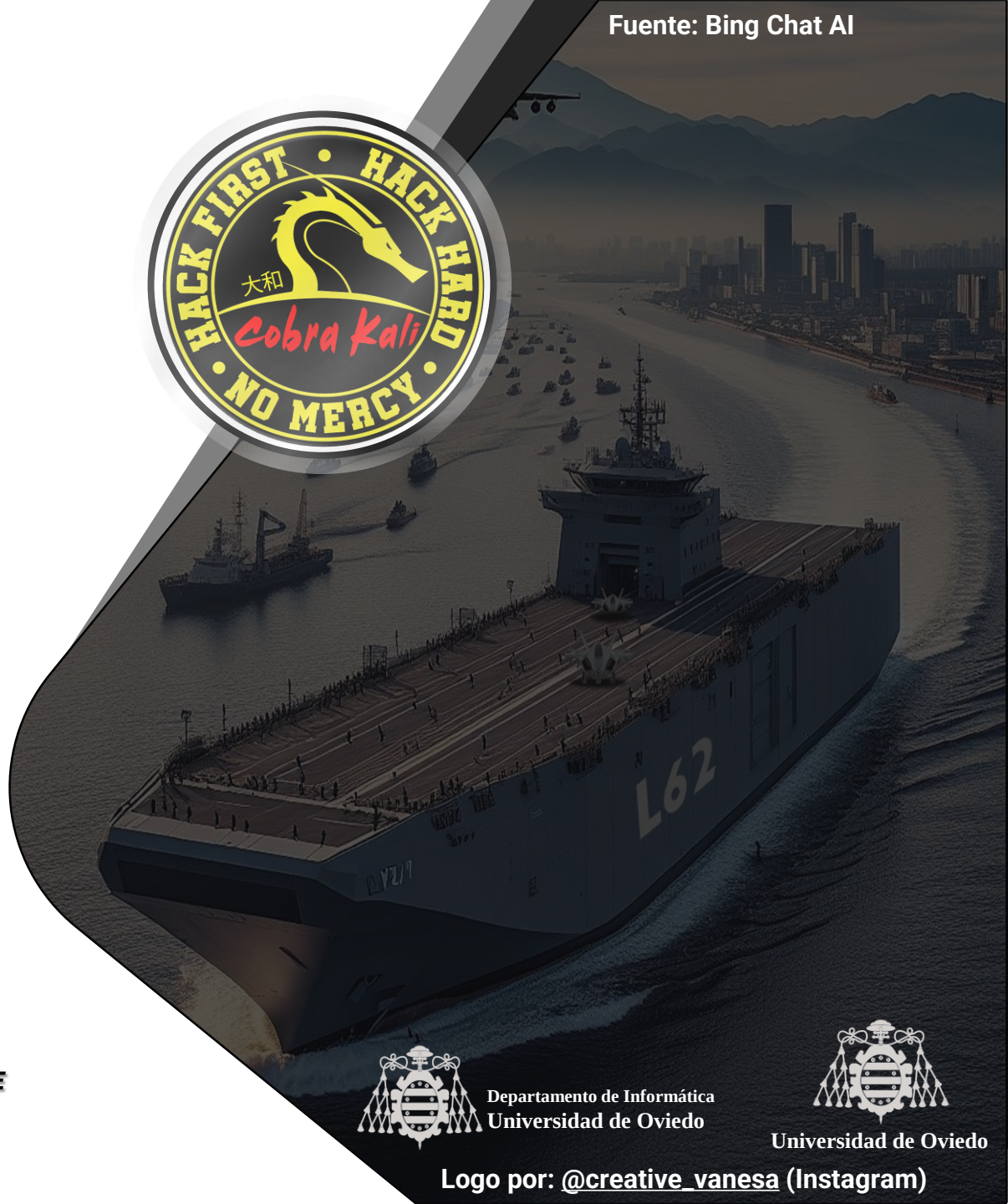
FORTIFICANDO APACHE 2 DE FORMA PROFESIONAL

Siguiendo las políticas CIS

José Manuel Redondo López Proyecto "*L-62 Princesa de Asturias*" v1.2



Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

Logo por: @creative_vanesa (Instagram)

ÍNDICE

- Conceptos básicos
  - Aspectos básicos de seguridad
- Proxies con Apache 2
- Implementando el CIS Benchmark para Apache 2
- Operaciones complementarias al CIS Benchmark
  - Contenidos del libro que acompaña la presentación
  - Módulos de Apache 2 destacables
    - Defensa contra ataques DoS y DDoS
    - Reescritura y ocultación de URLs
    - Bloqueo de orígenes maliciosos
    - Transformación automática de las webs
    - Subida de ficheros
    - Pruebas del servidor
- Consideraciones finales

¿QUÉ VAMOS A VER EN ESTE TEMA?

José Manuel
Redondo López

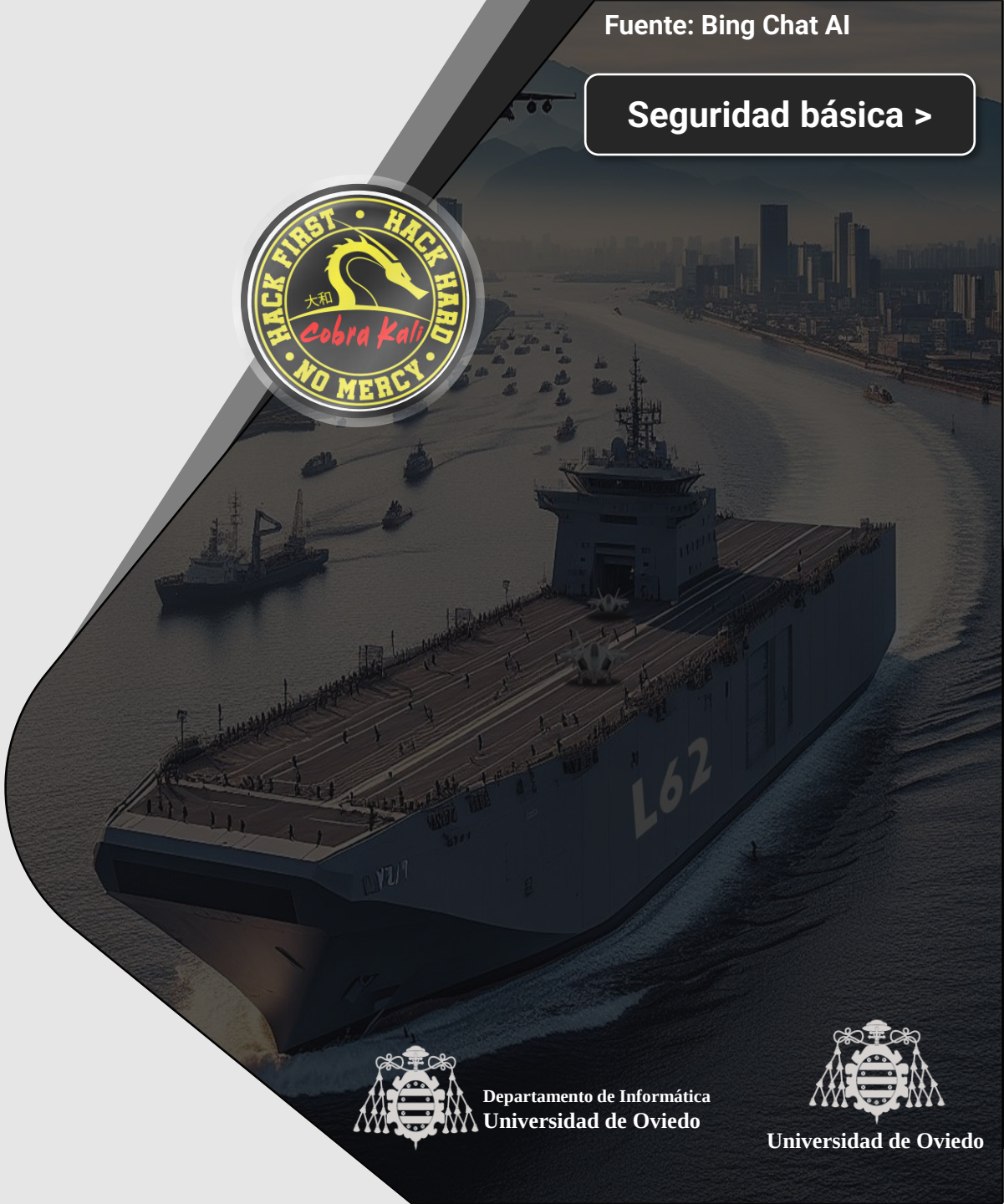
- En esta presentación veremos una serie de bloques
  - **Una política de securización de Apache 2 Web Server y por qué usar la misma**
  - **Los conceptos básicos necesarios para aplicar correctamente la política**
    - El libro adjunto en la asignatura ofrece una visión mucho más detallada de dichos conceptos
  - **La lista de acciones implementables que la política contempla**
    - El documento de la política adjunta incluye información en profundidad de todas ellas: por qué son necesarias, como comprobarlas, como implementarlas…
  - **Acciones de administración de Apache 2 adicionales**
    - Son acciones adicionales a los controles de seguridad de la política CIS
    - Los contenidos para implementar las mismas se encuentran bien en el libro o en la propia presentación
    - Están enfocadas a **dar servicios adicionales de forma correcta y segura**
    - No forman parte de la política de seguridad, **pero la complementan**
- Esta presentación es un complemento al libro entregado
  - Permite entender sus secciones de forma más sencilla

< Ir al Índice

Seguridad básica >

Conceptos básicos de Apache 2 Web Server

Antes de administrar...hay que conocer lo que se administra



Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

Apache 2 Web Server

José Manuel
Redondo López

- En esta sección sólo repasaremos los conceptos más importantes de cómo se ha construido el servidor y cómo se maneja el mismo
- Podéis encontrar dichos conceptos explicados en mucho más detalle en **el libro adjunto**
- Apache puede ser instalado en Linux y en Windows
  - Dentro de Linux, hay **dos ramas distintas**: las basadas en Debian y las basadas en Red Hat
  - El servidor es en esencia el mismo, pero hay una serie de cambios que debemos considerar a la hora de aplicar las técnicas en una distribución u otra
  - No obstante, no es difícil implementar las mismas acciones en las tres plataformas

