Extrelan 2008

Cáceres. Marzo de 2008

Desarrollo de Entornos Virtualizados de Intrusión

SG6 – Soluciones Globales en Seguridad de la Información

http://www.sg6.es

INDICE DE CONTENIDOS

Primera Parte: Conceptos Generales sobre Virtualización

Desarrollo de Entornos Virtualizados de Intrusión

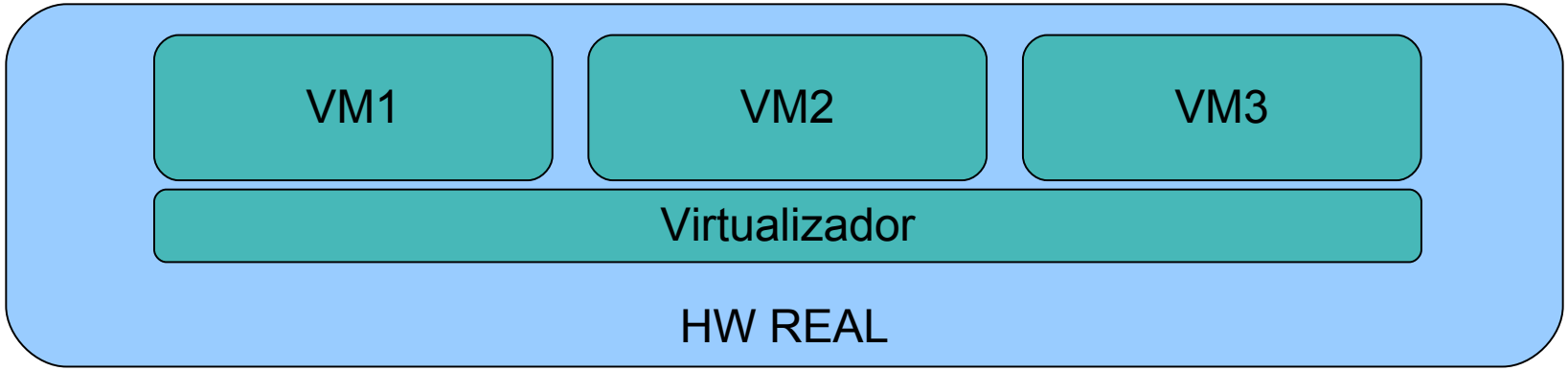
SG6 – Soluciones Globales en Seguridad de la Información

http://www.sg6.es

¿Qué es la virtualización?

- " .... un entorno entre la plataforma de una computadora y el usuario final, que permite que este ejecute un software determinado"

- La virtualización es la unión de una plataforma de hardware y un software "host" ("anfitrión", un programa de control) que simula un entorno computacional (máquina virtual) para su software "guest". Este software "guest", que generalmente es un sistema operativo completo, corre como si estuviera instalado en una plataforma de hardware autónoma. Típicamente muchas máquinas virtuales son simuladas en una máquina física dada.



Usos de la Virtualización

- **Consolidación de Sistemas Heterogéneos:** Múltiples servicios en múltiples arquitecturas pueden cohexistir sobre un sólo hardware real.

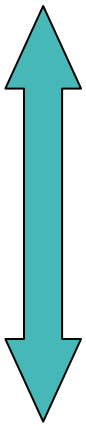
- **Despliegue efectivo de Arquitecturas:** Soluciones virtualizadas previamente pueden ser duplicadas y puestas en funcionamiento en cuestión de minutos.

- **I+D+i:** Permite trabajar sobre sistemas recién instalados sin esfuerzo, sometiéndolos a cualquier tipo de cambio, sin peligro, y pudiendo volver al estado original en cuestión de minutos.

- **Seguridad:** Permiten el aislamiento por servicios y por usuarios, permitiendo una mejor segmentación desde el punto de vista de la seguridad lógica.

Tipos de Virtualización

- Rendimiento
+ Versatilidad



+ Rendimiento
- Versatilidad

- **Emulación:** La máquina virtual simula un hardware completo, permitiendo que un sistema operativo sin modificar y diseñado para una Arquitectura CPU igual o diferente, pueda ejecutarse aisladamente.

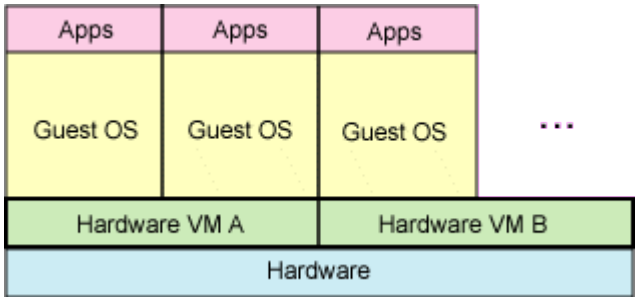
- **Virtualización Completa:** La máquina virtual simula un hardware suficiente para permitir que un sistema operativo sin modificar, pero diseñado para esa arquitectura, pueda ejecutarse aislado.

- **Paravirtualización:** La máquina virtual no necesariamente simula un hardware, sin embargo ofrece un API especial que solo puede usarse mediante la modificación del sistema operativo, permitiendo a este ejecutarse de forma aislada.

- **Virtualizacion asistida por HW:** Existe soporte HW al proceso de virtualización, ayudando de esta forma tanto a la virtualización completa, como a la paravirtualización.

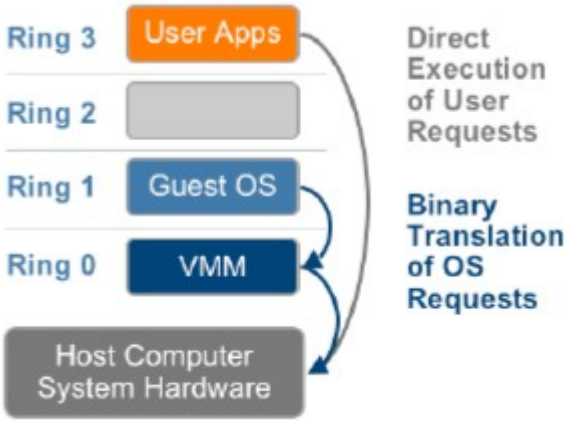
- **Virtualización a nivel de SO:** El entorno del sistema operativo virtualizado comparte el mismo sistema operativo que el del sistema real.

