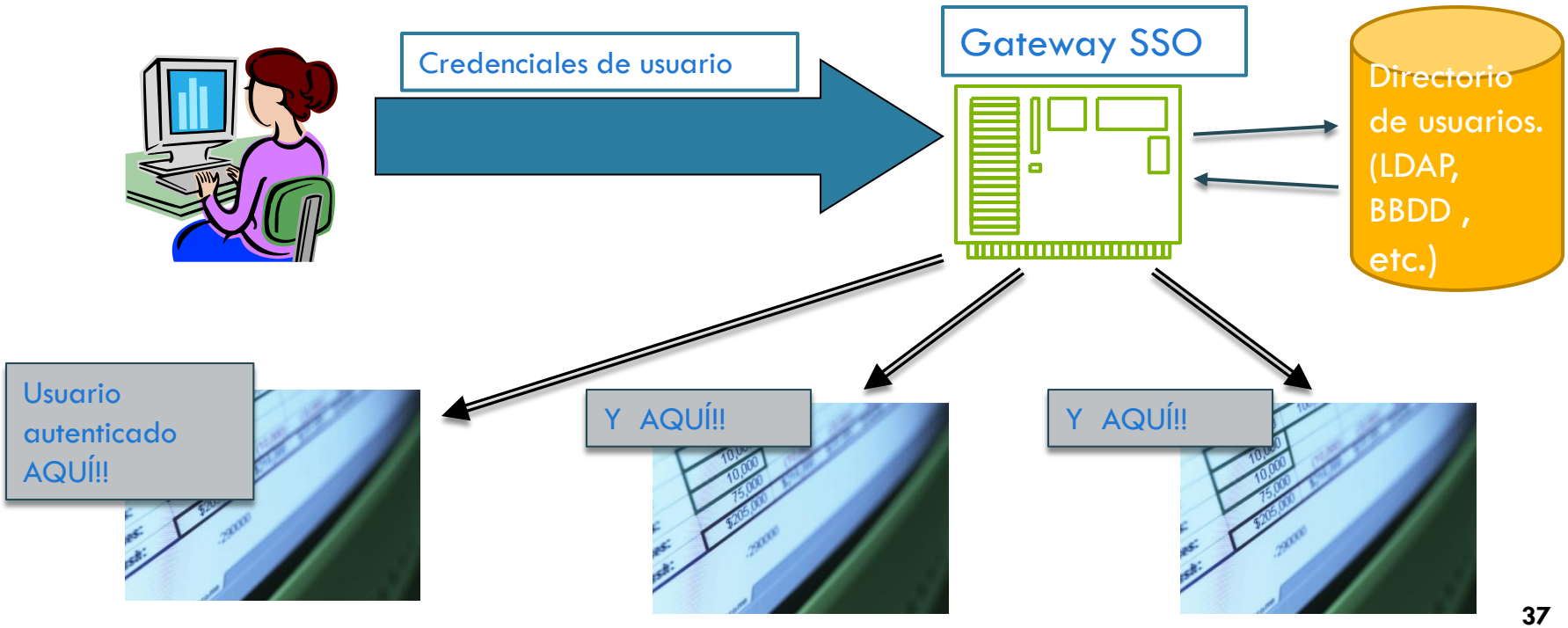
tinyCA2

- http://tinyca.sm-zone.net/
- Permite gestionar certificados de forma sencilla
- Permite crear y gestionar SubCAs
- Creación y eliminación de certificados S/MIME x509
- Peticiones PKCS#10 pueden ser importadas y firmadas
- Crear y utilizar claves RSA y DSA
- Exportar como: PEM, DER, TXT y PKCS#12
- Certificados de servidor
  - Apache, Postfix, OpenLDAP, Cyrus, FreeS/WAN, OpenVPN, OpenSWAN, FreeRadius
- Certificados de cliente
  - Netscape, Konqueror, Opera, Internet Explorer, Outlook (Express) y FreeS/WAN
- Gestión de CRLs (Certificate Revocation List)
  - Pueden exportarse como: PEM, DER y TXT
- Otros : gnoMint



