Seguridad Informática

Alumno: Santiago Vietto

Docente: Mariano Aliaga

DNI: 42654882

Institución: UCC

Año: 2021

Seguridad de la información

Definiciones de seguridad de la información

Diversos enfoques

_ Podemos tener diferentes enfoques a la hora de definir la seguridad de la información, no hay una definición única sino que hay distintos autores, distintas normas, o principios que la definen desde un determinado punto de vista o enfoque.

- Se puede considerar como un conjunto de medidas de protección.
- También se puede considerar como una serie de controles de accesos: es decir, mantener el control y protegerse de ataques.
- También como la preservación de la triada CID: confidencialidad, integridad, disponibilidad.
- O como la existencia de un estado: invulnerabilidad, protección.

Ejemplos de definiciones

Seguridad como conjunto de medidas: acá tomamos la definición de NIST Glossary (entidad que trabaja con seguridad de la información, define estándares, reporta vulnerabilidades y demás), esta nos dice que "La protección de la información y sistemas de información de acceso no autorizado, uso, difusión, disrupción, modificación o destrucción, a fines de proporcionar confidencialidad, disponibilidad e integridad". Es decir, todas aquellas medidas que vayan garantizando que la información sea confidencial, integral y disponible.

Seguridad como control de accesos: acá tomamos la definición de William R. Cheswick en donde afirma que "Hablando ampliamente, la seguridad es evitar que alguien haga cosas que no quieres que haga con o desde tu ordenador o alguno de sus periféricos".

Seguridad como preservación de CID: esta proviene de la ISO17799 (una de las normas que regula el tema de seguridad de la información en una organización) y expresa que "La seguridad de la información se puede caracterizar por la preservación de la confidencialidad, integridad y disponibilidad (CID)".

- Confidencialidad: capacidad de proporcionar acceso a usuarios autorizados, y negarlo a no autorizados. Tenemos algunos subcomponentes:
    - Identificación: saber quién es el sujeto o sistema que esta interactuando con nosotros. Uno de los mecanismos más comunes, baratos y frecuentes de identificación, es un nombre de usuario.
    - Autenticación: forma de asegurarnos que ese sujeto sea quien realmente dice ser y no alguien que se está haciendo pasar por él. El mecanismo que tenemos

acá es la contraseña o password, que es lo que nos va a autenticar o decir que el usuario que quiere interactuar con nosotros es autentico.

- Autorización: ahora que sabemos quién es el sujeto que esta interactuando con nosotros, podemos definir a que le dejo acceder y a que no. Mediante distintos niveles de permisos podemos autorizarlo a acceder a determinadas partes y no a otras, o acceder a un determinado archivo con lectura o lectura y escritura, etc.
- Auditoria: desde el punto de vista de la seguridad siempre tenemos que poder ver que paso. Auditar y analizar para atrás, como se dio una serie de hechos. La forma típica de esto, a modo de mecanismo, relacionado con el ejemplo que venimos viendo, es un log o registros, que mediante estos se registra todo lo que va ocurriendo.
- No repudio: es que, aquel que hizo algo, no pueda aducir que no fue el quien lo hizo porque se estaría desligando de la responsabilidad. Para el ejemplo de un usuario, esta forma de materializa de algún modo mediante políticas.

- Integridad: es la capacidad de garantizar que una información o mensaje no han sido manipulados o alterados y también de que los servicios procesen correctamente la información y no de forma errónea.
- Disponibilidad: capacidad de acceder a información o utilizar un servicio siempre que lo necesitemos, es decir, que la información esté disponible.

_ Se habla de una triada, porque los tres son necesarios y dependientes unos de otros. De nada me sirve que la información que tenga sea totalmente confidencial y se respeten todos estos parámetros que estamos mencionando si esa información se guarda en un disco por ejemplo que tiene sectores defectuosos y se corrompe a la ahora de grabarse o a la hora de la transmisión alguien puede modificar la información. De nada sirve tampoco que la información sea confidencial, integra y que no esté disponible. Es por eso que se necesitan de las tres componentes.

<u>Seguridad como existencia de un estado:</u> acá tomamos la definición de Gene Spafford donde afirma que “El único sistema verdaderamente seguro es aquel que se encuentra apagado, encerrado en una caja fuerte de titanio, enterrado en un bloque de hormigón, rodeado de gas nervioso y vigilado por guardias armados y muy bien pagados. Incluso entonces, yo no apostaría mi vida por ello”. Es muy difícil lograr un nivel óptimo de seguridad, y lo que vamos a ir buscando es lograr la mayor seguridad posible pero no necesariamente vamos a lograrlo siempre.

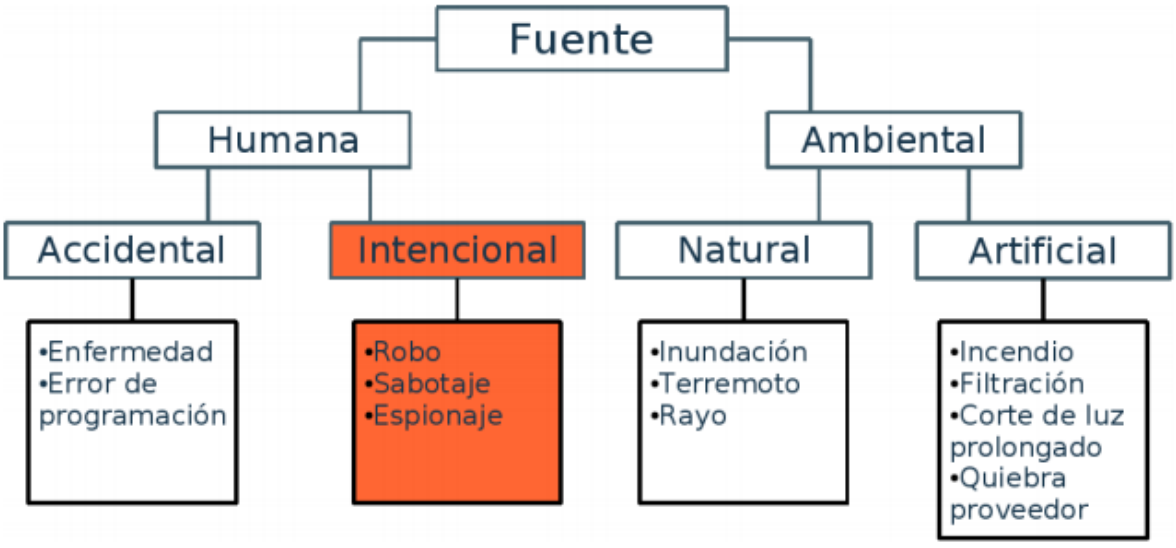
Conceptos y terminología

Activos: es decir que se protege o aquello que protegemos. Pueden ser recursos del sistema de información o relacionado con este, necesarios para que la organización funcione correctamente y alcance los objetivos propuestos.

Amenaza: es decir de que se protege o de que protegemos los activos. Es un evento que puede desencadenar un incidente en la organización, produciendo daños materiales o perdidas inmateriales en sus activos.

_ Cuando vemos las fuentes de amenazas, podemos hablar de que sean:

- Humanas: que a su vez pueden ser accidental, por ejemplo, una enfermedad, un error de programación o del operador, etc, por eso es que por ejemplo debemos tener la buena práctica de no trabajar como el usuario root o administrador en un sistema operativo, porque si yo me equivoco teniendo el usuario que tiene todos los privilegios, puedo causar un gran daño que estaría mitigado si estuviera trabajando con un usuario de menor privilegio, por eso se recomienda que siempre debemos trabajar con los privilegios mínimos, y solamente cuando necesitas algún privilegio específico para realizar alguna tarea se accede como root. Y también puede ser intencional, por parte de un ciber-criminal, o alguien que quiera afectar la organización en general y no solo la parte de la información, por ejemplo robo, sabotaje, espionaje, etc.
- Ambientales: que a su vez pueden ser natural, es decir, dependiendo del lugar geográfico donde nos ubiquemos debemos tener en cuenta una amenaza ambiental como terremotos, inundaciones, rayos, etc, donde todo esto puede afectar la disponibilidad de la información. Y ademas puede ser artificial, por ejemplo incendio, corte de luz, quiebra de un proveedor, filtración de agua, etc, donde dependiendo de la criticidad del recurso uno tiene que tener en cuenta esto.