Tipos de Virtualización: Emulación



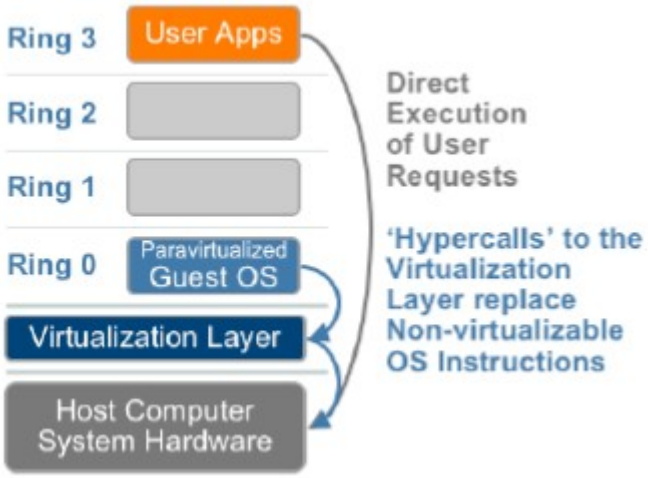
- Simula un entorno HW Completo.
- Ejecución en Espacio de Usuario.
- Virtualizador monitoriza de forma continua la máquina en ejecución:
  - Traduce instrucciones de la máquina virtual a la máquina real: permite ejecutar código compilado para otro tipo de CPU.
  - Intercepta el espacio de direcciones para controlar peticiones de I/O y de acceso a memoria.
- Escasa Eficiencia / Alta Versatilidad
- Ejemplos: Qemu, Bochs, etc.

Tipos de Virtualización: Virtualización Completa



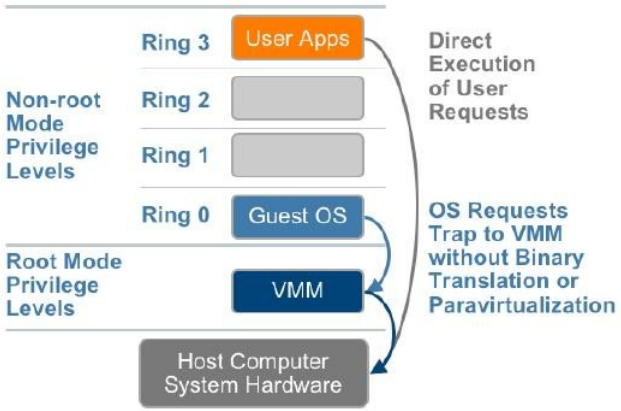
- Simula un entorno HW Parcial. Aprovecha las capas de abstracción de x86.
- Ejecución de Monitor en Ring0 (Kernel-Space).
- Monitor controla de forma continua la máquina virtual en ejecución (Ring1):
  - Ring1 es un anillo menos privilegiado, algunas instrucciones **NO** pueden ejecutarse en él.
  - En instrucciones con conflicto: Traducción binaria.
- Mayor Eficiencia / Buena Versatilidad
- Ejemplos: Vmware WS y GSX, KQEMU ...

Tipos de Virtualización: Paravirtualización



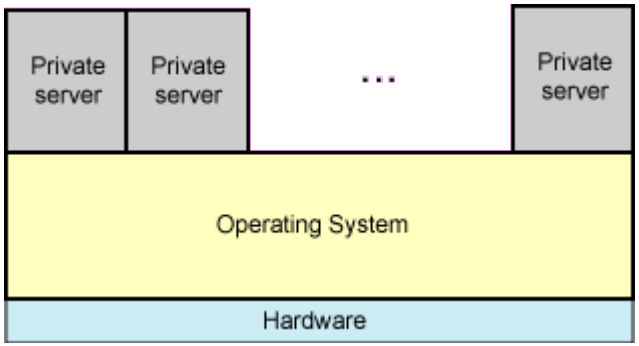
- No simula, el SO conoce que está siendo virtualizado. Usa el HW real.
- Ejecución en Ring -1: El Virtualizador es el proceso más privilegiado de todo el sistema.
- El SO llama al virtualizador cuando va a ejecutar instrucciones que pueden provocar un conflicto.
- Necesita un SO con el kernel modificado: limita su aplicación.
- Rendimiento Alto / Poca Versatilidad
- Ejemplos: VMWare Server ESX, Xen, UML ...

Tipos de Virtualización: Virtualización HW



- Problemas:
  - Traducción Binaria: Aunque permite ejecutar SO´s sin parchear, el rendimiento no es óptimo por la constante monitorización de instrucciones conflictivas.
  - Paravirtualización: Necesita de SO´s modificados para poder funcionar. Esto limita su aplicación.
- Solución: Asistencia del HW al proceso de Virtualización.
  - Intel VT-x y AMD-V: Capturan automáticamente llamadas conflictivas.
  - Permite un modo de privilegios mayor que ring0 donde se coloca el Hypervisor.
- Ejemplos: Xen 3.0, KVM, VMWare …

Tipos de Virtualización: Virtualización SO



- Núcleo modificado: Compartición de un sólo nucleo entre múltiples espacios de usuario aislados por éste.
- Cada servidor privado es una entidad separada, aislado del resto, con su propio espacio de direcciones, árbol de procesos, espacio de disco, etc.
- Mayor eficiencia: es capaz de virtualizar entorno a 300 servidores privados en un equipo con 2GB de RAM.
- Menor versatilidad: sólo se puede ejecutar un kernel. Eso limita a que todos los servidores privados sean idénticos: Linux, Solaris o BSD.
- Software: GNU/Linux OpenVZ, FreeBSD Jails, Solaris Containers …

Segunda Parte: Entornos de Intrusión Virtuales

Desarrollo de Entornos Virtualizados de Intrusión

SG6 – Soluciones Globales en Seguridad de la Información

http://www.sg6.es

¿Qué es un entorno de intrusión?

Simulación de carácter didáctico, lo más fidedigna posible, de un entorno real con uno o múltiples fallos de seguridad sobre los que desarrollar habilidades y conocimientos entorno a pruebas de intrusión y hacking ético.

Un poco de Historia

Descienden de los Wargames, literalmente “juegos de guerra”, cuyo objetivo era el hackeo real (competitivo) de un sistema dentro de un plazo dado. El mayor exponente de este tipo de juegos fue OpenHack, durante los años 1999, 2000, 2001 y 2002.

De esta idea nacen otro tipo de “retos”, cuya finalidad no es totalmente la competición, sino por el contrario el aprendizaje. Dentro de este tipo de retos encontramos a lo largo de los años una larga lista. Quizá los más significativos dentro de España:

- **Hackerslab:** Aparecido en el año 2000. Simulaba, originalmente durante 14 niveles, y posteriormente ampliado a 17, diversas técnicas de intrusión en un sistema Unix: shellsuid, IFS, race condition, stack buffer overflow, format strings, etc.

- **NGSec:** Aparecido en el año 2002, con posteriores versiones, hasta un total de 3, simulaba en un entorno web, por niveles, diferentes retos centrados en seguridad web.

- **Otros:** Izhal, Boinas Negras, CyberArmy

Tipos de Entornos de Intrusión

Hasta la aparición de SecGame, los diferentes entornos para la práctica de intrusiones que han ido apareciendo se pueden dividir en los siguientes tipos:

Basados en Retos: En este tipo de entorno aparecen pruebas aisladas unas de otras, sin correlación lógica entre ellas, y con carácter autoconclusivo. No tienen porqué simular escenarios reales, y cada reto se concibe como un todo. Un claro ejemplo de este tipo de Entornos, puede ser los proporcionados por “Un informático en el lado del mal”, en el carácter más lúdico, o los ofrecidos por OWASP en su WebGoat, con un carácter mucho más didáctico.