Single Sign On (SSO)

- Procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación. Ej: JOSSO



Retos en SSO

- Autorización
  - Hay que determinar qué recursos son accesibles por qué usuarios
- Soporte a aplicaciones distribuidas
  - Recursos pueden estar distribuidos en distintas máquinas
  - No tienen porque estar en la misma máquina que el mecanismo de SSO
- Aplicaciones de distintos fabricantes
  - Las aplicaciones pueden ser web y/o stand-alone
  - Implementadas distintos lenguajes
  - El sistema puede estar formado por distintas plataformas
- Gestión de la salida (*logout*)
  - Asegurar que la salida en una aplicación garantiza la salida en el resto

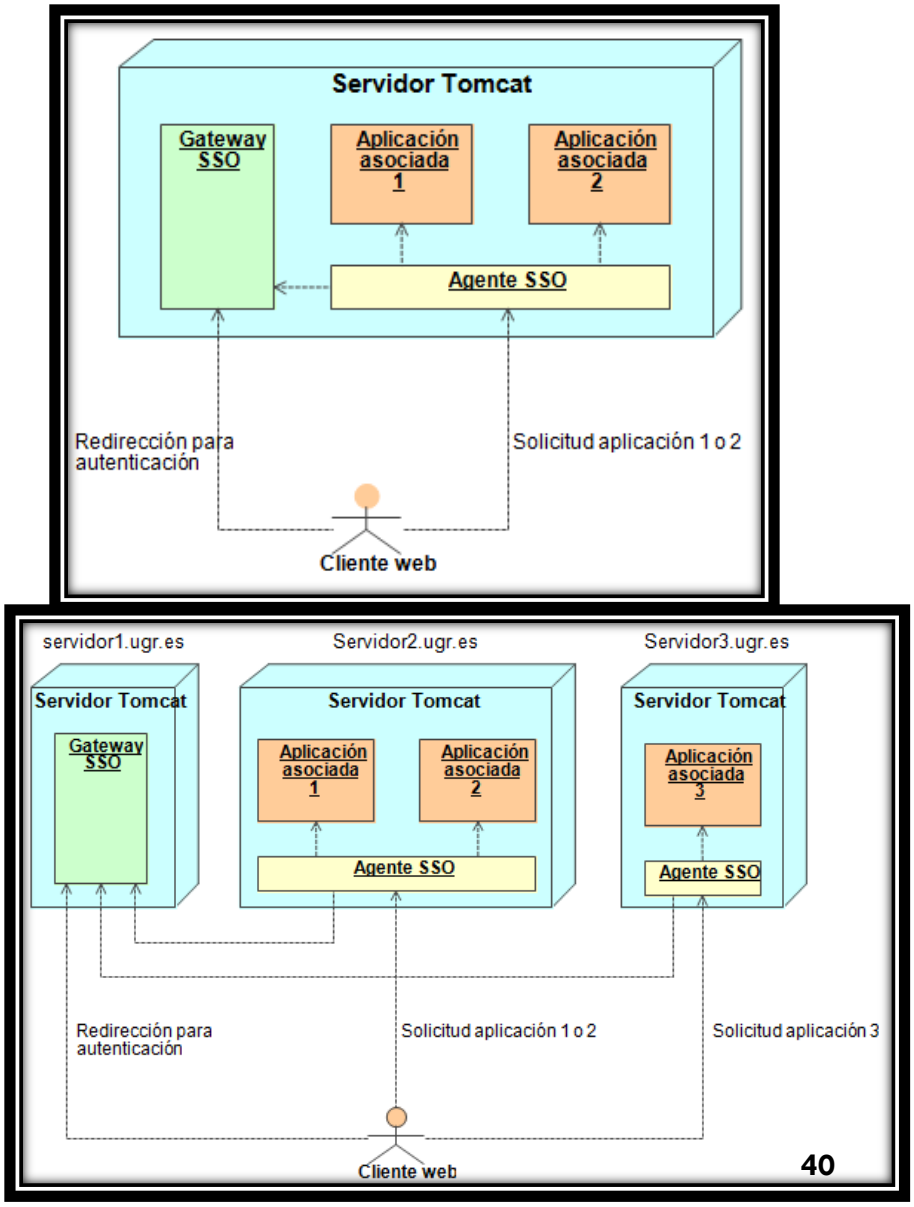
Autentificación basada en Tokens

- El Token es el elemento resultante de la autenticación.
- Contendrán información como:
  - **Nombre identificador de usuario (*username)*:** Permitirá al usuario identificarse ante cualquier aplicación.
  - **Roles del usuario**: determinan los privilegios de acceso del usuario.
  - Cualquier otra información relevante para el acceso a los distintos elementos del sistema.
- Tipos:
  - **Token en el cliente:**
    - cookie local (más susceptibles a ataques).
  - **Token en el servidor:**
    - sesiones en el servidor -> más óptimos y seguros.

Modelos SSO (I)

- **Modelo simple:**
  - El Gateway SSO y el Agente SSO de aplicaciones asociadas están desplegados en el mismo servidor de aplicaciones.

- **Modelo plano basado en dominio DNS:**
  - Todos los servidores de aplicaciones deben estar asignados al mismo dominio DNS.
  - El token de identificación es una cookie de sesión HTTP asignada al dominio compartido por todos los servidores



OpenID

- Autenticación en múltiples sitios web con un único registro.
- Modelo descentralizado:
  - Usuarios
  - Proveedor de identidad