_ Como vemos el panorama de amenazas es muy amplio y de nada sirve poner todas las balas apuntadas a una fuente de amenaza humana intencional si no pensamos en las demás.

Vulnerabilidad: es una debilidad latente que influye negativamente en un activo y que posibilita la materialización de una amenaza. Podemos pensar en vulnerabilidad como una puerta cerrada sin llave, en donde de por si no es que alguien se metió, pero está ahí latente, y si alguien prueba, puede abrir la puerta y acceder. Son fallas o debilidades que están latentes y que alguien puede encontrar y utilizar para determinadas acciones.

Exposición: valga la redundancia es una exposición innecesaria de información que puede utilizarse para llevar a cabo un ataque.

_ Por ejemplo en Linux usando el comando *nc* (net cat) que es sinónimo de *telnet* en Windows, que lo que hace es establecer una conexión en un puerto TCP, y a continuación de este le ponemos un servidor de destino por ejemplo www.cfg.com y a continuación separado por un espacio ponemos el puerto al que nos vamos a conectar que por default el puerto HTTP es 80. Entonces con *nc www.cfg.com 80* simulamos una conexión a esa página en el puerto 80. Si lo ejecutamos, puesto de que no nos rechaza el login quiere decir que estamos conectados, (en los sistemas Unix o Linux, si algo anda bien no dice nada, de lo contrario muestra un mensaje). Luego si colocamos por ejemplo GET / HTTP /1.1 nos devuelve información que no es necesario que la muestre, donde no quiere decir que sea vulnerable, pero estamos viendo un montón de cosa que no deberíamos saber y ahora esto no solo nos sirve para saber cómo encarar sino para saber que no hacer.

Exploit: es una pieza o porción de software, código, un grupo de datos o una secuencia de comandos que aprovechan una falla o vulnerabilidad para causar un comportamiento no deseado o imprevisto en sistemas de software o hardware, como por ejemplo un acceso, daño a la información, denegación de un servicio en un sistema de software o hardware. En las bases de datos de vulnerabilidades también tenemos los exploit.

Ataque: evento, exitoso o no, que atenta sobre el buen funcionamiento del sistema. Ataques tenemos permanentemente pero eso no significan que vayan a ser exitosos. Por ejemplo un keylogger.

Incidente (ataque exitoso): acceso, uso, divulgación, modificación o destrucción no autorizada de información; y además es un impedimento en la operación normal de las redes, sistemas o recursos informáticos.

Defensa: conjunto de productos, medidas y procesos que tenemos para a evitar la concreción de un incidente.

Riesgo: es un término probabilístico y hace referencia a la probabilidad de que suceda la amenaza o evento no deseado.

Impacto: poder medir la consecuencia al materializarse una amenaza.

Dificultades de la seguridad de la información

_ Por un lado no existe un sistema que sea 100% seguro, lo que se trata es hacerlo lo menos probable posible a las fallas o incidentes de seguridad. Debemos buscar un equilibrio entre la seguridad y la usabilidad, o sea hacer algo seguro pero que sea usable. Otro problema es el del eslabón más débil, ya que debemos saber encontrar y ubicar cual es el eslabón más débil. Una cadena es tan débil como su eslabón más fuerte, por ende debemos estar atentos a no introducir eslabones débiles o identificarlos para prevenirlos de alguna forma. La seguridad no es un problema únicamente tecnológico, ya que sería muy fácil si lo fuera. Pero este es un tema en el que entran en juego factores de tecnología, factores humanos, factores ambientales, o de varios tipos, que impiden que uno pueda reducirlo a un tema tecnológico. Finalmente, otra dificultad es que la seguridad de la información debe ser un proceso, no es un producto o algo que se hace una única vez y se llega al estado de seguridad, sino que es un proceso permanente que hay que ir teniendo en cuenta.

Virtudes de la seguridad de la información

Consideración diaria

_ Como es un proceso diario, lo que NO hay que hacer:

1. Hacer algo sin pensar en la seguridad.
2. Sufrir un incidente de seguridad.
3. Descubrir que lo que se hizo en el paso 1 introdujo una vulnerabilidad que permitió la ocurrencia del paso 2.
4. Asegurar la organización contra el ataque específico del paso 2.
5. Esperar que pase el tiempo.
6. Tener un incidente de nuevo.
7. Darse cuenta de que mientras se esperaba en el paso 5, otro nuevo ataque se llevó a cabo relacionado a lo que se hizo en el paso 1.

_ Ahora Lo que SI hay que hacer:

1. Pensar en la seguridad desde un comienzo.
2. Hacer algo (mientras se sigue pensando en seguridad).
3. Continuar pensando en seguridad.

_ Podemos evitar la vasta mayoría de incidentes de seguridad haciendo de esta una consideración diaria. Entonces como practicas recomendadas para lo que es la recomendación diaria:

- Hacer de la seguridad un pensamiento continuo.
- Promocionar en otros este pensamiento continuo.
- Incluir formalmente la seguridad en todo nuevo proyecto.

- Incluir formalmente la seguridad en toda nueva implementación.

Esfuerzo comunitario

_ Acá hablamos de que tenemos dos comunidades:

Comunidad de seguridad interna: que es propia de la organización, es decir, del borde de la organización conectada a internet hacia adentro. En esta comunidad debemos saber que la seguridad no puede ser lograda por profesionales de la seguridad solos, sino que es un sistema compuesto por personas, software, hardware, procesos, por recursos, donde cada uno con su rol debe tener en cuenta principios de la seguridad. Y además, nuestro rol debe ser el de integrar a los usuarios finales en las prácticas de seguridad locales.

Comunidad de seguridad externa: o sea lo que está de la organización para afuera. Esta es muy importante porque organizaciones con buenas prácticas de seguridad pueden verse afectadas por la falta de seguridad o las malas prácticas de otras. La mayoría de los ataques a sistemas “no importantes” se utilizan para lanzar ataques a otras organizaciones. Y nuestro rol es mantenernos seguros para que los otros estén seguros de nosotros, es decir, no solo buscamos que en nuestra organización no se produzcan incidentes o ataques exitosos sino que también debo asegurar que desde nuestra organización no se realicen ataques hacia los otros.

_ Las buenas prácticas de seguridad requieren un esfuerzo comunitario donde todos cumplan su parte para proteger sus propios sistemas. Entonces como practicas recomendadas, por un lado:

- Mantenerse informado de las distintas temáticas y ataques.
- Informar a los otros.
- Mantenerse actualizado.
- Informar a los usuarios finales.
- Tomar decisiones en equipo, donde participan las distintas áreas de la organización.

Foco generalizado

_ En el mundo de la SI (seguridad de la información), hay miles de vulnerabilidades explotables por decenas de miles de ataques con virtualmente millones de permutaciones posibles. Entonces la seguridad es muy dinámica, y requiere métodos elevados de pensamiento. Por ejemplo un firewall que es un dispositivo que puede ser software o hardware, y es la puerta de entrada o salida de una organización desde y hacia internet. El firewall por default se configura en una configuración que se llama diodo permitiendo todo el tráfico saliente desde la organización hacia cualquier lado pero el entrante esta todo bloqueado por default, aunque podemos configurar permisos de tráfico que sabemos que vamos a usar. Lo que nos protege del grueso, es esta política de mirar desde alto nivel de decir vamos a denegar todo menos lo que vamos a usar.

_ Entonces como practicas recomendadas tenemos:

- Aprender y compartir los conceptos detrás de las virtudes y reglas de la SI.
- Pensar en términos de la “vista panorámica”.
- Definir documentos escritos con políticas de alto nivel.