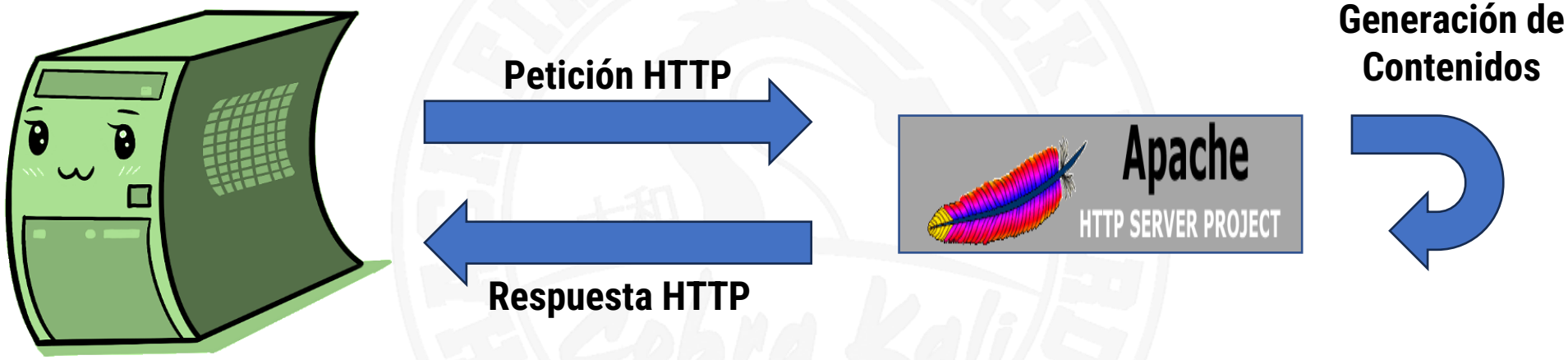
Glosario de términos

José Manuel
Redondo López

- **Módulo**: Extensión de la funcionalidad de **Apache 2** que puede activarse o no
- **Host virtual**: Unidad de configuración de **Apache 2** a la que se asigna una web
  - Le permite alojar múltiples webs al mismo tiempo
  - Por IP, puerto, nombre de dominio…
- **Contenedor de directivas**: Etiqueta que contiene directivas (opciones de configuración) de módulos y que determina a qué parte de los recursos se aplican
  - `<VirtualHost>`: Directivas para un host virtual concreto
  - `<Directory>`: Directivas sobre los contenidos de un directorio físico
  - `<Location>`: Directivas sobre una parte de una URI de la petición
    - No tiene por qué corresponder a una carpeta física
  - `<Files>`: Directivas para extensiones de archivos concretas
  - Ficheros `.htaccess`: Directivas solo para la carpeta que contiene este fichero, sobrescribiendo las generales

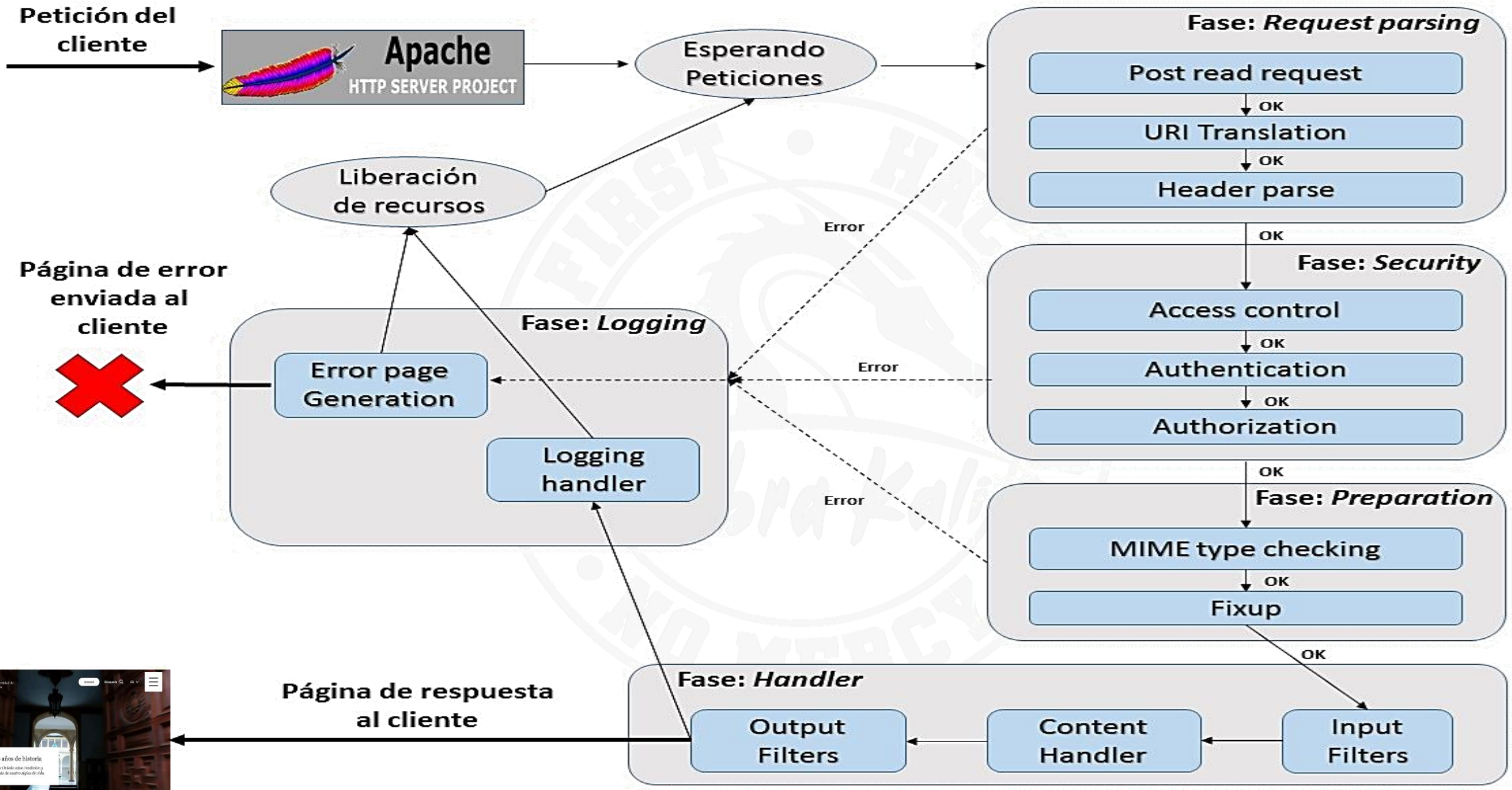
Procesamiento de peticiones en Apache (simple)

José Manuel
Redondo López



Procesamiento de peticiones de Apache (detallado)

José Manuel
Redondo López



Procesamiento de peticiones de Apache

José Manuel
Redondo López

- **La funcionalidad de Apache 2 está regida por los módulos que tenga activos**
  - Se pueden conseguir muchas cosas modificando la cadena de peticiones de Apache 2 con módulos ya existentes
  - O programados por nosotros (ampliar sus funcionalidades)
  - Módulos nuevos implica funcionalidades nuevas para el servidor web
- **Hay una enorme cantidad de módulos: https://httpd.apache.org/docs/2.4/es/mod/**
  - Internacionalización independiente del lenguaje
  - Soporte de lenguajes de servidor adicionales
  - *Server Side Includes*
  - Activar filtros de entrada y salida personalizados que manipulen los contenidos como queramos
  - Compresión automática de contenidos
  - …

Análisis de uso y forense

José Manuel
Redondo López

- **Apache 2 guarda un log de todas las peticiones que procesa**
  - Podemos usarlo para hacer análisis estadístico de uso
  - Y análisis forense de incidencias o ataques que el administrador haya sufrido
- **Normalmente para temas de seguridad se usan herramientas especializadas**
  - Para análisis de seguridad se usan herramientas de monitorización y correlación de logs
    - Ej.: Un SIEM ("**F-103 Blas de Lezo**")
- **La documentación aportada incluye como usar Awstats para hacer análisis estadístico de los accesos**
  - Volumen de peticiones, geolocalización, distribución horaria, etc.