  - Aplicación WEB que ofrece el servicio final.
- Inconvenientes:
  - ¿Una única identidad?
  - Ante una posible vulnerabilidad más "puertas abiertas" para *phishing*.

Seguridad en objetos distribuidos

- DCOM
- .NET Remoting
  - Tecnología de objetos distribuidos sucesora de DCOM
  - Forma práctica de administrar conversaciones RPC sincrónicas y asincrónicas cliente/servidor a través de dominios de aplicación.
  - Canales HTTP (protocolo SOAP) y TCP (carga binaria).
- CORBA
- Java RMI (RMISecurityManager)
  - El modelo de seguridad de Java incluye un gestor de seguridad que protege a las aplicaciones de la descarga de código no seguro vía invocaciones de método remoto. Tiene como objetivo el control de las clases cargadas dinámicamente en una aplicación cliente.

Seguridad en RMI (Remote Method Invocation)

- Hacer que un objeto invoque a otro objeto remoto con independencia de la JVM o el servidor en el que se encuentra, como si fuera un objeto local. Transparencia respecto a la máquina.
- Arquitectura RMI

CLIENTE	OBJETO REMOTO	OSI
Cliente invocando método en el objeto remoto	Objeto remoto ofrece el servicio	Capa de aplicación
Stub	Skeleton	Capa de presentación
JRMP	JRMP	Capa de sesión
TCP	TCP	Capa de transporte
IP	IP	Capa de red
Interfaz de hardware ———	Interfaz de hardware ———	Capa de vínculos de datos

- El objeto *stub* encapsula al objeto remoto. Tiene una identificación del objeto remoto a utilizar y además su interfaz (métodos que son invocados, los parámetros y el tipo de retorno). El *stub* recibe la llamada al objeto remoto y a continuación:
  - Envía (marshalling) los parámetros codificados en un bloque de bytes al objeto remoto.
  - Son recibidos por el método remoto. Este método realiza su servicio (hace un cálculo, devuelve un resultado, etc.). El resultado codificado (o la excepción) lo recibe el stub.

Modelo de funcionamiento RMI