Educación para todos

_ Es decir, si la seguridad va a ser un trabajo diario, un esfuerzo comunitario, y considerado en todo, entonces todos deben estar envueltos en algún grado en las prácticas de seguridad. Por lo tanto, todos los implicados deben tener algún nivel de educación en seguridad. Esto es sumamente importante como profesionales de la seguridad porque normalmente el usuario final puede o ser nuestro peor enemigo o nuestro aliado, donde el más afectado es el usuario final. Podemos transformar al usuario final de ser un riesgo de seguridad, a un colaborador en mantener la seguridad del entorno. No existe una solución “tecnológica” que no pueda ser deshecha por un grupo de usuarios finales no entrenados, no informados o no cooperativos.

_ Entonces dentro de esta educación y capacitación que debemos hacer periódicamente para los distintos usuarios de la organización, como practicas recomendadas tenemos:

- Buenas prácticas de instalación de software.
- Buena práctica de reconocimiento de eventos “sospechosos”.
- Buenas prácticas de navegación, por ejemplo que sitios son seguros o no.
- Buenas prácticas de confidencialidad, por ejemplo como gestionar contraseñas.
- Explicar conceptos de seguridad a empleados en forma permanente

Reglas de la seguridad de la información

Regla del menor privilegio

_ Permita solo el acceso requerido para hacer el trabajo, y nada más. Es la única forma en que podemos estar seguros de que sabemos qui en tiene acceso a que, y por qué. Acá podemos identificar distintos componentes:

Sujeto: la persona, lugar o cosa que obtiene el acceso. (Quien?).

Objeto: la persona, lugar o cosa a la cual el sujeto obtiene acceso. (A qué?).

Acceso: el nivel o grado de acceso dado al sujeto. (Como?).

Contexto: la situación o circunstancias que rodean al acceso. (Cuando? Por qué?).

_ Con esto vamos restringiendo el acceso y dándole al sujeto todo lo que necesita para hacer el trabajo y nada más. Este es un principio fundamental de la seguridad. Por default primero deniego todo y después permito las cosas puntuales que sé que el sujeto va a necesitar para hacer su trabajo.

_ Entonces como practicas recomendadas tenemos:

- Crear todas las políticas de seguridad desde el punto de vista del mínimo privilegio.
- Siempre comenzar por negar todo.
- Evaluar si el sujeto realmente necesita acceso al objeto y bajo que circunstancias.

Regla de los cambios

_ Los cambios deben ser administrados, coordinados y deben considerarse las posibles implicancias de seguridad. Es decir en toda organización siempre nos enfrentamos a situaciones dinámicas o cambios permanentes que van a ir ocurriendo pero que de algún modo debemos tenerlos gestionados y administrados.

- Los cambios solo deberían ocurrir después de que se ha probado que sean seguros. Generalmente se establece un entorno o forma de probarlo.
- Los cambios deberían ser consistentes y no introducir demasiada diversidad. Ya que mientras más diversos sean más difícil va a ser tener en cuenta todo y poder tomar políticas generales.
- Solo deberían poder realizar cambios quienes estén calificados para ello (aplicación del mínimo privilegio).

_ Entonces como practicas recomendadas tenemos:

- Implementar el control de cambios.
- Controlar cambios de seguridad.
- No implementar un producto hasta que haya sido correctamente probado.
- Estandarizar las tecnologías a utilizar.

Regla de la confianza

_ Un buen practicante de la seguridad es alguien que es amigo de todos, pero que en realidad no confía en nadie. Y en esto decimos que:

- Aplicar el principio de mínimo privilegio: solo confiar en aquello que es necesario.
- Aplicar distintos “niveles de confianza” según sea necesario. No por una cuestión humana sino por una cuestión estrictamente funcional de un principio que hay que aplicar para que el sistema vaya siendo lo más seguro posible.

_ Entonces como practicas recomendadas tenemos:

- Recordar que cualquiera puede ser el enemigo, incluso uno mismo, por ejemplo el caso de root.
- No confiar en nada que este fuera de nuestro control.
- Tener en cuenta de las derivaciones de confiar en algo o alguien.
- Crear políticas globales que vayan más allá de los niveles de confianza.

Regla del eslabón más débil

_ Una cadena es tan fuerte como su eslabón más débil, y una práctica de seguridad es solo tan fuerte como su control más débil. Algunos eslabones débiles que podemos tener en una organización son:

- Instalaciones por defecto (configuraciones, contraseñas, puertos).
- Malas contraseñas.
- Módems activos en equipos de la red.
- Falta de monitoreo y control de logs mediante un sistema.
- Servidores o equipos temporales.
- Backups descuidados o no probados.
- Aplicaciones no autorizadas.
- Antivirus desactualizado.

_ Entonces como practicas recomendadas tenemos:

- Buscar el “eslabón más débil” en forma continua.
- Documentar donde existen debilidades de seguridad.
- Evitar la introducción de nuevos eslabones débiles.

Regla de la separación

_ Para asegurar algo, esto debe estar separado de los peligros y amenazas del mundo que lo rodea. Esto habla de que las fortalezas y debilidades de un objeto están normalmente relacionadas con las tareas que dicho objeto lleva a cabo. Y cada servicio tiene su propia “sensibilidad” a la seguridad. Hay que tener cuidado al combinarlos.

_ Entonces como practicas recomendadas tenemos:

- Aislar servicios y datos importantes.
- Aislar servicios que son más propensos a ataques.
- Aislar todos los “servicios de seguridad”.
- Solo agrupar servicios basados en factores comunes de seguridad.

Regla del proceso de tres etapas

_ Toda medida de seguridad debe ser pensada como un proceso de tres etapas, y a lo que va esto es que en todas las etapas de un proceso debe estar incluida la seguridad, no solo al momento de comenzar con la implementación, sino posterior, monitorear que todo siga en orden y mantenimiento para que no se produzcan nuevas fallas o problemas de seguridad:

Implementación: realizamos un análisis, diseñamos una solución, adquirimos las herramientas, lo construimos, lo probamos y ponemos en producción.

Monitoreo: no existe la “seguridad completamente automatizada”.

Mantenimiento: cualquier dispositivo de seguridad sin actualizaciones por un tiempo prolongado fallara en reconocer nuevos ataques.

_ Entonces como practicas recomendadas tenemos:

- Considerar la regla del proceso de tres etapas desde un comienzo.
- Asegurarse de que existan controles de registros y monitoreo.
- Mantenerse actualizado.

Regla de la acción preventiva

_ Esta habla de que la seguridad solo puede ser exitosa si se enfrenta con un enfoque proactivo. De nada sirve aplicar una serie de medidas y procesos en un determinado momento o dejarlo y esperar que ocurra algún problema, por eso permanentemente lo que busca son puntos débiles o que vulnerabilidades puede producir este cambio, etc, por ende estar permanentemente atento.

- Es una tendencia humana el tener respuestas “reactivas” en la mayoría de las situaciones.
- Muchas veces se consideran las medidas proactivas una pérdida de tiempo o distracción.
- La proactividad “no vende”, ya que es difícil de convencer.

_ Entonces como practicas recomendadas tenemos:

- Mantenerse informado de los temas de seguridad actuales.
- Realizar pruebas periódicas en dispositivos de seguridad.
- No quedarse solo con los problemas más comunes.

Regla de la respuesta inmediata y apropiada

_ Se debe contar con un plan organizado de cómo responder a un ataque, analizar los riesgos pendientes y planificar los pasos futuros. Tiene que estar pensado un plan para actuar dado un problema de seguridad.

- Los pasos que demos luego de un ataque son tan importantes como los que dimos para evitarlo.
- Se debe estar preparado con anterioridad para reaccionar correctamente.

_ Entonces como practicas recomendadas tenemos:

- Desarrollar un buen plan de respuesta de incidentes.
- Tener una “cadena de mando” bien definida para estas circunstancias.
- Reaccionar rápidamente.
- Hacer un seguimiento del incidente.

Hackers

Conceptos y tipificaciones

Hacker: es una persona que disfruta de un conocimiento profundo del funcionamiento interno de un sistema (hardware y software), en particular de computadoras y redes informáticas, capaz de forzar sus capacidades al máximo. Es una persona muy curiosa que investiga y aprende permanentemente, normalmente por su cuenta, muchas veces es autodidacta. Este concepto también se aplica a un programador experto o entusiasta de la seguridad, por ejemplo a los que desarrollan kernel en Linux se los denomina Kernel-Hackers, y no es que sean delincuentes sino que son programadores super talentosos y expertos que utilizan sus habilidades para hacer algo que les sirva a todos.