Aspectos básicos de seguridad

Configurando inicialmente el servidor para darle un mínimo de seguridad

Restricción de acceso

José Manuel
Redondo López

- **Dos módulos para ello**
  - `mod_access_compat`: Mantenido por compatibilidad con configuraciones anteriores
    - **Obsoleto y debería dejar de usarse**
  - `mod_authz_host`: Nueva forma de especificar restricciones de acceso,
    - Hay que intentar usarla en lugar de la otra
- **Permiten restringir/permitir por**
  - IP del cliente (incluyendo subredes)
  - Listas de IPs
  - Nombres de dominio
  - IPV6
  - Presencia o no de variables de entorno
  - Sólo máquina local
  - Método HTTP
  - Expresiones evaluadas con variables de entorno (gran flexibilidad)

AUTENTICACIÓN

José Manuel
Redondo López

- **Consiste en requerir una identidad de usuario para ver ciertos contenidos**
- **Se puede conseguir con medios locales o externos, dependiendo de los módulos cargados para ello**
  - **Locales**: Ficheros de diferente tipo existentes en el disco duro de la máquina
    - Basic
    - Digest
    - Sobre `/etc/shadow`…
  - **Externos**
    - Datos de usuarios en LDAPs
    - Active Directory
    - Usando BBDD externas (MySQL…)

Sobre cifrado…

José Manuel
Redondo López

- **El cifrado de conexiones es imprescindible para mantener la seguridad**
  - **No se debería aceptan ninguna página en producción que use HTTP**
- **Algunas veces nos encontramos con páginas que usan HTTPs, pero mal**
  - Es decir, con métodos de cifrado obsoletos o mal configurados
  - Solo es cuestión de tiempo que les descifren el tráfico transmitido
- **Si se mantiene la compatibilidad con métodos de cifrado viejos por tener clientes muy viejos estamos comprometiendo todo el servidor**
  - Mejor intentar actualizar los clientes
- **Un atacante podría forzar a un cliente a usar un método de cifrado obsoleto**
  - Y por medio de ello hacerse con la información privada del cliente
  - Esto solo puede ocurrir si los seguimos soportando
- **Se recomienda usar guías de configuración actualizadas**
  - Ej.: https://ssl-config.mozilla.org/

MOD_SECURITY

José Manuel
Redondo López

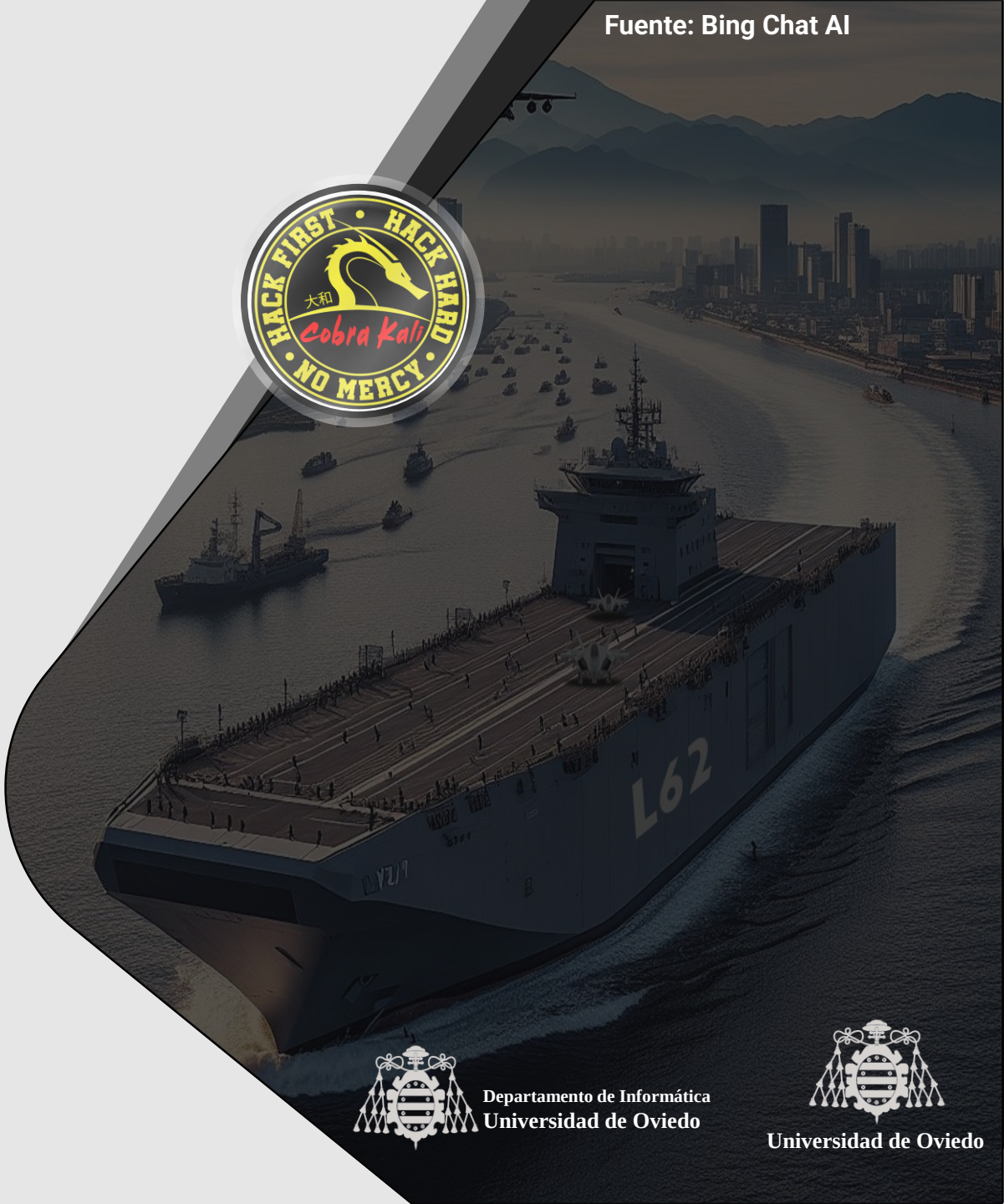
- **Es probablemente el WAF (Web Application Firewall) más conocido**
  - Entraremos más en detalle en ellos en la "**F-103 Blas de Lezo**"
- **Para funcionar necesita reglas, como las CRS (Core Security Rules)**
  - https://modsecurity.org/crs/
- **En las últimas versiones de Ubuntu las reglas se instalan junto con el propio WAF**
  - Por lo que es mucho más sencillo ponerlo en marcha
  - En otras distribuciones, hay que instalarlas independientemente
- **Una vez activo despliega un "escudo" para el servidor web que nos permite:**
  - **Protección HTTP**: Detecta violaciones del protocolo.
  - ***Blacklisting* de IPs en tiempo real**: Usando servidores de reputación de IPs de terceros

MOD_SECURITY

José Manuel
Redondo López

- **Detección de malware web**: con la API de Google Safe Browsing
- **Protección DoS**: Proporciona mecanismos de defensa contra ataques de saturación de peticiones o contra ataques basados en mandar peticiones con retardos preprogramados
- **Detección de ataques web comunes**
- **Detección de *bots*, *crawlers*, *scanners*** y otro tipo de herramientas maliciosas similares
- **Integración con antivirus** para examinar ficheros subidos por los clientes
- **Identificación de datos sensibles**: Como datos de tarjetas de crédito, bloqueando su posible robo
- **Protección contra troyanos**
- **Identifica problemas de configuración** de las aplicaciones
- **Detecta y oculta mensajes de error** enviados por el servidor para que no den demasiada información
- **Previene de que se puedan servir ficheros con código** fuente de la aplicación, por ejemplo por la presencia de *backups* o versiones viejas de una determinada página
- **Enmascaramiento** del servidor
- …

< Ir al Índice



Proxies con Apache 2

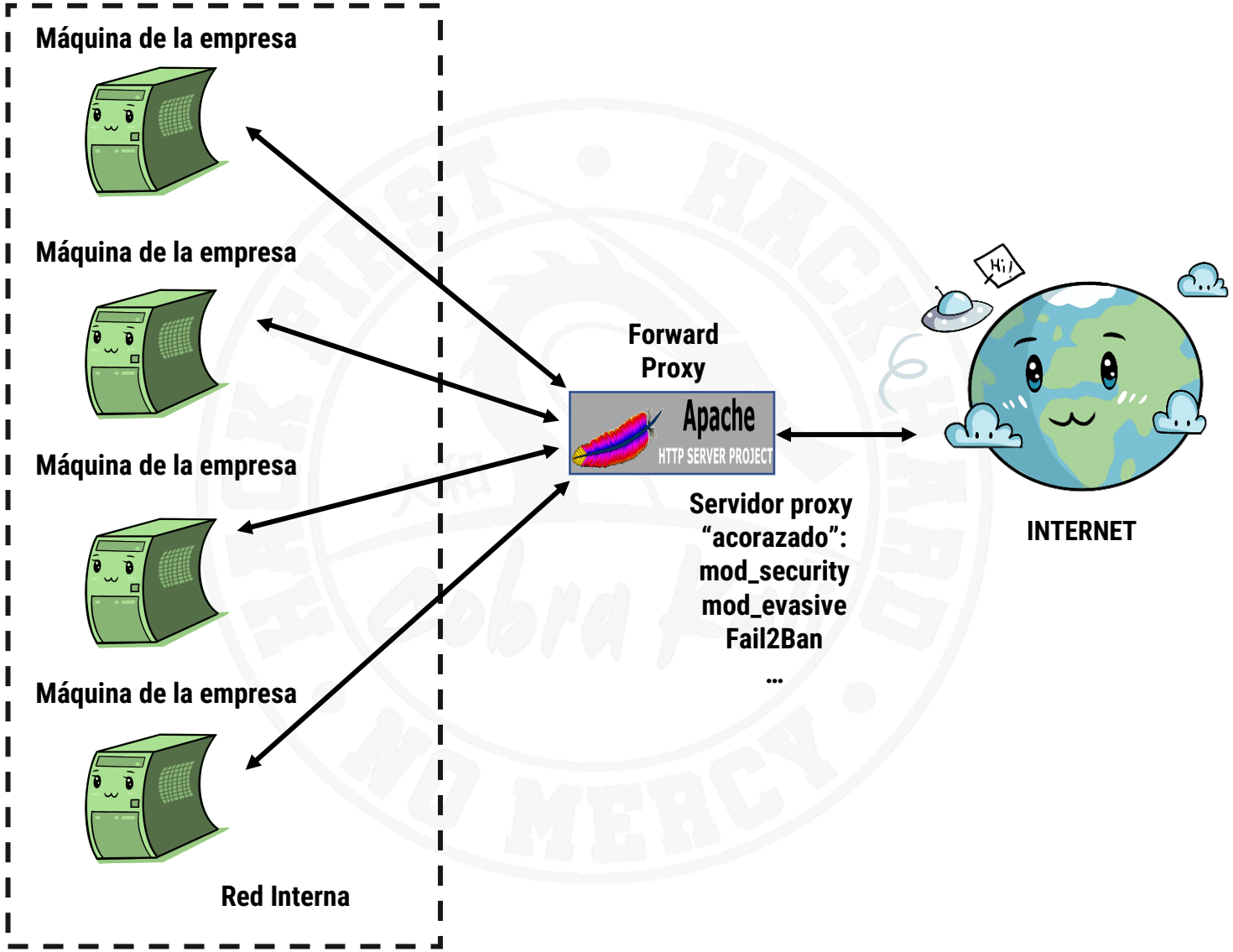
Poniendo Apache 2 como “barrera”

