UNIDAD 1

ANEXO

Seguridad – clase del 26/08

(para resumir después) Repaso de hash, Seguridad lógica y física, BYOD, Rastreo, copias de seguridad, VPN, impacto de todo eso en las organizaciones, Básicamente todo el pdf de introducción y Unidad 1

Hoy Unidad 2

Daño: perjuicio que se produce cuando un sistema falla. Debe ser cuantificable

Riesgo: Producto entre la magnitud de un daño y la probabilidad de que suceda

Amenaza: Situación de daño cuyo riesgo de producirse es significativo

Vulnerabilidad: Deficiencia en un sistema susceptible de producir un fallo. Tipos de vulnerabilidad:

* Debidas a la implementación
* Debidas al diseño
* Debidas al uso

Áreas de vulnerabilidad en entornos web:

* Cliente
* Red
* Servidor

Exploit: Técnica que permita aprovechar una vulnerabilidad para hacer daño

ATAQUE A APLICACIONES CONOCIDAS

CVE – Common Vulnerabilities Exposures (Vulnerabilidades y amenaza común) es un código asignado a una vulnerabilidad que le permite ser identificada de forma univoca. Creada por MITRE corp y permite conocer una vulnerabilidad en un programa o sistema – se identifica CVE – AÑO – NUMERO

CWE – Common Wakness Enumeration (Enumeracion de debilidades comunes) es una lista de tipos de debilidades de software y hardware, se identifica CWE – numero

NVD – National vulnerability Database, del NIST repo del gobierno de EEUU para la gestión de datos de vulnerabilidades basados en los estandares

CVSS – Common Vulnerability scoring System, es uyn conjunto de estandares abiertos para asignar valores o puntos de severidad. Va de 0 a 10

PREVENCION DE VULNERABILIDADES

* Listas bugtraq -> seclists.org/bugtraq
* Sistemas automáticos de análisis
  + DAST – Dynamic application security testing. Scanners de vulnerabilidades -> Vega, OpenVas, Uniscan
  + SAST – Static Application Security Testing. Auditoria automatica de codigo -> PMD, SonarQube, bandit
  + IAST – Interactive Application Security Testing. Detectan vulnerabilidades en tiempo real durante la ejecución de una app -> Contrast Assess, Contrast Security, Hdiv security
  + Redes Trampa -> honeynet.org

CERT/CSIRT

CERT – Computer Emergency Response Team

CSIRT – Centro de Respuesta a Incidentes de Seguridad Informática

Equipos responsables de gestionar los incidentes de seguridad informáticas según su alcance y comunidad. Interactúan entre si para facilitar info oportuna para actuar frente a diferentes tipos de incidentes, determinar su impacto, alcance y naturaleza, comprender las causas, investigar soluciones, coordinar y dar apoyo para la implementación de las estrategias de respuesta con las partes involucradas, difundir info sobre incidentes mas frecuentes y mitigación de sus defectos, coordinar y colaborar con otros actores

FUNCIONES

* Ayudar al publico objetivo a atenuar y prevenir incidentes de seguridad
* Protege info valiosa
* Coordina de forma centralizada la seguridad de la info
* Guarda evidencias por si hay pleitos
* Apoya y preseta asistencia a usuarios para recuperarse de incidentes
* Dirige de forma centralizada las espuesta a los incidentes, promueve confianza

DENEGACION DE SERVICIO (DoS)

Ataque DOS -> ataque a un sistema o red que causa que sea inaccesible a los usuarios legítimos, provoca perdida de conectividad por el consumodel ancho de banda de la red de la victima o sobrecarga de recursos

* Volume-based DDoS attacks: el atacante inunda a la victima con gran cantidad de conexiones de red
* App DDoS attacks: El atacante opera a nivel app usualmente por http intentndo saturar el server
* Low-rate Dos: El atacante usa una vulnerabilidad en el diseño o implementación de la app

FLOODING

Busca generar solicitudes maliciosa a un servicio para que el mismo se sature o entre en modo de espera. Esto limita o anula el funcionamiento

BotNet

Conjunto de terminales que ejecutan software que permite control desde ubicaciones remotas. Las terminales se denominan bots o zombies

Texto

Descripción generada automáticamente

\*MAN IN THE MIDDLE Es un Escalon evolucionado del sniffer, interceta paquetes

SNIFFERS (Escucha comunicación HTTP)

Programa de captura de trampas de red, se usa para gestionar la red con una finalidad docente, aunque también malicioso