_ Algunas clasificaciones de hacker en base al contexto en el que se utiliza el termino:

- White Hat: se refiere a un profesional en la seguridad de “ethical hacker” o “penetration tester” que se enfoca en asegurar y proteger sistemas de IT de ataques.
- Gray Hat: es un término medio, ya que es un hacker que actúa ilegalmente, algunas veces con buenas intenciones, y otras no. Un ejemplo de este es que hay quienes encuentran una vulnerabilidad y las reportan para que eso sea solucionado y no tengan un incidente de seguridad, pero al ser ilegal, ya que no tiene el consentimiento de la persona u organización como el de un white hat hacker que es contratado, muchas veces lleva a la persecución de la persona que lo reporta por más que sus intenciones sean buenas.
- Black Hat: un “cracker”, que ataca redes, equipos o sistemas, o realiza actividades con fines criminales. También son creadores de software maligno (virus, troyanos, etc) y quienes rompen las protecciones de sistemas de seguridad para cuestiones de copy right por ejemplo.
- Blue Hat: alguien externo a una firma de desarrollo de software que es contratado para detectar fallas de seguridad en sus productos.

Cracker: este término encaja con la idea que por lo general se tiene de un ciber-delincuente o la visión negativa de un hacker. Un cracker es una persona dedicada a romper o vulnerar las protecciones de un sistema. Generalmente ingresa por la fuerza y en forma oculta o no autorizada, obteniendo accesos del tipo administrativo o un rol que le permita hacer más cosas para luego poder ver información, o dañarla, o modificar algo con distintos fines.

Phreaker: viene de ”phone freak”, este término ya no se utiliza mucho pero estuvo en moda en los comienzos del desarrollo de software más sistemático donde se usaban las redes telefónicas como medio para la comunicación entre los distintos puntos de las redes. Es quien rompe o vulneraba las protecciones de redes telefónicas, por ejemplo para

hacer llamadas internacionales gratuitas o engañar gente para que haga consumo de telefonía.

Historia y ejemplos

_ El término “hacker” surge en los años ’60 en el MIT (Massachusetts Institute of Technology), donde a los ingenieros se les empieza a aplicar este término cuando empiezan a aplicar la curiosidad, capacidad y conocimientos profundos de cómo funciona todo el sistema en cuanto a hardware y software, donde estos son los primeros creadores de cosas más artísticas como juegos y música en computadores (con tarjetas perforadas), y estas eran hazañas. En los años ’70 estas personas hicieron posible el concepto de “una computadora en cada hogar”. Luego, la Internet de hoy fue posible por los desarrollos y filosofía de estos primeros hackers, que crearon cosas como el protocolo TCP/IP, WWW, HTML, etc, todo lo que hoy en día usamos de forma transparente.

_ Algunos de los históricos hackers:

Jon Postel: es uno de los padres de Internet. Formo parte del grupo que unió las dos primeras computadoras de Internet, en 1969. Fue director durante casi 30 años de la Internet Assigned Numbers Authority (IANA), que asigna las direcciones IP y controla los servidores raíz del sistema de nombres de dominios (DNS). Murió en 1998 a los 55 años. Cualquier domino que exista se debería cargar desde su ordenador personal allí en el IANA.

Ken Thompson: al comienzo de los ’70 desarrollo junto a Dennis Ritchie el sistema operativo UNIX en los Laboratorios Bell. Trabajo en el desarrollo de múltiples herramientas para UNIX en equipos PDP11. Desarrollo en 1992 el formato de codificación de caracteres UTF-8. Desde 2006 trabaja en Google y co-invento el lenguaje de programación Go.

Steve Wozniak: es el gurú de los ”hardware hackers” porque se aboco más a la parte del hardware. Junto a John Draper desarrollo las primeras “blue boxes”. En 1977, junto a Steve Jobs, pusieron a la venta el primer Apple. Un año después, las ventas se habían multiplicado por diez. Fundo la Electronic Frontier Foundation (EFF), que es una ONG para defender a los hackers. Hoy es profesor de informática y filántropo.

Margaret Hamilton: ella lidero el desarrollo del software de vuelo del módulo lunar en la misión Apollo 11. Acuño el término “ingeniería de software” para diferenciarlo del hardware y otras ingenierías. Reporto y documento un bug en el “programa de prelanzamiento”, que evito un accidente en la misión Apollo 8.

Vinton Cerf: es uno de los padres de Internet. A principios de los 70 creo, junto a Roberth Kahn, el protocolo para la comunicación de paquetes Transfer Control Protocol/Internet Protocol (TCP/IP). En 1992 co-fundo la Internet Society y fue su primer presidente.

Actualmente preside la Internet Corporation for Assigned Names and Numbers (ICANN) y es “Chief Internet Evangelist” de Google.

John Drapper (Captain Crunch): el inventa la primera “blue box” (dispositivo electrónico que emitía un sonido en la frecuencia de un silbato de juguete que traía como sorpresa una caja de cereales). Es admirado por sus acciones legendarias. Fue una pesadilla para las compañías telefónicas norteamericanas, es detenido por fraude telefónico en varias ocasiones. Hoy trabaja como desarrollador de programas de seguridad.

Tim Berners-Lee: nacido en Londres en 1955. Cuando estudiaba en la Universidad de Oxford, construyo un ordenador a partir de un televisor viejo. En 1980 propuso un proyecto basado en el hipertexto, lo que posibilito luego la World Wide Web, que presento oficialmente en 1989, junto a un navegador, un editor y un servidor web. Entonces creo el concepto de navegador, o sea, un software al que se le pueda indicar la URL o URI de un recurso, y eso hace que se envié una petición a un servidor que va a tener los archivos ya sea HTML o imágenes, etc, y a través de este protocolo basado en el hipertexto, lograr unir ese cliente con ese servidor, entre otras cosas. Actualmente es director del World Wide Web Consortium.

Richard Stallman: fundador del movimiento del “software” libre y la Free Software Foundation, en 1984. Definió la idea del “software libre” y creo una licencia (GPL) que especificaba que cualquiera podía usar, estudiar, distribuir y mejorar los programas libremente. Actualmente vive de las conferencias que da por todo el mundo.

Linus Torvalds: nacido en 1969 en Helsinki. Creador del sistema operativo Linux en 1991. Dirige el desarrollo del kernel. Actualmente trabaja en la Linux Foundation.

_ Casos resonantes a lo largo de la historia:

- 1971: “Phone phreaking”. John Draper inventa la “blue box”, logrando realizar llamadas de larga distancia sin cargo.
- 1988: Robert Morris Jr. crea un gusano auto-replicable que se distribuye rápidamente a través de ARPAnet e infecta más de 60000 equipos UNIX.
- 1989: Kevin Mitnick roba software de DEC y es la primera persona encarcelada por acceder a redes y equipos con fines criminales.
- 1999: Jonathan James, un adolescente de 16 años, accede a la NASA y descarga codigo propietario de la Estación Espacial Internacional.
- 1999: Un grupo de crackers noruegos rompe la clave para descifrar la protección anti-copia de los DVD.
- 2001: se atacan los equipos que controlan el flujo de energía eléctrica del estado de California.
- 2008: Se detecta por primera vez el gusano Conficker. Se estimaban en 15 millones los equipos afectados.

- 2010: Anonymous lanza la operación “Avenge Assange”: Amazon, PayPal, MasterCard, Visa, BankAmerica, etc.