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

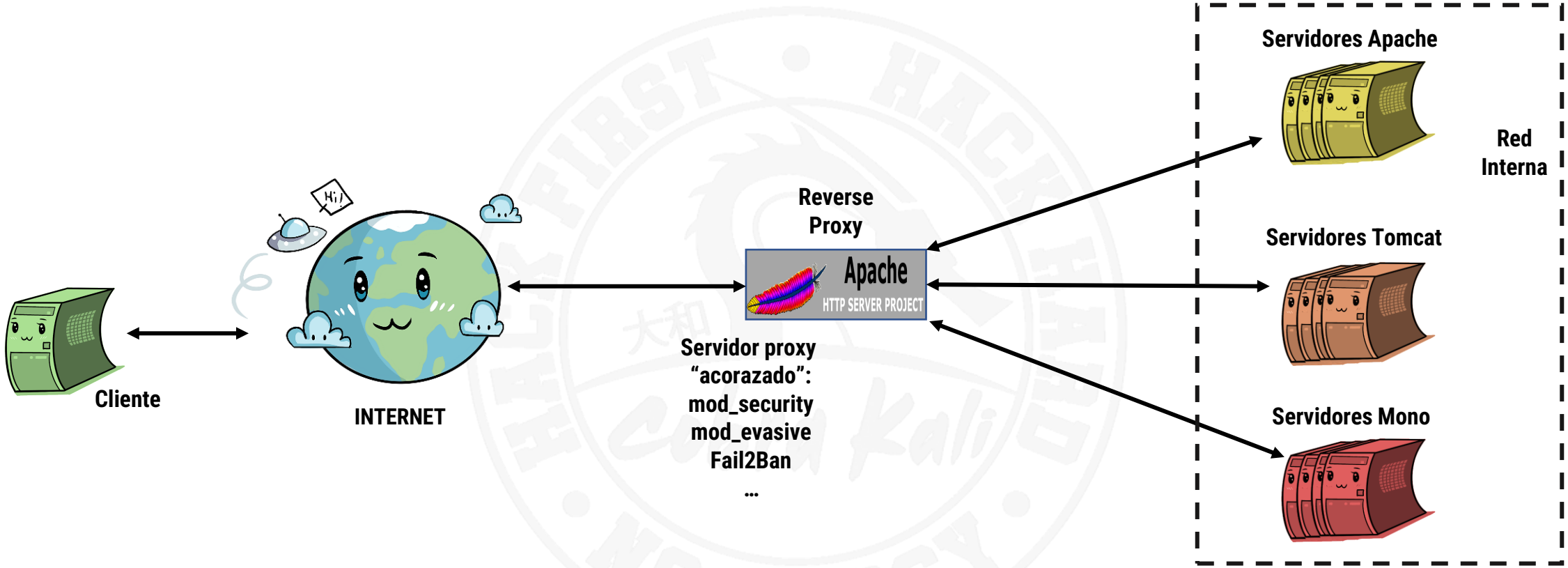
FORWARD PROXY

José Manuel
Redondo López



Reverse Proxy

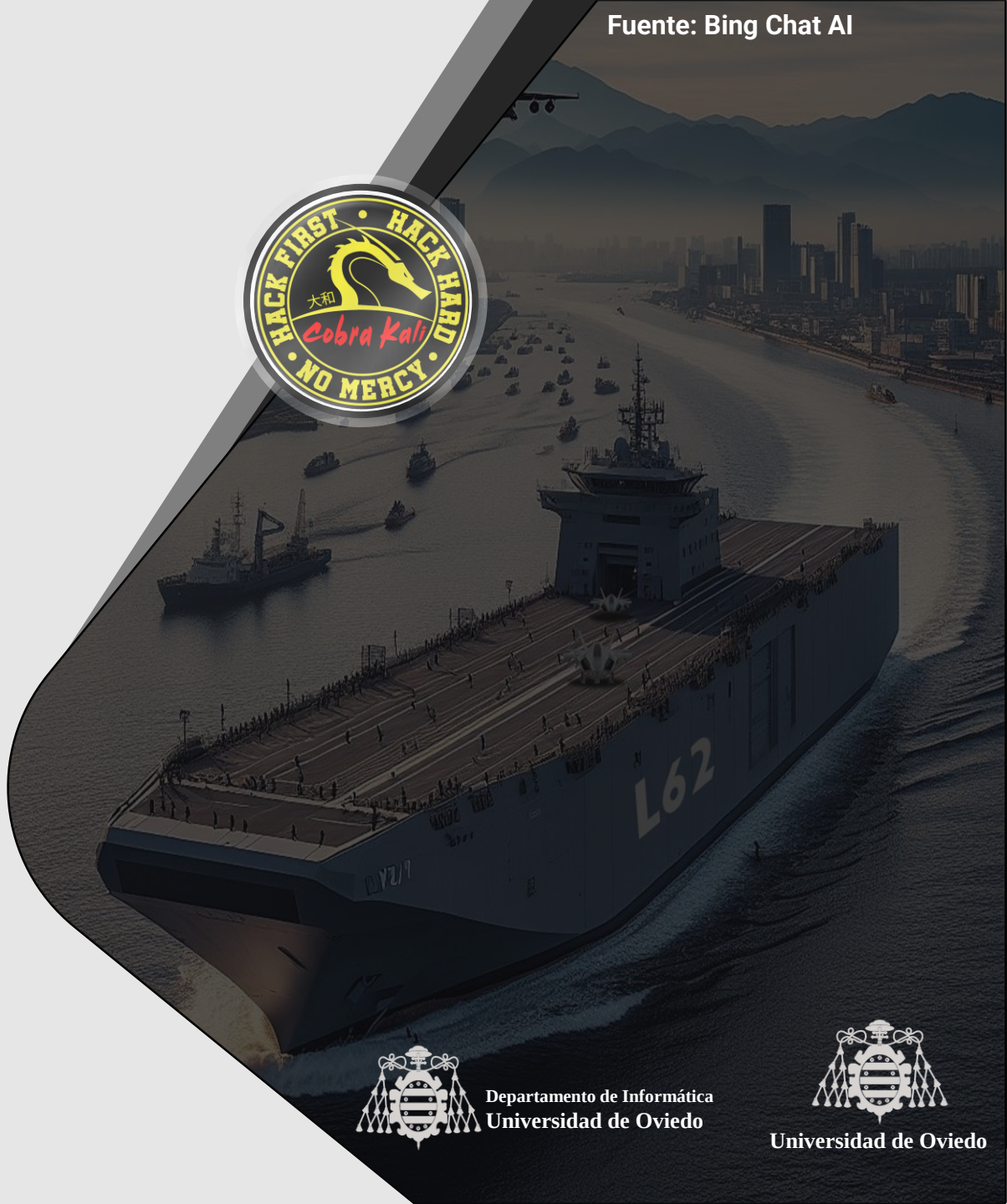
José Manuel
Redondo López



< Ir al Índice

Implementando el CIS Benchmark de Apache 2

Catálogo de acciones a realizar



Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

COSAS A SABER ANTES DE IMPLEMENTAR LA POLÍTICA

José Manuel
Redondo López

- **En Debian el ejecutable del servidor se denomina `apache2`, en Red Hat `httpd`**
- **La organización de las carpetas del servidor varía entre ambas distribuciones**
  - Para seguirla más fácilmente tenemos que **establecer el valor de algunas variables de entorno**
  - Para establecer variables de entorno, podemos seguir este procedimiento
    - https://www.cyberciti.biz/faq/set-environment-variable-linux/
  - En RHEL 8 (y distribuciones Red Hat), `$APACHE_PREFIX` debe valer `/etc/httpd`
    - En sistemas Debian su valor debe ser `/etc/apache2`
  - En RHEL 8, `$DOCROOT` es `/var/www/html` y ya no `/usr/local/apache2/htdocs` como antes
    - Este directorio es el mismo que en sistemas Debian
- **Un cambio en los ficheros de configuración de Apache 2 requiere un reinicio del servidor en sistemas Debian y (por precaución) también en Red Hat**
  - Aunque en estos últimos sistemas los cambios suelen aplicarse automáticamente
  - En Debian se puede hacer el reinicio con `service apache2 restart`
  - En Red Hat se puede seguir estas indicaciones: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/s1-apache-startstop

COSAS A SABER ANTES DE IMPLEMENTAR LA POLÍTICA

José Manuel
Redondo López

- **Directorios de `/etc/httpd`**
  - `conf`: Aquí se encuentra la configuración principal de Apache 2, en el fichero `httpd.conf`
  - `conf.d`: Contiene unos ficheros de configuración adicionales que se cargan desde el principal, relativos a ciertos módulos especiales (`userdir`, `ssl`…)
  - `conf.modules.d`: Contiene ficheros `.conf` que cargan los módulos correspondientes a ciertas funcionalidades
    - Si se cambia la extensión de los ficheros, **no se cargarán sus módulos**
    - Si se comenta alguna línea, no se cargará su módulo correspondiente
    - No cargar algún módulo puede causar algún problema con los `.conf` del directorio anterior
      - Normalmente no se reconocerán ciertas directivas
      - Cada cambio debe ser revisado con cuidado reiniciando el servidor

Cosas a saber antes de implementar la política

José Manuel
Redondo López

- `logs`: Logs de Apache 2 varios (acceso, errores…), incluyendo peticiones bajo HTTPs (`ssl_*.log`) y otros módulos que requieran un log propio (como `mod_security`)
- `modsecurity.d`: Reglas en uso de `mod_security`, si el módulo se ha instalado
- `modules`: Binarios de los módulos disponibles en Apache 2 localmente
- `run`: Ficheros necesarios durante el funcionamiento de Apache 2
- `state`: Ficheros necesarios durante el funcionamiento de Apache 2