Basados en Niveles: Estos entornos plantean una sucesión encadenada de niveles, en los que no existe una lógica real, más que el avanzar y generalmente aumentar la dificultad sucesivamente. Cada nivel se suele poder superar con una técnica diferente de hacking. Ejemplos pueden ser NGSec o Hackerslab.

Basados en Aplicativos: En este caso el entorno simula una aplicación real con una lógica de funcionamiento inherente. Generalmente se hayan centrados en aplicativos web. Ejemplo de este grupo son la serie de aplicativos ofrecida por Foundstone: Hacme Bank o Hacme Casino.

SecGame: Entornos de Intrusión Virtuales

- ¿Qué es?
  - Evolución natural de los entornos de intrusión clásicos al aprovechar las características ofrecidas por las máquinas virtuales
- Objetivos:
  - Enfoque Didáctico del Proceso Técnico de Auditoría de Seguridad en SI.
  - Fidelidad y Realismo de los Escenarios: no hay niveles.
- **Ventajas:**
  - Versatilidad: Permiten simular todo tipo de ataques y escenarios: remotos, locales, web, etc.
  - Realismo: Simulan sistemas y entornos completos y fidedignos.
  - Libertad: El usuario puede experimentar y probar cualquier ataque o técnica sin limitación.
- **Desventajas:**
  - Complejidad de Diseño: No existen niveles, pero debe existir aislamiento en el proceso de intrusión.
  - Tamaño: Son máquinas virtuales completas con 2GB de tamaño.
  - Ausencia de Competitividad: Son juegos didácticos, no hay rankings

EIV: Problemáticas Asociadas

- Ausencia de Competitividad
  - Cambio del público objetivo y de la finalidad. Nuestro público son estudiantes y profesionales y la finalidad es didáctica.
- Tamaño
  - Compresión: Permite reducir el tamaño de la imagen entorno al 75%
  - Distribuciones Mínimas: Uso de Instalaciones Base.
- Complejidad
  - Principal problema.
  - Impacto: Tiempo desarrollo largo para un EIV completo.
  - Posibles Soluciones:
    - Guías, tutoriales y conferencias.
    - Reutilización de código.

Tercera Parte: Diseño de EIV´s

Desarrollo de Entornos Virtualizados de Intrusión

SG6 – Soluciones Globales en Seguridad de la Información

http://www.sg6.es

Diseño de EIV´s

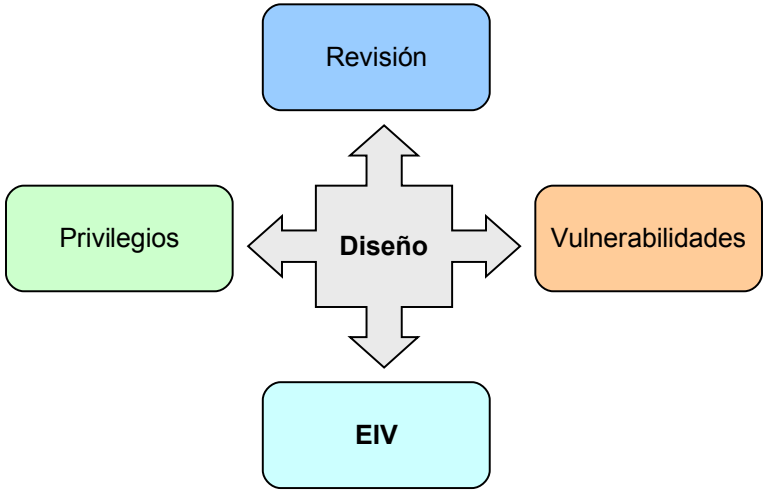
- Objetivo
  - Aislar y parcelar el avance de la intrusión sin establecer una diferenciación por niveles físicos. El aislamiento creará niveles lógicos, los cuales deberán ser superados para avanzar en la intrusión.

- Claves
  - La consecución del objetivo, si bien no es trivial, sí que depende en gran medida de la sistematización del proceso de construcción, mediante la combinación de 3 factores.

• **Revisión:** Los EIV´s se diseñan en función del tipo de revisión de seguridad a realizar: aplicativo web, servicios de red, privilegios locales, revisión de configuraciones, etc.

• **Privilegios:** Permiso, concedido por un estamento superior, para llevar a cabo una determinada acción que de otra forma no sería posible llevar a cabo.

• **Vulnerabilidades:** Errores que pueden ser aprovechados, de forma intencionada, para obtener privilegios.



Diseño de EIV´s: Revisiones

- El EIV se adaptará al tipo de revisión que queramos desarrollar. Los principales tipos de revisión de seguridad desde el punto de vista de la auditoría técnica son:

  - **Servicios Externos (Caja Negra):** Centrada en auditar todos los servicios existenes en un SI, repartido en uno o varios sistemas informáticos, desde el punto de vista del atacante, sin contar con acceso privilegiado a los mismo.

  - **Interna (Caja Blanca):** Centrada en auditar los servicios existentes en un SI, repartido en uno o varios sistemas informáticos, desde el punto de vista del administrador o de un usuario, contando con acceso privilegiado al mismo.

  - Dentro de estos grandes grupos se pueden encontrar auditorías más específicas, entre las que destacan:

    - **Servicios Web.**
    - **Servicios de Correo.**
    - **Sistemas de Gestión de Bases de Datos.**
    - **Wireless**

Diseño de EIV´s: Privilegios

- Según la Ubicación:
  - **Remotos:** Privilegios desde el exterior del sistema de información. Estos privilegios son concedidos por los servicios de red y por los aplicativos externos. Existirán tantos privilegios por nivel como servicios y/o aplicativos existan.
  - **Locales:** Privilegios desde el interior del sistema de información. Estos privilegios son concedidos por el sistema operativo.
- Según el Nivel:
  - **Públicos:** Concedidos a usuarios sin autenticación.
  - **De Usuario:** Concedidos a cada usuario tras su autenticación.
  - **De Grupo:** Concedidos al grupo de pertenencia del usuario.
  - **Administrativos:** Privilegios especiales otorgados al administrador o conjunto de administradores.

Diseño de EIV´s: Vulnerabilidades

- **Tipos de Vulnerabilidades**

Existen multiples clasificaciones para las vulnerabilidades, las más aceptadas de forma general son las siguientes:

- **CWE (Common Weakness Enumeration):** Es la mayor y más extensa catalogación de vulnerabilidades, contando en la actualidad con más de 600 entradas para catalogación, dividas en: configuración, código y entorno. Esta clasificación es la usada por CVE (Common Vulnerabilities and Exposures).