Ética hacker

_ En la comunidad hacker, donde hay maliciosos o no, hay como un código de conducta o como una filosofía que se aplica, y que se dio a llamar como la ética hacker, con algunos principios que respetan quienes se sienten parte de esto. Tenemos que:

1. El acceso a los ordenadores y a todo lo que te pueda enseñar alguna cosa sobre cómo funciona el mundo debe ser ilimitado y total. Se propicia que el acceso a la información sea libre y que uno pueda aprender de cualquier cosa pudiendo acceder sin restricciones a la misma.
2. Toda la información debería ser libre y que uno pueda aprender de cualquier tema o cosa pudiendo acceder sin restricciones a información.
3. No creas a la autoridad, donde se promueve la descentralización.
4. Los hackers deberían ser juzgados por su hacking, sin importar sus títulos, edad, raza o posición. Lo que se valora y promueve es una meritocracia, es decir, que el valor de cada profesional esta dado por lo que sabe o lo que puede hacer y no tanto por lo que es el.
5. Puedes crear arte y belleza con un ordenador.
6. Los ordenadores pueden cambiar tu vida para mejor.

Malware

Definiciones y Clasificaciones

Malware

_ El termino Malware que proviene del inglés malicious software, es un software que tiene como objetivo infiltrarse en el sistema y/o dañar la computadora sin el conocimiento de su dueño. Hay muchos tipos de malware, ya que es un término muy genérico, por ende tenemos algunos como:

Adware: muestra o baja anuncios publicitarios que aparecen inesperadamente en el equipo.

Backdoor: es un software que permite el acceso al sistema sin el conocimiento del propietario y sin dejar rastros, ignorando los procedimientos normales de autenticación, y eliminando toda evidencia de su existencia. Es una “Puerta trasera por la cual acceder”.

Botnet: son robots de software o bots, que se encargan de realizar funciones rutinarias y operaciones masivas, que pueden ser desde enviar spam a atacar un objetivo. Normalmente están compuestos de equipos que llamamos zombi o infectados que

ejecutan un malware que se reporta en algún lugar donde se controla el botnet y desde ese punto el atacante puede tirar las tareas que debe realizar esa red de equipos.

Crackers: este término no solo se aplica a las personas, sino también a programas o software que descifran las contraseñas tanto del sistema operativo como de las aplicaciones y sistemas.

Dialers: programas que llaman a través del modem, y sin el consentimiento del usuario, a un número telefónico de larga distancia o de tarifas especiales para redituar beneficios económicos al creador del malware.

Exploit: software, conjunto de comandos, etc, que ataca una vulnerabilidad particular de una aplicación, servicio o sistema operativo, y así lograr una acceso o ruptura de lo que se quiere atacar.

Keylogger: programa espía que registra todas las pulsaciones del teclado para robar claves e información sensible del usuario. Puede ser hardware pero por lo general es software.

Pharming: es engañar al ordenador suplantando lo que es el servicio de DNS mediante el archivo hosts local, dirigiendo así al usuario a un sitio distinto del que cree estar abriendo, y así engaña al ordenador.

Phishing: consiste en obtener información confidencial engañando al usuario con paginas o correos que se hacen pasar por entidades a las que normalmente uno accede o utiliza, para cargar un malware por ejemplo, y cuanto más convincente son mayor éxito tienen. Básicamente es mandar un mensaje al usuario que lo preocupe, que lo haga abrir un archivo o ir a un enlace y así poder ser engañado.

Ransomware: cifra los archivos del usuario con una determinada clave, que solo el creador del ransomware conoce y pide al usuario que la reclame a cambio de un pago. Este utiliza técnicas modernas de criptografía, reconoce los archivos útiles de las personas y los cifra con una llave que para obtenerla hay que pagar. No se recomienda pagar el rescate y hacer mucho backup ya que al atacante no le importa mucho que recuperemos la información.

Rootkit: programas que se agregan a un equipo luego de haber ganado el control del mismo. Suponiendo que a través de una vulnerabilidad logramos el acceso, instalamos un rootkit para permitir accesos futuros sin ser detectados y sin que queden rastros. Generalmente borran los rastros del ataque y ocultan los procesos del atacante.

Spam: es correo electrónico masivo no deseado con fines publicitarios.

Spyware: son aplicaciones que recopilan información del sistema en el que se encuentran instaladas para luego enviarla a través de Internet. Al fin y al cabo cualquier red social es un gran spyware, pero donde uno es consciente y acepta para poder utilizarlo.

Troyano: es un programa o conjunto de programas relacionados que bajo una apariencia inofensiva se ejecuta de manera oculta en el sistema y permite el acceso remoto de un malware o usuario no autorizado al sistema. Este viene bajo el concepto de caballo de Troya, donde por fuera es algo y por dentro otra.

Virus: software malicioso que se adosa a un ejecutable, macro, correo, etc. Tiene la capacidad de auto-replicarse infectando otros archivos, y generalmente se utiliza como un vector para transferir troyanos o backdoors. En fin este se adosa a archivos y se replica a través de estos.

Worm (gusano): es un tipo de virus, pero que se replica de un sistema a otro en forma automática a través de la red. Este se adosa a la red.

Virus y Gusanos

_ La diferencia entre ambos es que el virus se adosa a archivos y se replica a través de estos y un gusano se replica automáticamente a través de la red. Mas en detalle, los virus afectan a los siguientes:

- Archivos
- Macros, por ejemplo una hoja Excel o documento que contiene macros.
- Librerías
- Sectores del disco, típicamente se apunta a los vectores de arranque.
- Archivos de scripts (.BAT), entre otros.
- Código fuente

_ De acuerdo a como infectan los virus podemos clasificarlos en:

Virus polimórficos: cifran, mutan o cambian su código o características de distintas maneras con cada infección para evadir su detección por los antivirus. Si van mutando dificultan la tarea de los antivirus.

Virus furtivos: ocultan las características normales de un virus para evitar ser detectados.

Virus multiparte: se dividen en varias partes para no ser detectados.

Virus de “cavidad”: aprovechan los espacios vacíos en algunos archivos.

Virus de túnel: se envían a través de protocolos alternativos o canales encubiertos.

Troyanos y Backdoors

Objetivos de un troyano:

- Robo o eliminación de información.
- Provocar caídas o demoras en los sistemas.

- Realizar ataques distribuidos de denegación de servicio DDoS. Afectar un montón de equipos que puedan ser controlados desde un punto único y de allí lanzar un ataque masivo.
- Anunciar la infección de un equipo para que pueda ser utilizado.

Métodos de comunicación: tenemos dos tipos de canales:

- Overt channels (canales descubiertos): el modo "normal" como los programas se comunican con computadoras o redes. Utilizando los protocolos estándares logramos sortear las protecciones y así enviar y comunicar el malware.
- Covert channels (canales encubiertos): usa medios de comunicación de formas para las que no fueron pensados. Estos son los que más existo tienen. Por ejemplo un avión es un medio de transporte aéreo, pero también puede ser usado como un misil, entonces este ejemplo es una forma de utilizar una herramienta para algo para lo cual no fue pensada.
- Conexión reversa: normalmente una red tiene un dispositivo que se llama firewall donde tenemos por un lado la red interna y por el otro internet, donde el firewall normalmente bloquea todos los accesos externos y permite los accesos de adentro hacia afuera. Permite el acceso externo a una red interna generando conexiones de adentro hacia afuera. O sea un método de conexión reversa es afectar de algún modo un equipo de la red que inicie el una conexión hacia afuera y le permite de ese modo al atacante entrar a la red.

_ De acuerdo a como infectan los troyanos podemos llamarlos:

Troyanos de acceso remoto (RATs): usados para obtener acceso al sistema de forma remota. Por ejemplo un backdoor envuelto en un troyano envuelto en un zip.

Troyanos que envían datos: buscan datos específicos en un sistema y los envían al atacante.

Troyanos destructivos: se utilizan para eliminar o corromper archivos en un sistema.

Troyanos de denegación de servicio: se usan para lanzar ataques de DoS.

Troyanos proxy: permiten lanzar ataques desde otro sistema. El proxy es una pasarela, o sea alguien que se pone al medio y que alguien pide un recurso el proxy va y lo trae y se lo entrega a quien lo pide, en donde este también controla que es lo que entrega, si entrega o no, etc. En este caso, el troyano que funcione como proxy es algo que permite al atacante conectarse a ese equipo y desde ahí lanzar el ataque, y de ese modo quien recibe el ataque lo ve como proveniente del proxy y no del atacante en sí que esta oculto.