- **Aplicar la política de seguridad CIS de Apache 2 consiste en rellenar las tablas siguientes**
  - Que contemplan una serie de acciones divididas en categorías
  - Están sacada directamente del documento de la política suministrado

Política CIS Benchmark v2.0.1 para Apache 2.4

José Manuel
Redondo López

		YES	NO
1	**Planning and Installation**		
1.1	Ensure the Pre-Installation Planning Checklist Has Been Implemented (Not Scored)		
1.2	Ensure the Server Is Not a Multi-Use System (Not Scored)		
1.3	Ensure Apache Is Installed From the Appropriate Binaries (Not Scored)		
2	**Minimize Apache Modules**		
2.1	Ensure Only Necessary Authentication and Authorization Modules Are Enabled (Not Scored)		
2.2	Ensure the Log Config Module Is Enabled (Scored)		
2.3	Ensure the WebDAV Modules Are Disabled (Scored)		
2.4	Ensure the Status Module Is Disabled (Scored)		
2.5	Ensure the Autoindex Module Is Disabled (Scored)		
2.6	Ensure the Proxy Modules Are Disabled (Scored)		
2.7	Ensure the User Directories Module Is Disabled (Scored)		
2.8	Ensure the Info Module Is Disabled (Scored)		
2.9	Ensure the Basic and Digest Authentication Modules are Disabled (Scored)		

Política CIS Benchmark v2.0.1 para Apache 2.4

José Manuel
Redondo López

		YES	NO
3	**Principles, Permissions, and Ownership**		
3.1	Ensure the Apache Web Server Runs As a Non-Root User (Scored)		
3.2	Ensure the Apache User Account Has an Invalid Shell (Scored)		
3.3	Ensure the Apache User Account Is Locked (Scored)		
3.4	Ensure Apache Directories and Files Are Owned By Root (Scored)		
3.5	Ensure the Group Is Set Correctly on Apache Directories and Files (Scored)		
3.6	Ensure Other Write Access on Apache Directories and Files Is Restricted (Scored)		
3.7	Ensure the Core Dump Directory Is Secured (Scored)		
3.8	Ensure the Lock File Is Secured (Scored)		
3.9	Ensure the Pid File Is Secured (Scored)		
3.10	Ensure the ScoreBoard File Is Secured (Scored)		
3.11	Ensure Group Write Access for the Apache Directories and Files Is Properly Restricted (Scored)		
3.12	Ensure Group Write Access for the Document Root Directories and Files Is Properly Restricted (Scored)		
3.13	Ensure Access to Special Purpose Application Writable Directories is Properly Restricted (Not Scored)		
4	**Apache Access Control**		
4.1	Ensure Access to OS Root Directory Is Denied By Default (Scored)		
4.2	Ensure Appropriate Access to Web Content Is Allowed (Not Scored)		
4.3	Ensure OverRide Is Disabled for the OS Root Directory (Scored)		
4.4	Ensure OverRide Is Disabled for All Directories (Scored)		

POLÍTICA CIS BENCHMARK V2.0.1 PARA APACHE 2.4

José Manuel
Redondo López

		YES	NO
5	**Minimize Features, Content and Options**		
5.1	Ensure Options for the OS Root Directory Are Restricted (Scored)		
5.2	Ensure Options for the Web Root Directory Are Restricted (Scored)		
5.3	Ensure Options for Other Directories Are Minimized (Scored)		
5.4	Ensure Default HTML Content Is Removed (Scored)		
5.5	Ensure the Default CGI Content printenv Script Is Removed (Scored)		
5.6	Ensure the Default CGI Content test-cgi Script Is Removed (Scored)		
5.7	Ensure HTTP Request Methods Are Restricted (Scored)		
5.8	Ensure the HTTP TRACE Method Is Disabled (Scored)		
5.9	Ensure Old HTTP Protocol Versions Are Disallowed (Scored)		
5.10	Ensure Access to .ht* Files Is Restricted (Scored)		
5.11	Ensure Access to .git Files Is Restricted (Scored)		
5.12	Ensure Access to .svn Files Is Restricted (Scored)		
5.13	Ensure Access to Inappropriate File Extensions Is Restricted (Scored)		
5.14	Ensure IP Address Based Requests Are Disallowed (Scored)		
5.15	Ensure the IP Addresses for Listening for Requests Are Specified (Scored)		
5.16	Ensure Browser Framing Is Restricted (Scored)		
5.17	Ensure HTTP Header Referrer-Policy is set appropriately		
5.18	Ensure HTTP Header Permissions-Policy is set appropriately		

Política CIS Benchmark v2.0.1 para Apache 2.4

José Manuel
Redondo López

		YES	NO
6	**Operations - Logging, Monitoring and Maintenance**		
6.1	Ensure the Error Log Filename and Severity Level Are Configured Correctly (Scored)		
6.2	Ensure a Syslog Facility Is Configured for Error Logging (Scored)		
6.3	Ensure the Server Access Log Is Configured Correctly (Scored)		
6.4	Ensure Log Storage and Rotation Is Configured Correctly (Scored)		
6.5	Ensure Applicable Patches Are Applied (Scored)		
6.6	Ensure ModSecurity Is Installed and Enabled (Scored)		
6.7	Ensure the OWASP ModSecurity Core Rule Set Is Installed and Enabled (Scored)		

Política CIS Benchmark v2.0.1 para Apache 2.4

José Manuel
Redondo López

		YES	NO
7	**SSL/TLS Configuration**		
7.1	Ensure mod_ssl and/or mod_nss Is Installed (Scored)		
7.2	Ensure a Valid Trusted Certificate Is Installed (Scored)		
7.3	Ensure the Server's Private Key Is Protected (Scored)		
7.4	Ensure the TLSv1.0 and TLSv1.1 Protocols are Disabled		
7.5	Ensure Weak SSL/TLS Ciphers Are Disabled (Scored)		
7.6	Ensure Insecure SSL Renegotiation Is Not Enabled (Scored)		
7.7	Ensure SSL Compression is not Enabled (Scored)		
7.8	Ensure Medium Strength SSL/TLS Ciphers Are Disabled (Scored)		
7.9	Ensure All Web Content is Accessed via HTTPS (Scored)		
7.10	Ensure OCSP Stapling Is Enabled (Scored)		
7.11	Ensure HTTP Strict Transport Security Is Enabled (Scored)		
7.12	Ensure Only Cipher Suites That Provide Forward Secrecy Are Enabled (Scored)		
8	**Information Leakage**		
8.1	Ensure ServerTokens is Set to 'Prod' or 'ProductOnly' (Scored)		
8.2	Ensure ServerSignature Is Not Enabled (Scored)		
8.3	Ensure All Default Apache Content Is Removed (Scored)		
8.4	Ensure ETag Response Header Fields Do Not Include Inodes (Scored)		

Política CIS Benchmark v2.0.1 para Apache 2.4

José Manuel
Redondo López

		YES	NO
9	**Denial of Service Mitigations**		
9.1	Ensure the TimeOut Is Set to 10 or Less (Scored)		
9.2	Ensure KeepAlive Is Enabled (Scored)		
9.3	Ensure MaxKeepAliveRequests is Set to a Value of 100 or Greater (Scored)		
9.4	Ensure KeepAliveTimeout is Set to a Value of 15 or Less (Scored)		
9.5	Ensure the Timeout Limits for Request Headers is Set to 40 or Less (Scored)		
9.6	Ensure Timeout Limits for the Request Body is Set to 20 or Less (Scored)		
10	**Request Limits**		
10.1	Ensure the LimitRequestLine directive is Set to 512 or less (Scored)		
10.2	Ensure the LimitRequestFields Directive is Set to 100 or Less (Scored)		
10.3	Ensure the LimitRequestFieldsize Directive is Set to 1024 or Less (Scored)		
10.4	Ensure the LimitRequestBody Directive is Set to 102400 or Less (Scored)		
11	**Enable SELinux to Restrict Apache Processes**		
11.1	Ensure SELinux Is Enabled in Enforcing Mode (Scored)		
11.2	Ensure Apache Processes Run in the httpd_t Confined Context (Scored)		
11.3	Ensure the httpd_t Type is Not in Permissive Mode (Scored)		
11.4	Ensure Only the Necessary SELinux Booleans are Enabled (Not Scored)		
12	**Enable AppArmor to Restrict Apache Processes**		
12.1	Ensure the AppArmor Framework Is Enabled (Scored)		
12.2	Ensure the Apache AppArmor Profile Is Configured Properly (Not Scored)		
12.3	Ensure Apache AppArmor Profile is in Enforce Mode (Scored)		

< Ir al Índice

Fuente: Bing Chat AI

Libro >

Módulos >

Operaciones complementarias al CIS Benchmark para Apache 2

Yendo a un nivel de seguridad mayor

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

Contenidos del libro que acompaña a esta presentación

Conceptos básicos para entender su contenido

¿Qué hay en el libro de Apache 2?

José Manuel
Redondo López

- **Una vez implementada la política CIS Benchmark en su totalidad o parcialmente gracias a su guía, podemos empezar a construir más capas de seguridad**
  - Hay una serie de acciones que permiten incrementar aún más la seguridad del servidor web
- **¿Por qué no se incluyen en la política?**
  - Son muy dependientes de las aplicaciones ejecutadas, no reglas generales
  - Se suelen aplicar en servidores ajenos que dan acceso al que protege la política (proxies)
  - Tienen que ver más con programación que con la securización de la plataforma
- **Pero debemos verlos igualmente para lograr sistemas aún más seguros**

¿Qué hay en el libro de Apache 2?