- **NVD (National Vulnerability Database):** Desarrollada por el Instituto Nacional de Estándares y Tecnología Norteamericano (NIST) es una catalogación más liviana que CWE. En su versión 2.1, contiene un total de 23 tipos de vulnerabilidades.

- **OWASP (Open Web Application Security Project):** Similar a NVD en extensión, pero centrada completamente en la seguridad de aplicativos web. Contempla 24 clasificaciones para las vulnerabilidades.

- **SAMATE (Software Assurance Metrics and Tool Evaluation):** También desarrollada por el NIST se centra en los errores de codificación de aplicativos.

Diseño de EIV´s: Vulnerabilidades NVD

- **Authentication Issues**
- **Credentials Management**
- **Permissions, Privileges, and Access Control**
- **Buffer Errors.**
- **Cross-Site Request Forgery (CSRF)**
- **Cross-Site Scripting (XSS)**
- **Cryptographic Issues**
- **Path Traversal**
- **Code Injection**
- **Format String Vulnerability**
- **Configuration**
- **Information Leak / Disclosure**

- **Input Validation**
- **Numeric Errors**
- **Command Injections**
- **Race Conditions**
- **Resource Management Errors**
- **SQL Injection**
- **Link Following**
- Other
- Not CWE
- Insufficient Information
- **Design Error**

Diseño de EIV´s: Matriz de Posibilidades (I)

- Existen tantas matrices **remotas** como servicios y aplicativos con acceso por red existan instalados en el sistema de información.

- Existe una **única** matriz local, a menos que el sistema se encuentre virtualizado y existan posibilidades de vulnerar la seguridad del virtualizador.

- El movimiento en el horizontal por la matriz se debe hacer de izquierda a derecha.

Errores / Privilegios	Público	Usuario	Grupo	Administrador
Authentication Issues				
Credentials Management				
Permissions, Privileges, and Access Control				
Buffer Errors				
Cross-Site Request Forgery (CSRF)				
Cross-Site Scripting (XSS)				
Cryptographic Issues				
Path Traversal				
Code Injection				
Format String Vulnerability				
Configuration				
Information Leak / Disclosure				
Input Validation				
Numeric Errors				
Command Injections				
Race Conditions				
Resource Management Errors				
SQL Injection				
Link Following				
Design Error				

- El movimiento horizontal no debe ser obligatoriamente columna a columna.

- El movimiento vertical no presenta ningún tipo de limitación.

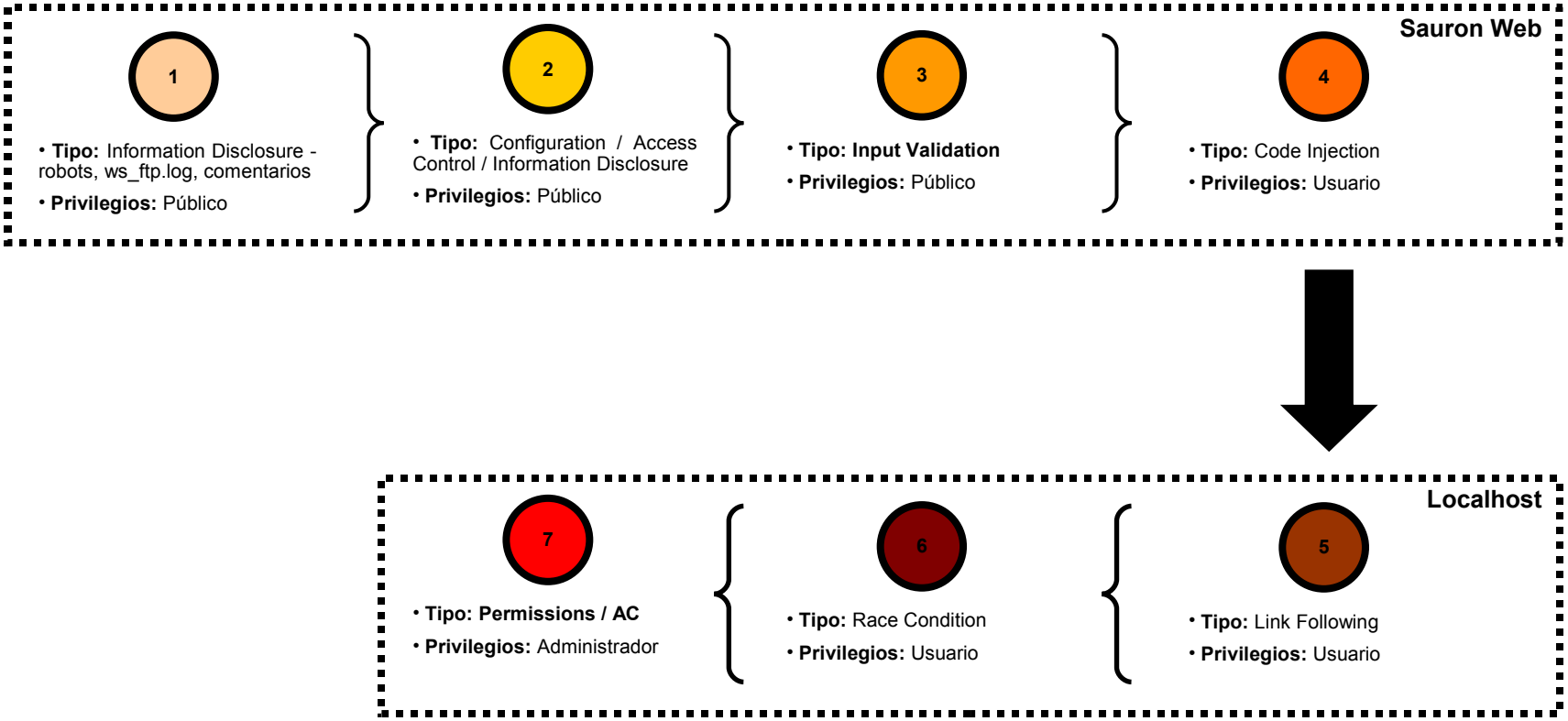
Diseño de EIV´s: Matriz de Posibilidades (II)

- En función del tipo de **revisión** deberemos tener en cuenta lo siguiente:
  - En las revisiones de seguridad en caja negra **debe** existir un acceso público al sistema.
  - En las revisiones en caja blanca debemos proporcionar el usuario de acceso al sistema.
- El paso al sistema **local** desde **remoto** se realiza habitualmente mediante 2 técnicas:
  - **Ejecución Remota de Comandos:** usuario del servicio de red.
  - **Consecución de Acceso Remoto:** usuario del que se adquieren los privilegios.