Troyanos que deshabilitan software de seguridad: generalmente detienen o rompen el antivirus o firewall.

Ransomware

_ Este es el malware más dañino o más en moda en estos tiempos, y para la cual debemos estar siempre preparados.

Historia:

- AIDS (1989): un biólogo repartió 20.000 diskettes a los asistentes de la conferencia sobre el SIDA de la OMS. Los diskettes con el nombre de “Información sobre el SIDA – Diskettes de introducción”, tenían advertencias como “tu ordenador dejara de funcionar de manera normal”. Se cifraban todos los archivos y se pedían $189 como rescate.
- Archievus (2005): primer ransomware en usar criptografía asimétrica. Cifraba la carpeta ”Mis Documentos” y obligaba a hacer compras en una farmacia online para recuperar la información.
- Cryptolocker (2013): ingresa a través de archivos adjuntos de correos electrónicos, o del puerto remoto 3389 (RDP). Modifica las claves de registro y se copia a varias carpetas. Cifra solo ciertos archivos: Office, fotos, AutoCAD, etc. Utiliza la red Tor y Bitcoin para el rescate. Habría recaudado entre U$ 3 y 27 millones.
- Ransom32 (2016): el primero escrito en Javascript y uno de los primeros “ransomware as a service”, con la ventaja de ejecutarse en cualquier navegador. Es decir, cualquiera puede meter su dirección de Bitcoin e intentar infectar a la gente para ganar dinero.
- Petya y Mischa (2016): se instala en el sector de arranque del disco y cifra ”todo el disco”. Funciona uno u otro dependiendo de si se tiene permiso de administrador.
- WannaCry (2017): consiguió paralizar al servicio nacional de salud de Reino Unido, y a compañías como Telefónica, FedEx o Deutsche Bahn. Es decir, consiguió un efecto brutal en el mundo empresarial. Un investigador británico consiguió detener su expansión encontrando un dominio web en el código y registrándolo.

Herramientas de malware

Troyanos:

- Tini: es un pequeño backdoor que escucha en un puerto (7777/TCP) y brinda al atacante un prompt (línea) de comandos remoto para realizar distintas tareas.
- SubSeven: es un troyano de acceso remoto (RAT) que notifica al atacante cuando el sistema infectado se conecta a Internet, enviándole información sobre el sistema.
- BackOrifice 2000: herramienta de administración remota que un atacante puede utilizar para controlar un sistema a través de una conexión TCP/IP usando una interfaz GUI. Se divide en un server (victima) y un cliente (atacante), donde en el

servidor se configura como va a funcionar y que direcciones va a tener cargadas, y con el cliente luego me conecto a ese server para dar órdenes al servidor.

- BoSniffer: utilidad que intenta conectarse a un canal de chat IRC y anuncia su dirección IP para que pueda ser utilizado para futuros ataques.
- Beast: corre en la memoria asignada al servicio Winlogon.exe y se inserta en el ejecutable de Windows Explorer o Internet Explorer. Posee la funcionalidad de cliente, servidor y editor en la misma aplicación.
- Firekiller: deshabilita los programas de antivirus y firewall.
- Hard drive killer Pro: destruye toda la información de un disco de manera irrecuperable. Pisa muchas veces la información para que no pueda ser recuperada.
- Fport: reporta todos los puertos TCP y UDP que estén abiertos y cuál es el proceso correspondiente que están utilizando.
- TCPView: muestra información detallada de todas las conexiones TCP y UDP en el sistema.
- PrcView: un visor de procesos un poco más avanzado que el administrador de tareas de Windows que muestra información detallada de los procesos corriendo bajo Windows.
- Inzider: muestra los procesos con los puertos que cada uno utiliza.

Wrappers: programas que se utilizan para “envolver” un troyano en un software legítimo, es decir, Wrapper es la herramienta que se usa para crear un troyanos, donde se envuelve la utilidad valida con el malware, y lo que sale de eso sería un troyano. Tanto el troyano como el programa se combinan en un único ejecutable. Tenemos algunos como:

- Graffiti: es un juego animado que puede envolverse con un troyano.
- Silk Rope 2000: un wrapper que combina el BackOrifice con cualquier otra aplicación que se especifique.
- ELiTeWrap: un avanzado wrapper de .exe’s para Windows usado para instalar y correr programas.

La línea de comandos

Introducción

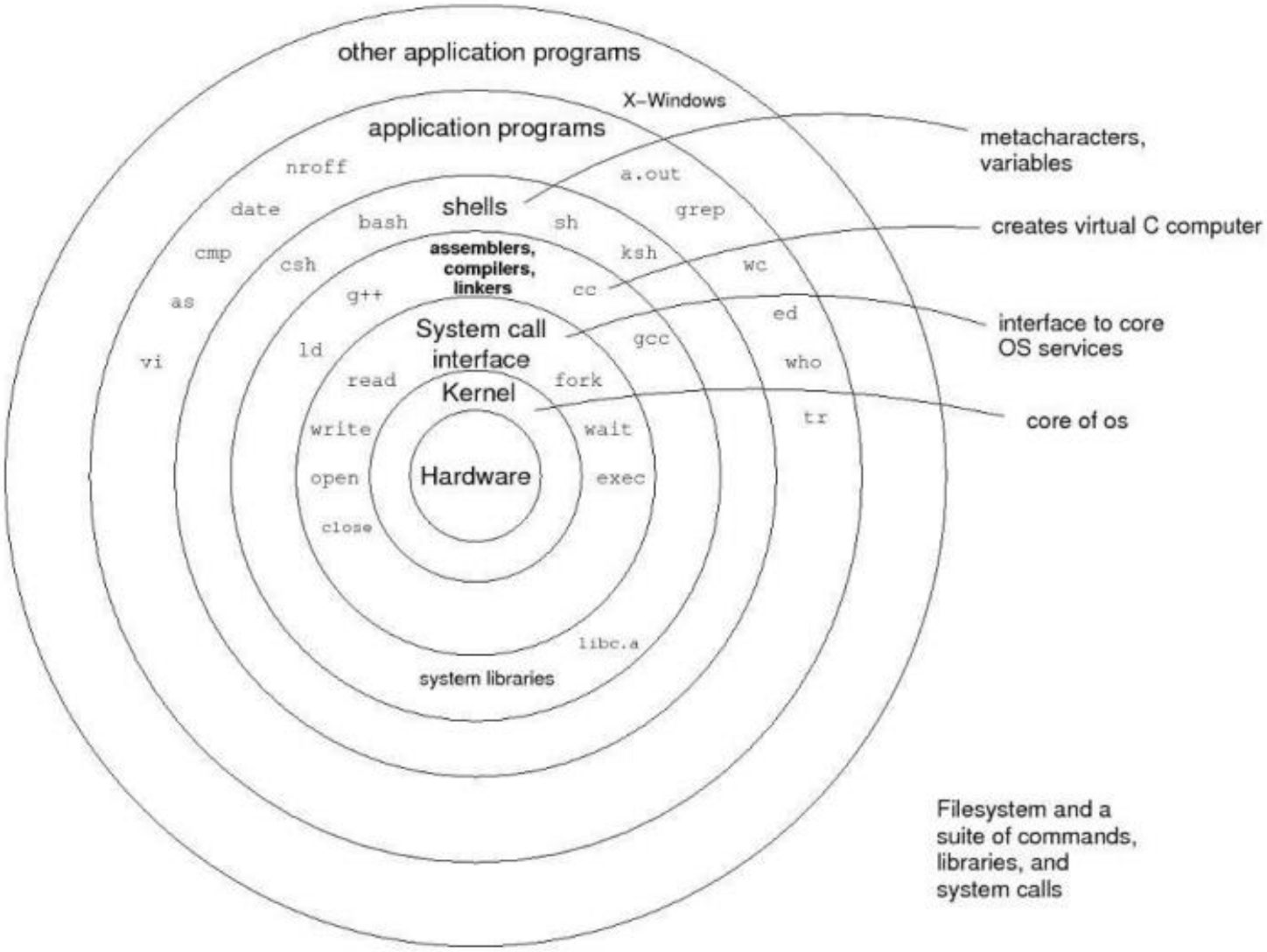
_ Los hackers no usan mouse. Con esto nos referimos a que cuando se hace un uso avanzado de los sistemas o de los dispositivos, muchas veces se termina reduciendo todo a comandos o un intérprete de comandos, donde uno puede ir gestionando distintas tareas. Y cuando uno quiere pasar lo más desapercibido posible, que en general es lo que busca un atacante, se trata de utilizar la interfaz más eficiente y más furtiva posible que como vemos es texto a través de la interfaz de comandos, y es mucho más fácil eso que lograr establecer una sesión de un escritorio remoto con todas las utilidades. El malware tiene que ser lo más pequeño e indetectable posible, entonces programar algo que utilice una interfaz de texto es mucho más simple y pequeño que hacer algo más sofisticado para manejar gráficos y demás.