José Manuel
Redondo López

- **En esta asignatura se suministra un libro de administración de Apache 2 que complementa a esta presentación**
- **Muestra fundamentalmente dos cosas**
  - Cómo implementar muchas de las acciones en sistemas Debian
    - La política CIS está enfocada a sistemas Red Hat
  - Cómo implementar las acciones complementarias a la política también bajo Debian
    - Aunque su implementación es muy similar en Red Hat
    - Las posibles diferencias se detallarán en esta presentación
- **En concreto todo lo que cuenta se divide en tres grandes bloques**
  - Muchas de ellas están vinculadas a la asignatura "**F-103 Blas de Lezo**"
  - Para la "**L-62 Princesa de Asturias**" no tienen aplicación
    - Pero no está de más que sepáis lo que se puede hacer ☺

¿Qué hay en el libro de Apache 2?

José Manuel
Redondo López

- **Antes de que una petición llegue al servidor…**
  - **Ocultación del servidor y de las tecnologías que usa**
    - Hacer el servidor “invisible” a clientes no autorizados
    - Para los clientes que sea visible, que les aparezca de tipo y versión desconocida
    - Que el servidor solo sirva a las máquinas autorizadas e ignora al resto, incluso por región
    - Cifrado de comunicaciones
  - **Filtrado de peticiones**
    - Software “escudo de amplio espectro” contra las peticiones maliciosas (WAF)
    - Sistemas anti-denegación de servicio
    - Mecanismos de reacción proactiva ante detección de ataques

¿Qué hay en el libro de Apache 2?

José Manuel
Redondo López

- **Cuando una petición llegue al servidor…**
  - Cómo servir sólo tipos de recursos autorizados
  - Minimización de tecnologías y módulos admitidos
  - Configuración de permisos para impedir acceder a ficheros no autorizados
  - Monitorización de rendimiento y accesos, análisis de peticiones
- **Cuando una petición salga del servidor…**
  - Optimización de las páginas generadas independiente del lenguaje de servidor que usen
  - Bloqueo de robo de ancho de banda por parte de terceros
  - Análisis de posibles fugas de información
- **Transversalmente a las tres fases**
  - Técnicas para optimizar el rendimiento y la administración del servidor

Fuente: Bing Chat AI

DDoS >

Reescritura URLs >

Bloqueo orígenes >

Transformación >

Subida ficheros >

Pruebas >

Módulos de Apache 2 destacables

Módulos del servidor sobre los que tenemos que prestar especial atención

Lenguajes del servidor

José Manuel
Redondo López

- Lo mejor es instalar **1 sólo lenguaje en cada servidor web**
  - Es lo mejor de cara a la seguridad
- Si no es posible, mantener host virtuales separados para cada lenguaje
- Y, en caso de que sea necesario más de uno, siempre hay que **intentar instalar los menos posibles**

Defensa contra ataques DoS y DDoS

MOD_EVASIVE

José Manuel
Redondo López

- **Módulo anti DoS/DDoS para Apache 2**
- **Potencial complemento anti DoS/DDoS para `mod_security`**
- **Retrasa, no evita, este tipo de ataques**
  - Realmente, si el atacante pone el empeño suficiente, es muy difícil evitarlos totalmente...
- **Actúa cuando**
  - Se solicite la misma página más de un nº determinado de veces por segundo
  - Se hagan más de un nº concreto de peticiones concurrentes por segundo
  - Otros casos operados por una botnet
- **Instalación en sistemas Red Hat**
  - https://phoenixnap.com/kb/apache-mod-evasive

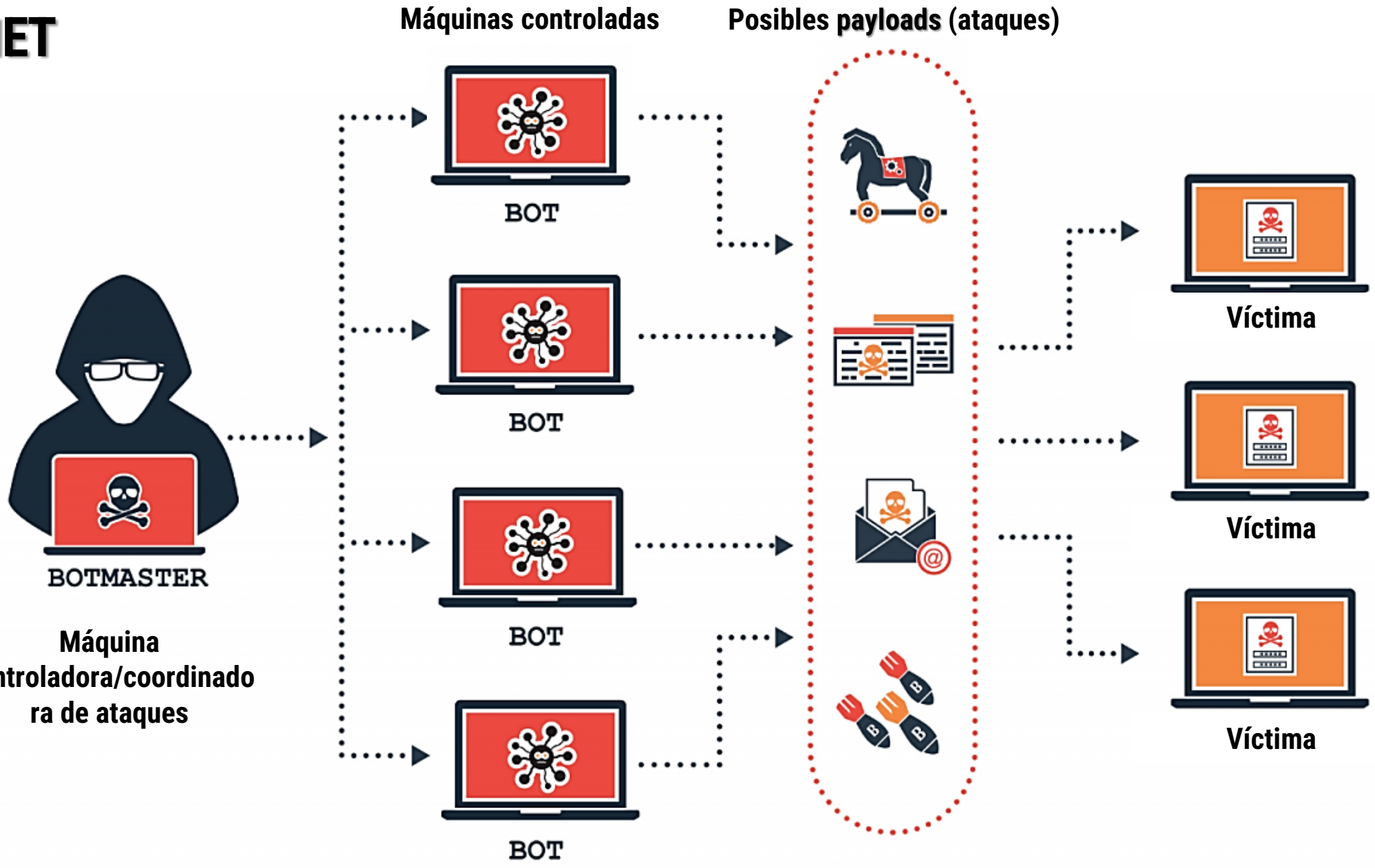
Fail2Ban

José Manuel
Redondo López

- **Detecta y bloquea intentos de intrusión**
  - Analizando el log de peticiones/errores de **Apache 2** o de otros servicios
- **Si detecta algún tráfico sospechoso bloquea su origen**
  - Es por tanto una herramienta de acción/reacción
- **Para varios servicios (especialmente SSH), no solo Apache 2**
- **Configurable y programable, con bastantes jails disponibles recién instalada**
  - Una **jail** es un esquema de protección de un servicio ante un ataque activable
  - Cubren varios escenarios tradicionalmente ejecutables por una **botnet**
  - Instalación en sistemas **Red Hat**
    - https://www.tutorialspoint.com/articles/how-to-secure-the-sshd-using-fail2ban-on-rhel-7-xcentos-7-x
  - Uso en sistemas **Red Hat**
    - https://www.tecmint.com/use-fail2ban-to-secure-linux-server/

BOTNET

José Manuel
Redondo López



Esquema de una botnet clásica. Fuente: https://www.redseguridad.com/actualidad/cibercrimen/que-es-una-botnet-y-como-puedo-prevenirla-y-evitarla_20210630.html

Reescritura y ocultación de URLs

MOD_REWRITE

José Manuel
Redondo López

- **Módulo de reescritura de URLs**
- **Es una “navaja suiza” (¡vale para casi todo! ☺)**
- **Ejemplos de uso**
  - Reescritura de URLs para dejarlas más “bonitas”/usables/entendibles
  - Eliminar partes de la dirección
  - Evitar el *hotlinking*
  - Acceder a páginas de error personalizadas
  - Proteger el acceso a directorios y tipos de fichero concretos
  - Restringir los métodos HTTP admitidos
  - Prohibir el acceso mediante proxies