Errores / Privilegios	Público	Usuario	Grupo	Administrador
Authentication Issues				
Credentials Management				
Permissions, Privileges, and Access Control				
Buffer Errors				
Cross-Site Request Forgery (CSRF)				
Cross-Site Scripting (XSS)				
Cryptographic Issues				
Path Traversal				
Code Injection				
Format String Vulnerability				
Configuration				
Information Leak / Disclosure				
Input Validation				
Numeric Errors				
Command Injections				
Race Conditions				
Resource Management Errors				
SQL Injection				
Link Following				
Design Error				

Diseño de EIV´s: Ruta de Explotación

- A partir de la matriz para cada uno de los servicios externos y de la matriz local se puede obtener una ruta de explotación, el cual muestra el camino lógico para avanzar en el EIV.



Cuarta Parte: Desarrollo de un EIV – extreHACK 2008

Desarrollo de Entornos Virtualizados de Intrusión

SG6 – Soluciones Globales en Seguridad de la Información

http://www.sg6.es

Fases de Desarrollo

- Fase 1: Instalación del Sistema Base
  - En esta fase crearemos la máquina virtual básica para la ejecución de un EIV. Esta máquina corresponderá con una instalación base de Fedora 7. Sobre este sistema se procederá a la actualización y minimización de servicios.
- Fase 2: Diseño del EIV
  - En esta fase se diseñará el entorno de intrusión, acorde a lo visto en los puntos de la tercera parte de este documento.
- Fase 3: Implementación
  - Una vez diseñado el EIV se llevará a cabo la implentación dividida en los siguientes apartados:
    - Instalación de paquetes necesarios.
    - Configuración de los servicios y aplicaciones instalados.
    - Implementación de las fases en el sistema.
- Fase 3: Pruebas y Resolución
  - Una vez implementado, se verificará el correcto funcionamiento y se procederá a su resolución, para comprobar la viabilidad del mismo.

Fase 1: Instalación del Sistema Base

- Elementos Necesarios:
  - VMWare Workstation
  - ISO Fedora 7

- Procedimiento:
  - Crear una Nueva Máquina Virtual en VMWARE

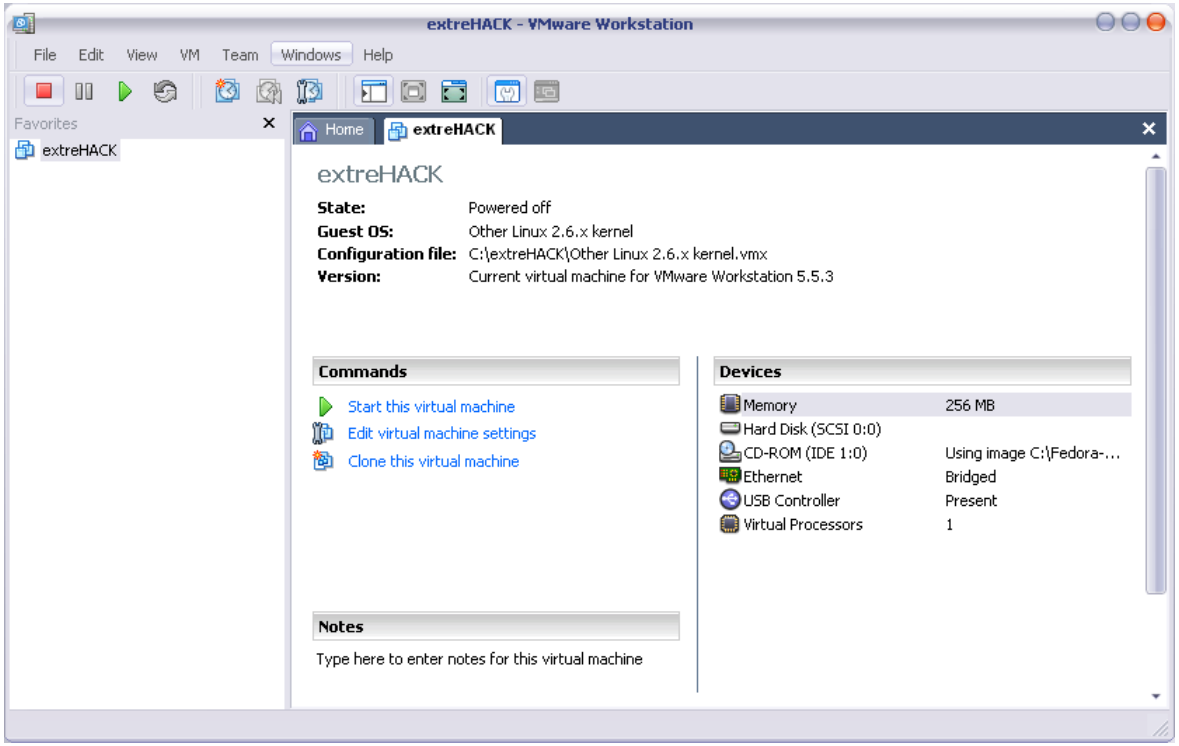
  - Realizar una instalación BASE del sistema sobre la máquina virtual.

  - Securizar el sistema a un nivel básico:
    - Minimizar el número de servicios activos en el sistema.
    - Actualizar los paquetes a las últimas versiones

    - **Adicionalmente (no contemplado):**
      - Endurecimiento de los permisos por defecto.
      - Tweaks en autenticación
      - Modificaciones en el sistema de ficheros
      - Firewall
      - Módulos de seguridad avanzados: SELinux, etc.