Hay para LAN y algunos son Ethereal, WinPcap, Ettercap, WinDump, WinSniffer,Hunt

Para Wireless, Kismet, netwokr Stumbler

El sniffer se conecta a un Hub conectado a los equipos y también puede ser por WIFI

ATAQUE A NAVEGADORES (CLIENTE)

* Tampering o Data Diddling: Modificacion no autorizada de la info. Ej, múltiples sitios web han sido afectados al detectar cambios en el contenido de sus paginas
* Mediante JS: Programas para detectar fallas de seguridad de navegadores Web y correos
* Drive-by download: Infectan a los usuarios simplemente ingresando a una web. Mediante esta técnica los desarolladores del malware propagan e inyectan código dañino en el original

Texto, Carta

Descripción generada automáticamente

OTROS ATAQUES A CLIENTES

* Hjackers: Programas que alteran el funcionamieno del cliente para que se pueda robar info. Ej: Page hijacking, Session, Browser
* Rootkits: Permiten que una app permanezca oculta en el sistema o que la misma no se puede eliminar
* Backdoors: habilita un acceso alternativo al sistema evitando el método de autenticación principal. Se suelen instalar en los sistemas comprometidos para su posterior uso remoto DE PARTE DEL ATACANTE
* Stealers: Acceden a la info del equipo. Contraseñas reecordadas.
* Keylogger: registran la actividad de los dispositivos de entrada, (teclado)
* Ransomware: retienen el control del euipo o cifran info almacenada para que no pueda ser accesida (Pagame para devolverte tus cosas)

OWASP TOP TEN

FACTORES DE DATOS}

Tasa de incidencia: % de las apps que tienen al menos una instancia vulnerable. No importa si fue algo puntual o sistémico.

CWEs Mapeadas: Nro de CWEs (Debilidad) en una categoría

Tasa de incidencia: % de apps vulnerables a esa CWE de la población analizada por esa org ese año

Cobertura: % de apps testeadas por las org para una CWE

Explotabilidad ponderada

Impacto ponderado

Total de ocurrencias

Total de CVEs

TOP TETE WEB APP SECURITY RISKS 2021

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

A01 – PERDIDA DEL CONTRO DE ACCESO

Muchas apps web verifican el acceso a las funciones. Pero también hay que hacer eso del lado del servidor, si no se hacen los hackers pueden forzar peticiones para acceder a funciones sin permiso. Por URL, Ej que alguien no ADMIN pueda acceder a /admin bla bla

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Se cambia el acct y si no esta verificado el hacker entra a cualquier cuenta

O también, se puede acceder sin autorización por URL

PREVENCION

* Proceso de acceso y permisos actualizable y auditable
* Negar accesos por defecto con ROLES
* NO implementar controles en la capa de visualización

A02 – FALLAS CRIPTOGRAFICAS

Info sensible demanda protección adicional como la encriptación en su almacenamiento y transito

EJ: App que encripta Números de tarjetas con cifrado automático de bdd. Lo que significa que se descifra al ser solicitada. Para eso necesitamos un soporte criptografico adecuado

Ej: Robo de Cookies viendo la conexión de HTTPS a HTTP y de ahí saca la sesión del usuario y a sus datos privados

Ej: La base de pass utiliza unsalted hashes una debilidad de carga de archivos deja que el hacker tenga ese archivo

ENTONCES -> Cifrar bien, no almacenar datos sensibles al pedo, aplicar algoritmos fuertes de cifado y usar CLAVES ROBUSTAS Xddd, Deshabilitar autocomplete

A03 – INYECCION (de bdd)

Ocurren cuando datos no confiables se mandan a un interprete como cmd o consulta y así te fuiste hackeado maquinola

Captura de pantalla de un celular

Descripción generada automáticamente

PREVENCION

* Uso de apis
* Codificar caracteres especiales
* Listablanca
* Usar LIMIT

REFERENCIA CSRF

App que permite que los usuarios envíen peticiones de cambio de estado, que no incluyen nada secreto. El hacker incluye una etiqueta img o iframe donde la victima clickea y va en lugar de al lugar original

REFERENCIA XSS – Cross Site Scripting

Ocurren cada que una app toma datos no confiables y los manda al navegador sin valida. XSS permite a los hacer ejecutar secuencias de comando en el nav secuestrando sesiones o dirigir a la victima a sitios malos

* XSS reflejado: La app usa datos de un usuario sin ser validados. ENLACE,
* XSS Almacenado: La app almacena datos sin validar lo que se entraga a un admin. Este XSS es de riesgo alto
* XSS Basados en DOM: Datos modificables como JS

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

PREVENCION:

* Codificar datos no confiables basados en HTML
* Validacion de lista blanca
* Usar APIS de auto sanitización
* Política de seguridad de contenido CSP

A4 – Diseño Inseguro

Riesgos de diseño.