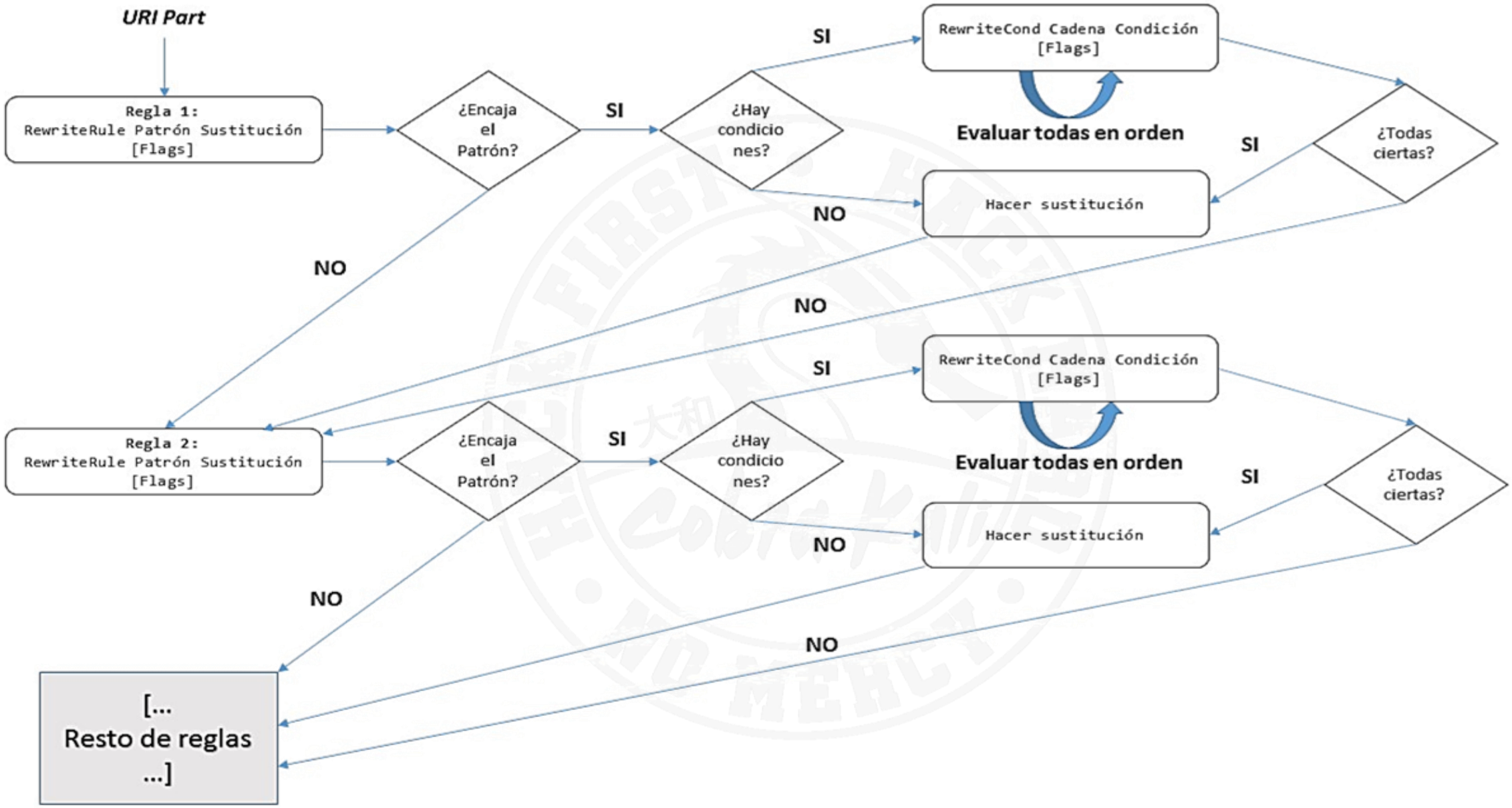
MOD_REWRITE

José Manuel
Redondo López

- Prohibir versiones concretas del protocolo HTTP
- Limitar los caracteres presentes en las URI
- Prohibir peticiones erróneas/corruptas/mal formadas
- Manipulación de los parámetros de la petición (*querystring*)
- Bloquear a clientes que se conectan desde una lista de IPs concretas (**lista de bloqueo**)
  - Y lo contrario (lista de permitidos) también
- Bloquear a spider bots y otros agentes de usuarios maliciosos:
  - Se pueden encontrar filtros ya hechos con `mod_rewrite` por terceros adaptados a nuestras necesidades

MOD_REWRITE

José Manuel
Redondo López



Bloqueo de orígenes maliciosos

Red ToR

José Manuel
Redondo López

- El software llamado ToR (The Onion Routing project, https://www.torproject.org/) es una forma popular de incrementar el anonimato en Internet
- Es la implementación más conocida de una técnica llamada Onion Routing
  - http://www.onion-router.net/Publications/CACM-1999.pdf
  - Gracias a esta técnica se hace mucho más difícil saber el origen real de un cliente
- Si bien el anonimato en Internet es un derecho, la red ToR suele ser usada para fines maliciosos
  - Aprovechándose del anonimato que proporciona, atacantes intentan realizar ataques de diversa índole a servidores
  - Siendo estos muy difíciles de parar y, sobre todo, de localizar al culpable

Red ToR

José Manuel
Redondo López

- **El problema es que la propia estructura de la red *ToR* hace que identificar conexiones realizadas a través de la misma sea muy complejo**
  - Las peticiones “saltan” cifradas por un conjunto aleatorio de nodos de la red ToR
  - Al final, llegan a su destino por unos nodos ToR especiales (“nodos de salida”)
  - https://www.xataka.com/basic/red-tor-que-como-funciona-como-se-usa
  - https://www.csoonline.com/article/565798/what-is-the-tor-browser-how-it-works-and-how-it-can-help-you-protect-your-identity-online.html
- **Se puede intentar cortar el mayor nº posible de conexiones que provengan de dicha red bloqueando precisamente esos nodos de salida**
  - Si lo hacemos estaremos perdiendo posibles visitas legítimas sin fines maliciosos a nuestra web
- **Y es gracias a `mod_rewrite` y su habilidad para bloquear listas de IPs**
- **Pero ¿De dónde saco una lista de IPs de nodos de salida de la red ToR?**
  - https://www.dan.me.uk/torlist/ (se actualiza cada media hora)

GEOLOCALIZACIÓN

José Manuel
Redondo López

- **Módulo de Apache 2 para geolocalizar a los clientes que nos hacen peticiones**
  - País
  - Región
  - Ciudad
  - ISP
  - Organización
  - Velocidad de conexión
- **Se puede usar esta información para permitir/bloquear/redirigir peticiones de muchas formas**
  - Prohibir países
  - Admitir sólo ciertos países
  - …

Transformación automática de las webs

PAGESPEED

José Manuel
Redondo López

- **Módulo de Apache 2 creado por Google**
- **Contiene una gran cantidad de funcionalidades que ayudan a optimizar el servicio de las páginas web de diversas formas**
  - Y un gran nº de filtros asociados a diversos tipos de ficheros (.css, Javascript, HTML e imágenes, entre otros)
  - Permiten la optimización de estos tipos de ficheros de acuerdo a diversos criterios
- **Mejora la capacidad de Apache 2 para servir webs optimizando sus contenidos**
  - Además, sin que haya que hacer cambios en el código de las páginas web (es transparente)
  - Instalación en Red Hat: https://lintut.com/how-to-install-mod_pagespeed-in-rhel-centos-fedora-linux/
  - Configuración en Red Hat: https://haloseeker.com/install-google-mod_pagespeed-on-rhel-centos-and-fedora/

PAGESPEED

José Manuel
Redondo López

- **"Minificar" cualquier contenido de la página (HTML, CSS, Javascript): Es bueno desde el punto de vista del rendimiento y de la seguridad**
  - Un código más ofuscado es más difícil de interpretar
  - Por tanto, **dificultará saber detalles técnicos de nuestra página** que pueden abrir la puerta a ataques de seguridad
  - Un fichero "minificado" ocupa menos y por tanto supone una ganancia de rendimiento, al transmitirse menos información vía red
    - Optimizamos así el ancho de banda a coste cero
  - El módulo lo hace "en caliente", nosotros no tenemos que modificar nuestras páginas para ello

PAGESPEED

José Manuel
Redondo López

- **Quitar los comentarios HTML es imprescindible en una web en producción**
  - Un comentario puede dar información que comprometa la seguridad de nuestra plataforma
  - Una página sin comentarios ocupa menos, por lo que supone también una mejora de rendimiento
  - Este *mod* hace este trabajo por nosotros, si activamos el filtro correspondiente
  - Así es más fácil el paso de desarrollo a producción
  - Aunque aún tendremos que hacerlo en Javascript o CSS embebidos
- **Quitar los metadatos de los recursos también es conveniente por temas de seguridad**
  - En especial de documentos e imágenes
  - Los metadatos de una imagen alojada en nuestro servidor pueden ser muy “golosos” para un atacante, ya que podría permitir averiguar datos sobre nuestra
  - ¿Y cómo sé si alguna de mis imágenes tiene metadatos?
    - https://www.metadata2go.com/

Subida de ficheros

WebDAV (*Web-based Distributed Authoring and Versioning*)

José Manuel
Redondo López

- **Protocolo usado para publicación y gestión de contenido en servidores remotos, y también servidores web**
  - Extensión de HTTP 1.1 (puerto 80)
- **Permite a los usuarios editar contenidos de forma colaborativa y administrar archivos en servidores web remotos**
  - Podemos crear, modificar o eliminar documentos en un servidor remoto o “carpeta compartida web”
- **El uso principal es permitir editar los contenidos de una web en remoto**
  - Sin necesidad de acceder directamente al servidor
- **A efectos de uso, WebDAV funciona como un almacenamiento remoto de archivos accesible vía web desde cualquier parte**
- **Debe evitar usarse salvo que sea estrictamente necesario por las vulnerabilidades que puede llegar a causar**
  - **En caso de usarse, solo desde máquinas expresamente autorizadas**

Mecanismos de protección en directorios de subida de archivos

José Manuel
Redondo López

- Mediante directivas, definir qué archivos son accesibles y qué archivos no
- Si se define un fichero `.htaccess` con directivas de protección, nunca ponerlo en el mismo directorio donde se suben los archivos (ponerlo en el directorio padre)
- Si es posible, hacer que el directorio donde se suban los archivos esté aislado del resto de contenidos web
- No permitir que se sobrescriban archivos usando permisos
- Crear una lista de tipos MIME permitidos y derivar de ella posibles extensiones admitidas, como comprobación extra a todas las demás
- Dar un nombre aleatorio a los ficheros subidos, y asignarle manualmente la extensión correspondiente a su tipo MIME declarado
- Incluir validación en el cliente de los ficheros a subir y **TAMBIÉN EN EL SERVIDOR**