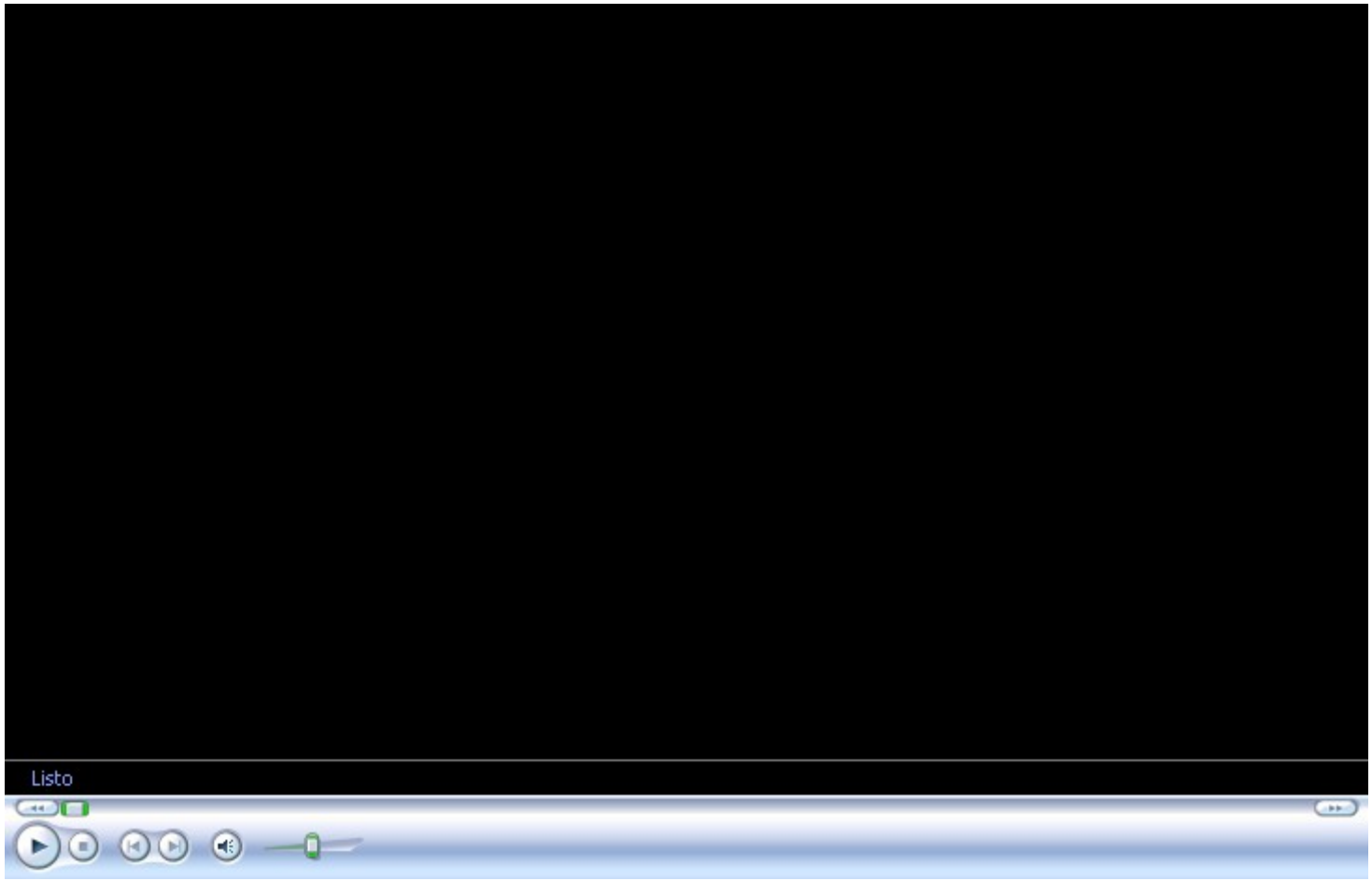
Fase 1: Instalación del Sistema Base

- Creación de una Nueva Máquina Virtual en VMWare:
  - File. New. Virtual Machine: Creamos una máquina con 128MB de RAM y 2GB de HD y Bridge Networking. El CDROM se asociará a la ISO de FC7



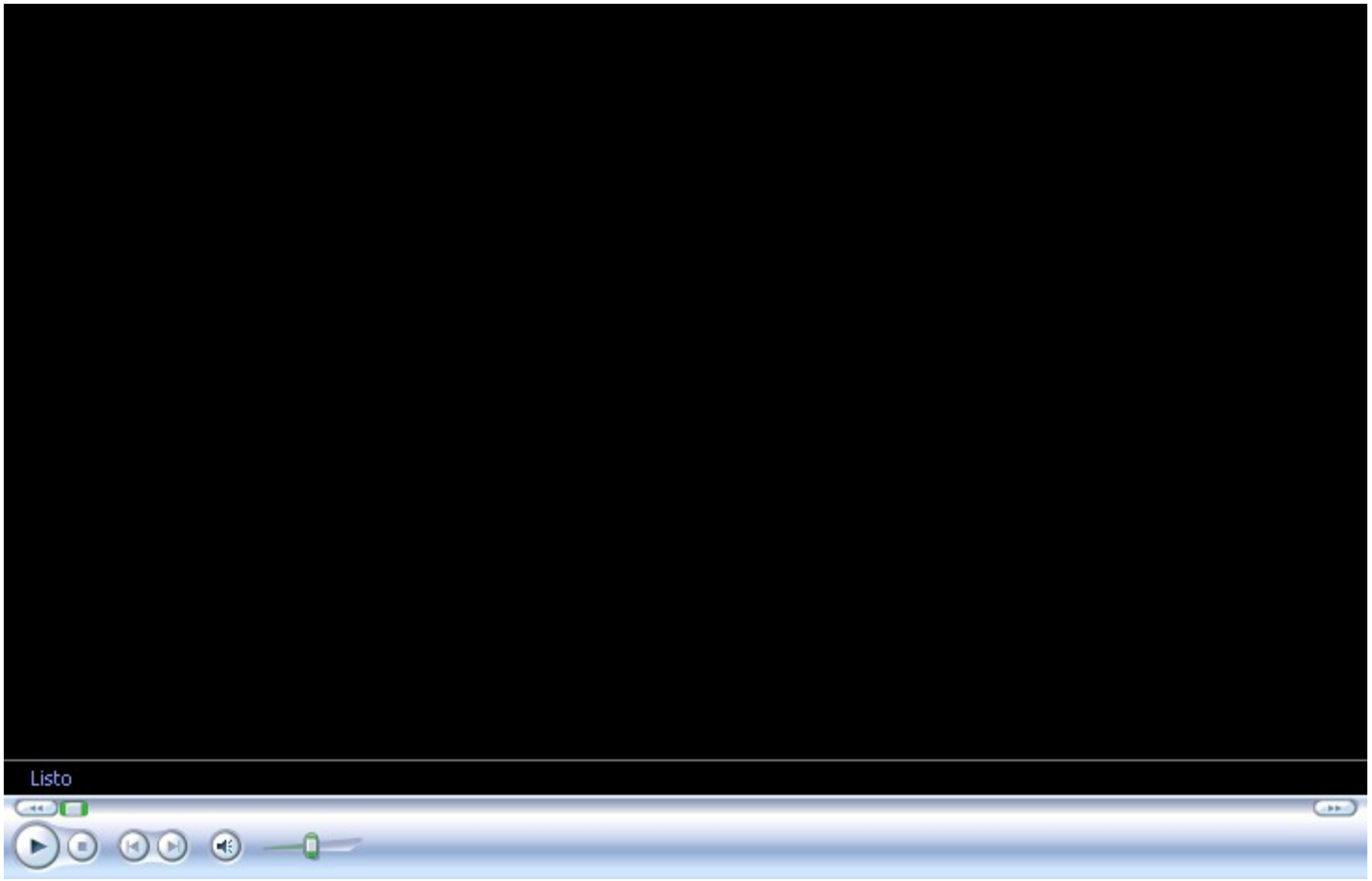
Fase 1: Instalación del Sistema Base

- Instalación de la BASE de FC7 en la máquina virtual:



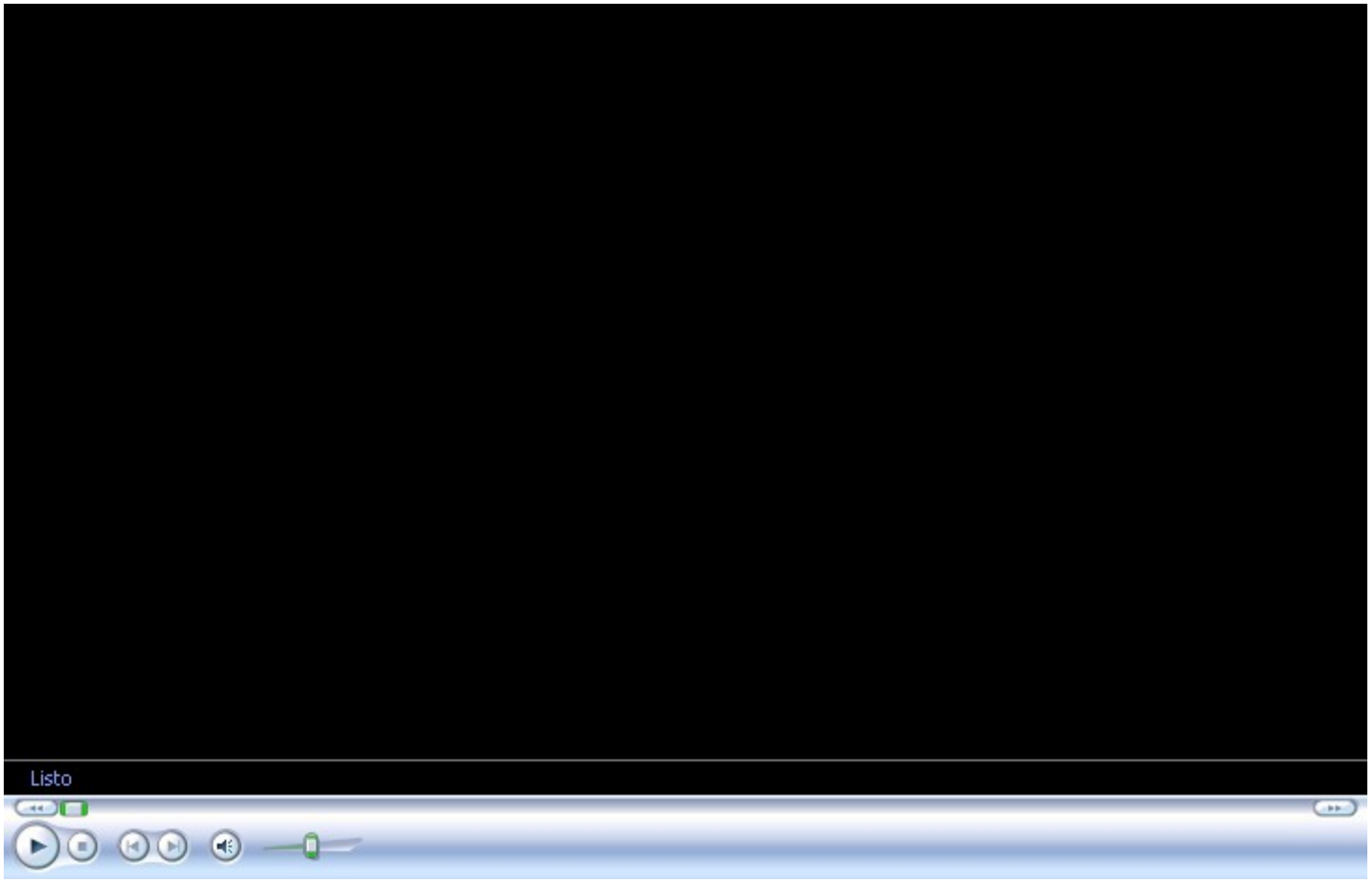
Fase 1: Instalación del Sistema Base

- Minimización del número de servicios activos (firstboot):



Fase 1: Instalación del Sistema Base

- Actualización de la distribución (yum update):



Fase 2: Diseño del EIV - ExtreHACK

- Objetivos:
  - Intervenir Múltiples Servicios y Aplicativos.
  - 3 Niveles
  - Complejidad Baja

Fase 2: Diseño del EIV - ExtreHACK

- Nivel 1 – FTPD Information Disclosure
  - RoadMap:
    - Existencia de usuario anónimo en el FTPD
    - Banner con información relevante al autenticar a un usuario anónimo

Fase 2: Diseño del EIV - ExtreHACK

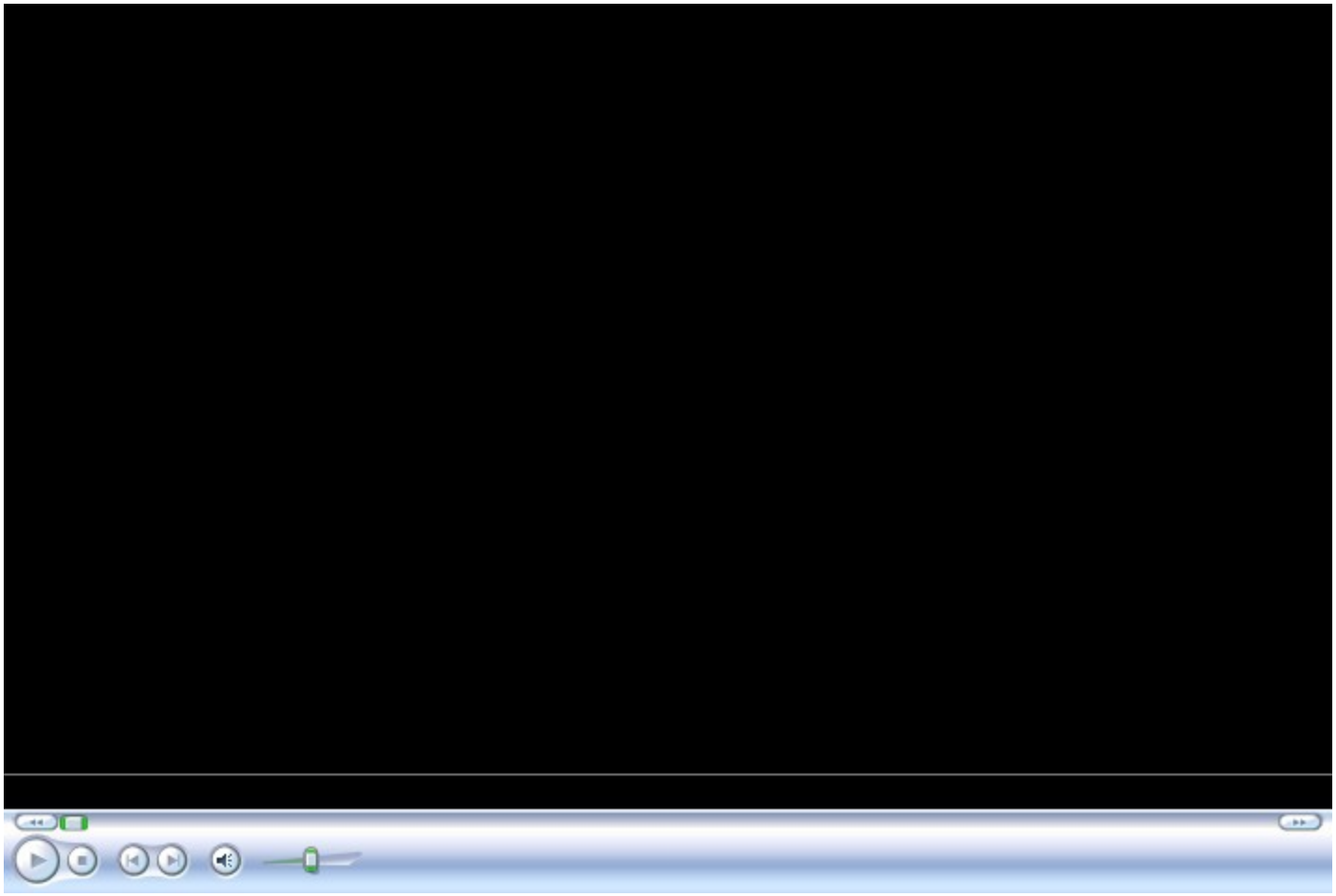
- Nivel 2 – HTTPD Auth Bypass
  - RoadMap:
    - Autenticación basada en cookies: cookie con referencia a status administrador.
    - Almacenaje de identificación permanente en HD visualizable remotamente.
    - Directory Indexes en Apache