Interfaz de Comandos en Línea (CLI): una interfaz de comandos en línea es un mecanismo para interactuar con un sistema operativo de computadora o software escribiendo comandos para realizar tareas específicas. Quien recibe e interpreta estos comandos es el Shell.

Shell (o Command-line Interpreter): es un programa que brinda todas las herramientas necesarias para recibir la entrada del usuario, procesar el comando que se envió, enviarlo al sistema operativo, mostrar el resultado y entre otras cosas más. Es decir, es el encargado de leer, recibir, analizar y ejecutar el comando solicitado, donde luego de completar el comando, muestra al usuario la salida correspondiente en forma de texto. Básicamente es la pantalla negra en el sistema operativo en la que colocamos los comandos. Hay distintos shells o programas que realizan esta tarea, dependiendo del sistema operativo y de las versiones.

_ En el siguiente diagrama vemos como se suele esquematizar un sistema Unix pero lo podemos pensar en general para cualquier sistema operativo, donde se esquematiza como una cebolla donde hay distintas capas que van cubriendo a las interiores y que van cumpliendo una función. En el núcleo tenemos el hardware propiamente dicho, y envolviendo el hardware tenemos el kernel que es el núcleo del sistema operativo encargado de hacer la interfaz entre el hardware crudo y todo lo que hay por arriba del sistema operativo. Arriba del kernel, los sistemas operativos implementan distintas llamadas de sistemas o sistems calls que son una serie de primitivas implementadas en el sistema operativo y que instruyen al kernel sobre qué acción realizar y luego el kernel realiza la acción sobre el hardware real. Mas arriba tenemos herramientas como compiladores, enlazadores o ensambladores, y dependiendo del lenguaje de programación que usemos vamos a necesitar una herramienta u otra. Por encima de estos tenemos los shells, y por una capa superior tenemos las distintas utilidades con programas

que hacen funciones específicas que luego son recibidas por el shell procesando su ejecución y su salida. Por último puede haber programas de más alto nivel como son las aplicaciones, sistemas gráficos, etc.



Conceptual Architecture of UNIX SYSTEMS

Operadores de redirección

_ Existen una serie de operadores de redirección que son comunes a distintos sistemas operativos y shells con los cuales podemos manipular los estándar input y output de los comandos y combinarlos para lograr cosas más complejas y potentes. A continuación tenemos los siguientes:

Operador de redirección (>): este envía la salida de un comando a un archivo. Es decir, en vez de que vaya al estándar output que es la pantalla, se redirige la salida a un archivo. Este comando pisa el archivo de destino con la salida del comando. El formato es comando > archivo. Por ejemplo, redirigimos el listado de archivos del comando *ls* en Linux a un archivo de texto:

ls > archivo.txt

_ Si luego hacemos *cat* archivo.txt podremos ver lo que hubiéramos visto en la pantalla. Si hacemos ls a un archivo inexistente podremos ver el mensaje de error, que si queremos lo podemos mandar al archivo.txt haciendo:

ls archivo123.txt

ls archivo123.txt > archivo.txt

Operador de redirección (>>): agrega la salida de un comando al final (append) del archivo indicado, sin pisar lo que había antes en el archivo, donde por ejemplo si el archivo contiene información, se agrega la salida al final. El formato es comando >> archivo. Por ejemplo, redirigimos el listado de archivos del comando *ls* en Linux a un archivo de texto y redirigimos al final la salida del comando *dmesg*:

ls dmesg >> archivo.txt

Operador de redirección input (<): usa los contenidos del archivo indicado como entrada para el comando. Con este cambiamos la entrada. El formato es comando < archivo.

Operador pipe (|): lo que hace es enviar la "salida" de un comando a la "entrada" del siguiente. Entonces podemos ir combinando distintos comandos que realizan distintas tareas haciendo que el segundo comando procese en base a lo que genero el primero. El formato seria comando1 | comando2. Por ejemplo, en Linux mostramos con *cat* el contenido de un archivo, y si lo combinamos con *grep* (comando para filtrar un determinado patrón de texto) filtramos para que nos muestre solamente las líneas que contengan el patrón systemd:

cat archivo.txt | *grep* systemd

Operador AND (&&): realiza una ejecución condicional, es decir, es una combinación condicional donde ejecuta el comando2 si solo si el comando1 se completó exitosamente. El formato seria comando1 && comando2.

Operador OR (||): el comando2 se ejecuta si solamente si el comando1 falla. El formato seria comando1 || comando2.

_ En todo comando que uno ejecuta, uno tiene una entrada, ese comando produce una salida y eventualmente puede llegar a producir errores. Entonces la entrada del comando es lo que llamamos estándar input, la salida es lo que llamamos estándar output, y la

parte de errores es lo que llamamos estándar error. Dependiendo de la tecnología, época y cadena de comandos que se procesen, puede variar la entrada, la salida y los errores. Hoy en día en una CLI la entrada es el teclado, donde el shell recibe del teclado las instrucciones y comandos del usuario, la salida típica es el monitor pantalla, y los errores coinciden con la salida en la pantalla.

Referencia comandos Windows

Shells MS Windows

_ A continuación vemos los Shells típicos de Windows, en donde dependiendo del Windows que sea es el que se utilice o puede haber varios:

COMMAND.COM: usado en los primeros Windows. Utilizado en MS-DOS y hasta Windows 9.x

cmd.exe: este se desarrolló con la aparición de Windows NT en adelante.

PowerShell: surgió como un intérprete de comandos y lenguaje de scripting basado en .NET. Se hizo portable y fue liberado en 2006, luego fue hecho open source y multiplataforma en 2016.

Manejo de archivos y directorios

_ De la sintaxis de los comandos, lo que está entre corchetes es opcional, o sea puede o no estar, y lo que está entre llaves normalmente es un OR, o sea o un comando o el otro, que puede estar o no.

_ Vemos que los comandos tienen distintos operadores que tienen un sentido que no hace falta saberlos de memoria pero si razonarlos ya que tienen una lógica y un nombre por algo. Y sino lo que podemos hacer es apelar a la ayuda en donde dependiendo del comando, si lo ponemos solo nos muestra información, y si no lo acompañamos con *-h* (help) en Linux o */?* en Windows, donde obtenemos una referencia rápida de que hace el comando, como se usa y que parámetros tiene.

assoc: viene de asociación y muestra o modifica las asociaciones de extensiones de archivos y el tipo. En Windows la extensión del archivo determina su tipo, entonces un archivo por ejemplo que termina en .exe es un archivo de tipo ejecutable, .zip es un archivo comprimido, .jpg es un archivo de imagen, etc. Cada tipo de archivo tiene un nombre, y con este comando asociamos la extensión con el tipo de archivo, por ejemplo a un jpg le ponemos archivo de tipo imagen, un .zip un archivo comprimido, etc. Si al comando lo ponemos solo, nos muestra todas las extensiones con los archivos que tenemos, sino podemos setear.

assoc [.ext [= [fileType]]]

_ La asociación entre la extensión y el tipo, en Unix o Linux esto no ocurre ya que el nombre no tiene nada que ver con lo que es el tipo de archivo, podemos tener por ejemplo archivo.exe y no es un ejecutable y es un archivo de texto, una imagen, etc.