Pruebas del servidor

Chequeo automático de seguridad para pagínas PHP

José Manuel
Redondo López

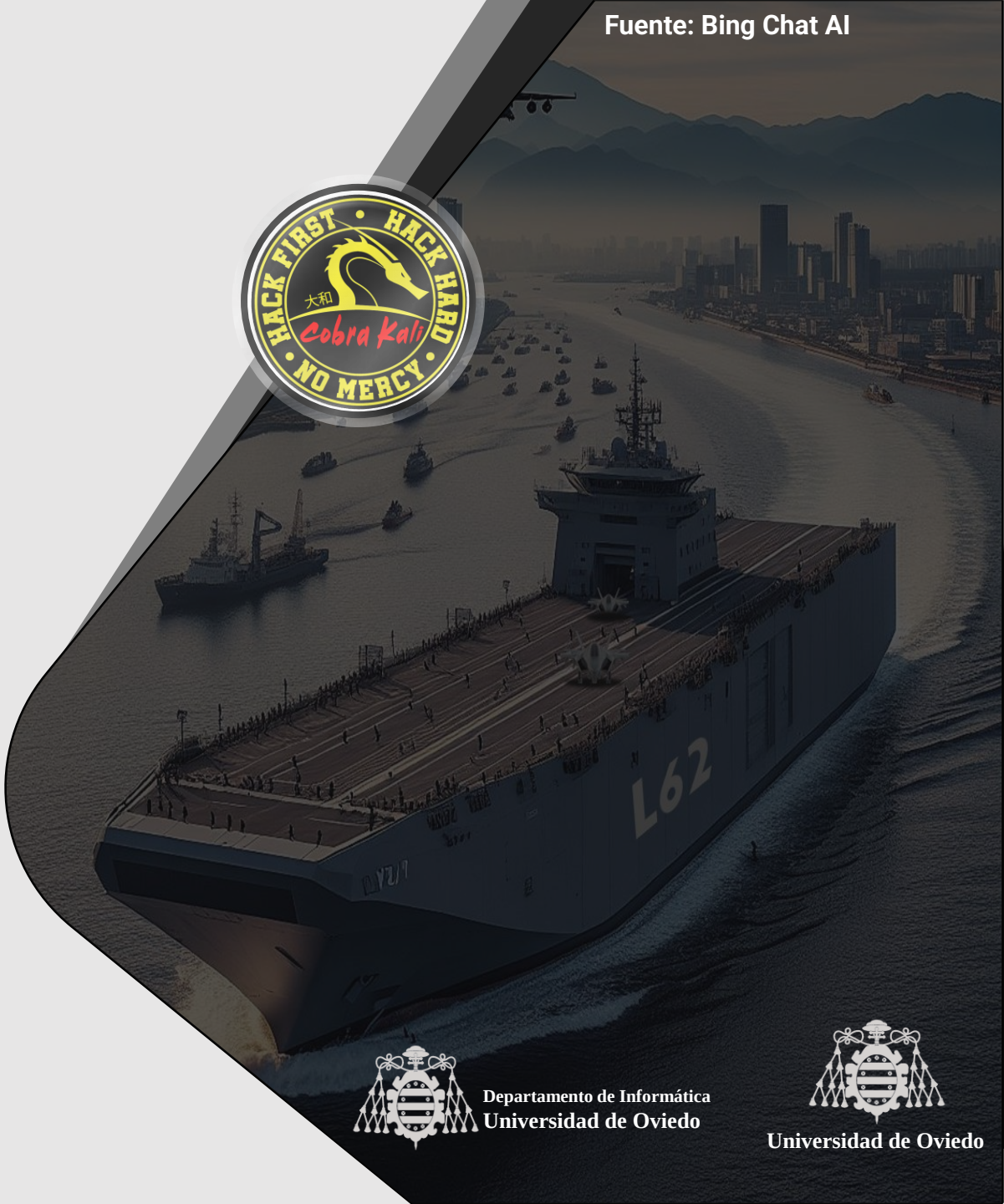
- `PHPSecInfo` (http://phpsec.org/projects/phpsecinfo/) **es una utilidad que informa del nivel de seguridad de una instalación de PHP existente en un servidor web**
  - Y ofrece consejos para mejorarla
  - Es similar a `phpinfo()`, pero da información relativa a seguridad
- **No hace ninguna clase de auditoría**
  - Pero es útil para comprobar que la instalación existente cumple con unos mínimos
- **Necesita simplemente copiarse en la carpeta donde se sirvan las webs de un servidor habilitado para procesar páginas PHP**
  - http://phpsec.org/projects/phpsecinfo/README

ApacheBench (ab)

José Manuel
Redondo López

- `ApacheBench` **es una herramienta para hacer benchmarking HTTP**
  - Comando `ab`, https://httpd.apache.org/docs/2.4/programs/ab.html
- **Permite saber el rendimiento aproximado de una instalación de Apache 2**
  - Especialmente cuantas peticiones por segundo es capaz de servir
- **Es útil para ver la capacidad de resistir ataques DoS y DDoS de nuestro servidor**
  - Sobre todo, cuando se han tomado medidas contra ellos, con cambios en la configuración o mediante programas tipo `PortSentry` o `Fail2Ban`
  - Se verán en "**F-103 Blas de Lezo**"
- **Si se combina con `top`, además se puede tener una idea del consume de CPU del servidor cuando se le somete a "stress"**
- **Es sencilla de usar**
  - https://www.devside.net/wamp-server/load-testing-apache-with-ab-apache-bench

< Ir al Índice



CONSIDERACIONES FINALES

Para finalizar...

MOD_ITK

José Manuel
Redondo López

- El MPM `prefork` de Apache 2 es más estable que el resto, pero más lento
- Existe un MPM novedoso llamado `mod_itk` (http://mpm-itk.sesse.net/)
  - También llamado `mpm-itk`
  - Permite ejecutar cada host virtual bajo un `uid` y `gid` distinto
  - Esto quiere decir que los scripts y ficheros de configuración de un host virtual no se pueden leer desde los otros
  - Se basa en el MPM `prefork`, por lo que no usa threads
- Esto implica que pierde rendimiento respecto a los que sí lo son e incluso es más lento que el `prefork` por requerir más computación por petición
  - Sin embargo, logra el mayor nivel de aislamiento entre host virtuales
  - Por ello, debemos elegir si queremos máximo aislamiento (seguridad) o más rendimiento
- El MPM `mpm-itk` viene incluido con muchas distribuciones de Linux
  - Incluida Ubuntu, por lo que su instalación es similar a la de otros MPM

Consideraciones finales

José Manuel
Redondo López

- **¿Todos los servidores web deben tener una misma configuración?**
  - Todos deberían tener una que les garantice un mínimo de seguridad para la función que cumplen
  - Pero, en general, no hay una configuración "buena para todos", depende de varios factores
    - Aplicaciones a servir y sus particularidades
    - Si estamos en producción o en desarrollo
  - Un entorno **de desarrollo podría** permitir
    - Ejemplos de los *frameworks* usados para desarrollar, documentación, etc. alojada en el servidor
    - Errores con información de depuración ampliada como trazas de pila, etc.
    - Más programas instalados que el propio servidor y lo que necesite para funcionar
      - Entornos de desarrollo, etc.
    - Comentarios en el código
    - Código no ofuscado (para poder entenderlo un poco mejor ☺)
    - *Backups*, versiones viejas, etc. alojadas en el propio servidor

Consideraciones finales

José Manuel
Redondo López

- **Pero en producción…**
  - Los errores deben ser todos personalizados de manera que den la menor información técnica posible
    - En cualquier caso, nunca revelar productos o versiones
  - Ofuscar el código y *scripts* para evitar que nos averigüen *frameworks*, productos, versiones, etc.
    - Aparte de `Pagespeed`, hay hasta herramientas online para ello: https://javascriptobfuscator.com/Javascript-Obfuscator.aspx
  - **NI UN SOLO COMENTARIO**: Pueden suministrar mucha información, incluso sin ser consciente de ello
    - Este punto y el anterior hace también la web más "ligera"
  - **JAMÁS** alojar nada que no pertenezca a la web en si
    - Ejemplos, *readmes*, *backups*, …
  - Log y configuración deben ir en sitios independientes no accesibles por el resto de los usuarios
- **Por tanto, usar el mismo servidor para desarrollo y producción es muy mala idea**

CONSIDERACIONES FINALES

José Manuel
Redondo López

- **Y si como desarrollador no queremos "cantar" la tecnología / *frameworks* utilizados para el desarrollo de las webs alojadas…:**
  - No usar estilos visuales por defecto ("esto canta a *Joomla*", "esto canta a *WordPress*"…)
  - Eliminar todo mensaje / comentario / *tag* `META` / etc. que delate el producto usado
    - Conlleva una revisión exhaustiva del código
  - No usar *scripts* ni CSS por defecto
    - Una alternativa es ofuscarlos
  - No usar plantillas o webs prefabricadas para un determinado propósito "tal cual"
  - Personalizar los mensajes de error
- **¿Por qué?**
  - *WhatWeb*, *BlindElephant* y otras son herramientas usadas para averiguar los *frameworks*, tecnologías y versiones usadas para el desarrollo de una web
    - Y con ello encontrar posibles vulnerabilidades
  - Se basan en la información de la *web* que acabamos de enumerar para ello
  - Úsalas sobre tu propia *web* para ver si "canta" o no ☺

Referencias

José Manuel
Redondo López

Referencias

- http://httpd.apache.org/docs/2.4/misc/security_tips.html
- http://www.tecmint.com/apache-security-tips/
- http://www.petefreitag.com/item/505.cfm
- http://www.hongkiat.com/blog/symlink-alias-folder/

FORTIFICANDO APACHE 2 DE FORMA PROFESIONAL