Fase 2: Diseño del EIV - ExtreHACK

- Nivel 3 – HTTPD Auth Bypass
  - RoadMap:
    - Autenticación basada en cookies: cookie con referencia a status administrador.
    - Almacenaje de identificación permanente en HD visualizable remotamente.
    - Directory Indexes en Apache

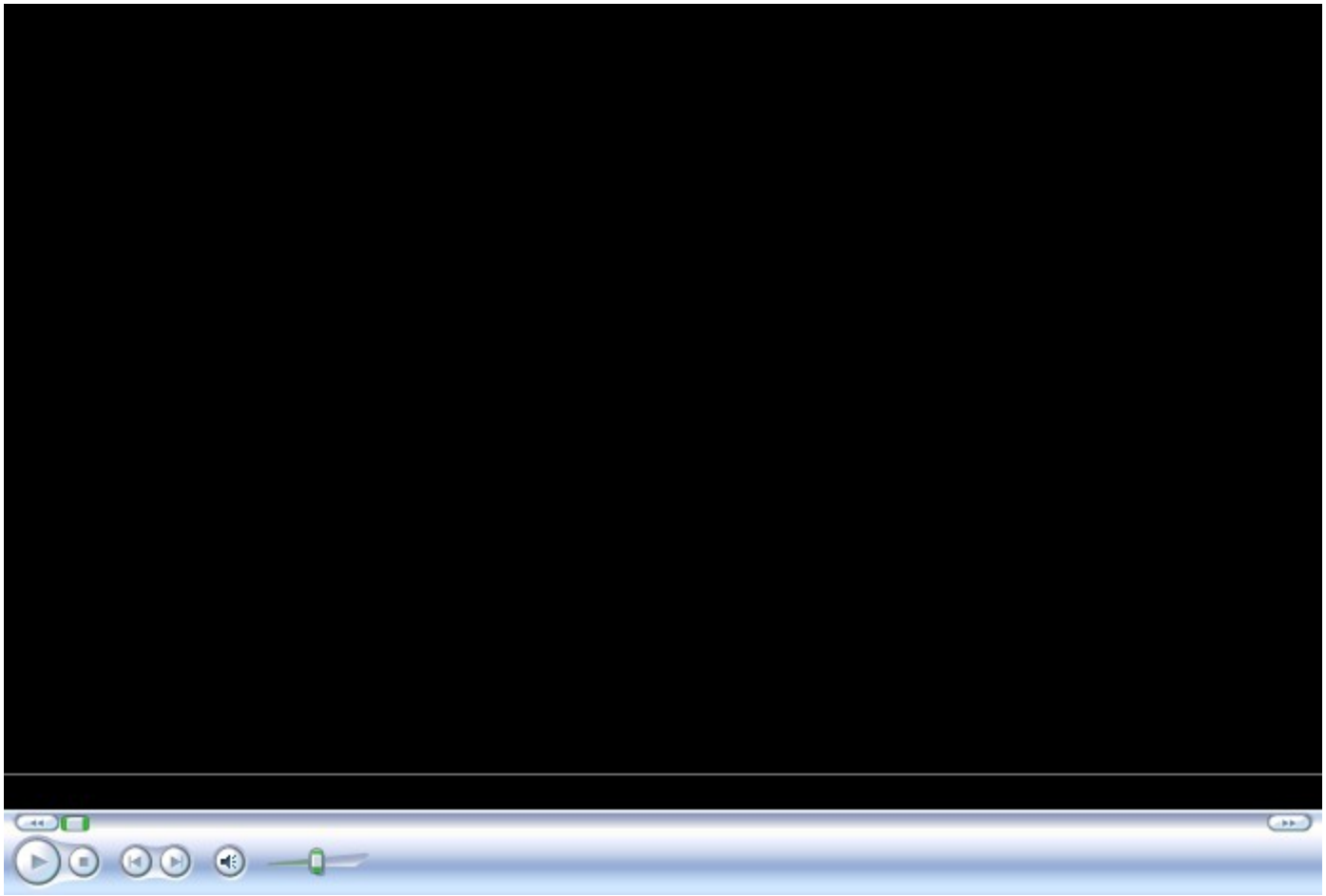
Fase 3: Implementación del EIV

- 1º Instalación de Servicios LAMP + FTPD



Fase 3: Implementación del EIV

- 2º Information Disclosure FTPD



Fase 3: Implementación del EIV

- 3º Auth Bypass

```
<?
if (isset($_COOKIE["admin"])) {
   if ($_COOKIE["admin"]=="true") {
         if (isset($_COOKIE["id"])) {
            if (file_exists("./id/".$_COOKIE["id"])) {
                  readfile("auth-success-2008.inc");
                  exit(0);
            }
         } else {
            readfile("notid.inc");
            exit(0);
         }
   }
}
setcookie("admin","false");
if (isset($_POST["user"])) {
   readfile("fail.inc");
} else {
   readfile("login.inc");
}
?>
```

+ HTTPD Indexes

+ id/ Directory

Fase 3: Implementación del EIV

- 4º Instalación Insegura de phpMyAdmin

  - Configuración incorrecta por unas malas políticas de seguridad.
  - Uso de ficheros .bak (sin protección de acceso) por un cambio de configuración.

Fase 4: Implementación del EIV

Demostración de la Resolución

Ruegos y Preguntas

GRACIAS A TODOS POR VUESTRO TIEMPO