Ej: Cadena de cine permite descuentos en la reserva de hasta 15 personas antes de un depósito. El atacante puede modelar el flujo y probar si pueden reservar 600 asientos en todos los cines a la vez para provocar perdidas

Ej: Web de comercio minorista no tiene bot protect administrados por revendedores. Publicidad terrible para fabricantes y propietarias.

PREVENCION:

* Patrones de diseño seguro
* Lenguaje y controles de seguridad en historia de usuario
* Ciclo de vida de desarrollo seguro
* Verificaciones de admisión en cada nivel de app
* Hacer pruebas de integración y unitarias para validar flujos
* Casos de uso y uso indebido
* Separar capas de niveles en el sistema y la red

A5 – Configuración de seguridad incorrecta

Una buena seguridad requiere tener definidos configuraciones seguras para la app. Todo esto debe definirse ya que no son seguros por defecto.

Ej: App basada en ambiente de trabajo como sping. Presenta defectos de XSS se libera actualiacion para corergirlo. HASTA QUE NO SE ACTUALICE SUE PUEDE SEGUIR ATACANDO

EJ: CDM de admin del server de apps esta instalada y no fue removida, las cuentas predeterminadas no se cambiaron. El atacante descubre que las pags de admin están activas y se registran con claves predeterminadas (Como si en el Sheila la cuenta de admin clave Admin 123 siga intacta)

PREVENCION:

* Usar un proceso rápido, fácil y repetible de fortalecimiento para estar asegurados
* Un proceso para mantener las actualizaciones y parches en cada entorno
* Una Arquitectura de app que proporcione separación segura y efectiva entre componentes
* Escanear y auditar regularmente para ver fallos de config o parches omitidos

Entidad Externa de XML (XXE)

Se pueden usar para revelar archivos internos usando un gestor UTI, compartir archivos SMB internos en servers no actualizados, escanear puertos, ejecutar código remoto yt denegar servicios

EJ: Atacante intenta extraer datos del serverTexto, Carta

Descripción generada automáticamente con confianza media

EJ: Atacante sondea la red cambian la linea ENTITY por:

EJ: Atacanteintenta una denegacion de servicio inclclyendo un archivo infinito

PREVENCION

* Usar formatos de datos no tan complejos como JSON y evitar serialización de datos
* Actualizar Todo lo que sea XML
* Usar validadores de dependencias y SOAP a 1.2 o mas
* Deshabilitar XXE y procesamiento DTD, Validar el XML entrando usando XSD
* Usar Lista blanca

A6 – Componentes Vulnerables y Desactualizdos

Las librerías, framework y otros módulos de software, se usan con privilegios. Si un componente vulnerable es explotado podria dares lugar a un ataque que facilite una serie de perdidas de datos o comprometer al server

Apache CFX Authentication By pass: por fallar al proveer un bloque de id, los atacantes pudieron invocar cualquier web servie con permisos completos

Spring Remote Code Exxecution: Abusando de le implementación del Expression Language en Spring se permitia a los atacantes la ejecución de código arbitrario, tomando posesión efectiva del server

Disositivos de internet de las cosas (IoT): frecuentemente son imposibles o muy difíciles de actualizar, la importancia de estas actualizaciones puede ser enorme

PREVENCION

* Id todos los componentes y versión usadas
* Revisar la seguridad del componente en DB publicas, lista de mails y actualizarlos
* Establecer políticas de seguridad que regulen el uso de componentes , como requerir practicas en el desarrollo, Test de seguridad y licencias aceptables
* Seria apropiado agregar capas de seguridad alrededor del componente para deshabilitar funcionalidades no usadas y asegurar aspectos débiles o vulnerables del componente

A7 – FALLAS DE ID Y AUTH

Funciones relacionadas a autenticación y gestión de sesiones, se suelen usar mal permitiendo a los atacantes comprometer pass, tokens o explotar fallas para ASUMIR IDENTIDADES de users

EJ: RELLENO DE CREDENCIALES, USO DE CONTRAS

EJ: ATAQUES OCURREN CUANDO SE USA PASS COMO UNICO FACOO

EJ: MAL MANEJO DE SESIONES (NO CERRARLAS)

PREVENCION:

* Usar un único y robusto conjunto de controles de autenticación y sesiones
* Evitar vulnerabilidades de XSS que puedan dar robos de ID de sesión
* Auth multifactorial
* Control CONTR PASS DEBIL
* Limitar o incrementar tiempo de respuesta en cada intento de inicio de sesión
* Rotación de contras

A8 – FALLAS EN EL SW Y EN LA INTEGRGRIDAD DE LOS DATOS

ESTAN relacionados con CODIGO e INFRAESTRUCTURA NO PROTEGIDOS, DEPENDENCIA DE COSAS NO CONFIABLES. Actualizaciones automáticas adulteradas u objetos serializados

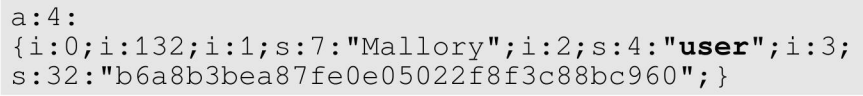
EJ: Actualizaciones no firmadas: Routers de hogar – No verifican FIRMWARE. FIRMWARE sin firmar 🡪 Objetivo de atacantes creciente.

EJ: Deserialización insegura: REACT con SpringBoot quieren inmutar código SERIALIZANDO la sesión y enviar los compo con cada soli. EL ATACANTE lo identifica (STRING: “rO0”) y lo ejecuta remotamente con JAVA SERIAL KILLER

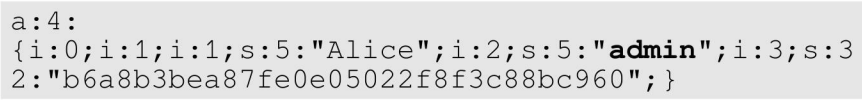
DESERIALIZACION INSEGURA (ANEXO)

Ocurre cuando una app recibe objetos serializados hostiles. Conduce a la ejecución REMOTA DE CODIGO,

EJ: Atacante pude modificar el objeto serializable de un foro PHP que contiene hasheada la PASS USUARIO, ROL, etc.



Asi



PREVENCION:

* + No aceptar objetos serializados de cualquiera
  + Verificación de integridad
  + Restringir las conexiones de red entrantes y salientes

A9 – FALLAS EN EL REGISTRO Y MONITOREO

Permiten persistir en el tiempo el ataque al sistema, pivotear a mas sistemas y manipular datos. El tiempo de detección de una violación de seguridad es mayor a 200 dias, siendo detectados por 3eros y no por procesos internos

EJ: SW de un foro de código abierto es hackeado usando una falla de seguridad. Se elimino el repo del código fuente que contiene la próxima versión y todos los contenidos del foro. Aunque el código fuente se pueda recuperar, la falta de monitorización condujo a una brecha PEOR.

EJ: Un loco tiene un sandbox de análisis de malware interno para archivos en mails. Se detecta uno pero nadie le dio pelota. El sandbox manda varias advertencias antes de que la brecha fuera detectada debido a Fraudes de tarjeta

PREVENCION:

* Todos los erroresde login, control de acceso y validación de entradas de dato del lado del server, se deben registrar con el contexto de user suficiente para identificar CUENTAS SOSPECHOSAS
* Transac de alto impacto tengan pista de auditoria con controles de integridad para prevenir alteración o delete
* Transac de alto valor IDEM ANTERIOR

A10 – FALSIFICACION DE SOLICITUDES DEL LADO DEL SERVER (SSRF)

Ocurren cuando una app esta obteniendo algo remoto sin validar la URL proporcionada por el usuario. Permite que un atacante coaccione a la app para que envie una soli falsificada a un destino inesperado

PREVENCION:

* Validar los datos de ENTRADA proporcionados por el CLIENTE
* Cumplir un esquema de URL, puerto y destino
* No envie respuestas en formato CRUDO a clientes
* Deshabilitar las redirecciones HTTP

NO MITIGAR SSRF MEDIANTE EL USO DE UNA LIOSTA DE DENEGACION O EXPRESION REGULAR. LOS HACKERS TIENEN LISTAS DE PAYLOADS, HERRAMIENTAS Y HABILIDADES PARA ELUDIARLAS

EJ: Escaneo de puertos de servidores internos – Si la arquitectura de red no esta segmentada los atacantes pueden hacer un mapa de redes internas y ver si los puetos están open o closed en servers internos a partir de los resultados de conexión para conectar o rechazar las SSRF

EJ: Exposicion de datos sensibles: hackers pueden acceder a archivos locales para obtener info confidencial