attrib: muestra, establece o elimina distintos atributos de archivos y directorios. Este comando realiza lo mismo que hacemos cuando le damos click derecho a un archivo, entramos a propiedades y establecemos read only por ejemplo.

attrib [{+r|-r}] [{+a|-a}] [{+s|-s}] [{+h|-h}]
[[Drive:] [Path] FileName] [/s [/d]]

_ Por ejemplo con el operador *r* que quiere decir read only, agrega (+) o quita (-) el atributo de solo lectura a el archivo especifico. Con *a* de archive hacemos que sea de tipo archivador. El parámetro *s* que viene de system indica archivo del sistema. El *h* es de hiden u oculto. Si al comando lo ponemos solo, se nos muestran los operadores que tiene asociado.

ftype: muestra o define con que programa o aplicación se abre cada tipo de archivo. Con este asociamos el tipo con el comando correspondiente. O sea en un paso asociamos una extensión con un tipo, y en otro paso con otro comando podemos asociar ese tipo con un comando.

ftype [fileType [= [openCommandString]]]

_ Con *ftype* y *assoc* podemos asociar un tipo de archivo con un programa malicioso. Por ejemplo podemos asociar los archivos de WinRAR con el programa que abre un malware y seria:

ftype WinRAR = “C:\temp\malware.exe” “%1”

_ Ahora si ejecutamos lo siguiente podremos ver que lo modificamos, entonces cuando un usuario legítimamente vaya a querer abrir un Zip o un RAR, en vez de ejecutarse estos, se va a ejecutar el malware que cargamos.

ftype WinRAR

more: muestra la salida dividiéndola en pantallas. Cuando uno tiene que ver información que es muy larga y que se pasa de largo, podemos partirla en pantallas y de ese modo poder ver más información.

more [[/c] [/p] [/s] [/tn] [+n]] < [Drive:] [Path] FileName

_ Por ejemplo, el siguiente comando nos muestra la información por pantallas, donde pasamos de una pantalla a otra con el space y con enter pasamos por líneas.

dir c:/windows | *more*

type: muestra el contenido de uno o más archivos de texto. Hay distintas formas de referenciar los archivos, uno puede hacerlo con un path absoluto o relativo, donde absoluto es la ruta completa desde la unidad en donde está pero con este último podemos ejecutar cualquier archivo si no lo hacemos bien pero con esta podemos ejecutar cualquier archivo si no lo hacemos bien, y relativo seria posicionarnos en el directorio en el que está el archivo y colocamos el nombre del archivo mismo.

type [Drive:][Path] FileName

_ A modo de ejemplo mandamos la salida del listado de directorios a un archivo. Para ver el contenido de ese archivo aplicamos el comando *type*.

dir C:\Windows > C:\listado.txt

type C:\listado.txt

xcopy: copia archivos o directorios de forma avanzada. Tenemos muchos operadores. Podemos hacer copias recursivas de un directorio, o copias avanzadas como migrar los archivos de un servidor a otro de forma remota. Y además, copia atributos ocultos o atributos en cuanto a la propiedad y los permisos que están seteados en los archivos.

xcopy source [destination] [/A | /M] [/D [:date]] [/P] [/S [/E]] [/V] [/W] [/C] [/I] [/Q] [/F] [/L] [/G] [/H] [/R] [/T] [/U] [/K] [/N] [/O] [/X] [/Y] [/-Y] [/Z] [/EXCLUDE:file1 [+file2] [+file3] ...]

Estado y configuración de red

Dirección MAC: es la dirección física del adaptador de red del que estemos hablando, puede ser una placa de red alámbrica ethernet o inalámbrica wifi, o placa de fibra. Cualquier placa que utilice el protocolo ethernet tiene una dirección física grabada en el firmware o en el chip del dispositivo, y esta es la dirección MAC. Esta nos permite identificar de forma univoca a una placa de red.

Dirección IP: es una dirección lógica, modificable y que me identifica al dispositivo en una red determinada. Entonces en una placa de red que siempre va a tener la misma dirección MAC (física), podemos tener distintas direcciones IP según la red de que se trate.

Puerta de enlace predeterminada: es el equipo a través del cual llego a otras redes y a internet.

arp: muestra y modifica las entradas del cache. ARP (Address Resolution Protocol) es el protocolo que asocia una dirección MAC con una dirección IP. Este protocolo funciona en la capa 3 del modelo TCP/IP. Hace un mapeo entre la dirección física y la dirección IP, donde se produce un dialogo en el que el protocolo pregunta cuál es la dirección IP de tal determinada dirección MAC, en donde el equipo que tiene esa mac responde que tiene esa mac y la dirección IP es tal. Entonces la primera vez que se hace esa asociación, queda

guardada en una tabla como un cache del sistema operativo para que las sucesivas veces que tengamos que realizar esa comunicación no tenga que hacer de vuelta el dialogo para encontrar la correspondiente mac a una IP.

> *arp* [-a [InetAddr] [-N IfaceAddr]] [-g [InetAddr] [-d InetAddr [IfaceAddr]] [-s InetAddr EtherAddr [IfaceAddr]]

_ Con *arp -a* vemos la tabla ARP del sistema operativo en el momento, Para Windows y Linux es lo mismo, donde nos muestra por un lado la dirección IP y por el otro la dirección física expresada siempre en hexadecimal con 6 pares. Uno puede hacer esta asociación de forma estática, esto que por default es automático y se pregunta en la red, nosotros podemos asociarlo en forma estática y cambiar la tabla ARP del sistema operativo, a esto lo podemos hacer para asociar un dispositivo conocido del cual sabemos la mac, queremos darle determinada IP y no queremos que nunca cambie. Por ejemplo si interactuamos con algún equipo y listamos la tabla vemos que se agregó una entrada como dinámica porque se obtuvo sola utilizando el protocolo:

> *ping* 10.0.2.2
>
> *arp -a*

_ En una asociación a una IP, los datos dinámicos son cuando se obtiene automáticamente por el sistema operativo usando el protocolo y estático es si hacemos una asociación una IP a mano con *-s*, donde por ejemplo pasamos una IP y una dirección física:

> *arp -s* 10.0.2.3 ff-aa-bb-cc-11-22

ipconfig: es la configuración de los distintos adaptadores de red que tengamos. Muestra la configuración actual de TCP/IP y renueva el estado de DHCP y DNS. Es la configuración de los distintos adaptadores de red que tengamos. Podemos usar este comando para ver o setear parámetros de la placa de red

> *ipconfig* [/all] [/renew [Adapter]] [/release [Adapter]] [/flushdns] [/displaydns] [/registerdns] [/showclassid Adapter] [/setclassid Adapter [ClassID]]

_ Si al comando lo ponemos sin nada nos muestra información básica de los distintos adaptadores, por ejemplo la conexión DNS, dirección IP, Mascara de subred y puerta de enlace predeterminada. Podemos ver esta información en la configuración de red de Windows. Si al comando lo acompañamos con */all* nos muestra información mucho más detallada por ejemplo el driver o la descripción de la placa, la dirección MAC (física), servidores DHCP, etc.

_ Tenemos una tabla como un cache de DNS que es la asociación entre un nombre y una IP que le corresponde, y con *ipconfig /flushdns* borramos ese cache de IPs viejas para refrescarlo.

netstat: muestra información detallada de todo el stack de la red como el estado de las conexiones TCP/UDP, puertos que están escuchando, estadísticas Ethernet, ruteo IP, etc. Muestra el estado general de la red.

netstat [-a] [-e] [-n] [-o] [-p Protocol] [-r] [-s] [Interval]

_ Si al comando lo ponemos solo nos muestra las conexiones TCP que están establecidas (established). Si ponemos *-a* (all) nos muestra todas las conexiones no solo TCP sino también UDP, y no solo las establecidas sino también las que están escuchando o abiertas en UDP, y el *-n* (no resolve) en vez de mostrar el nombre nos muestra la IP. Con *-r* de route, podemos ver las rutas de la red. Con *-s* vemos estadísticas de todo el stack de IP de los distintos protocolos para ver si todo funciona como debería.

netstat -an

net: es una suite de comandos para ver y manejar servicios de red. Es un mega comando que tiene un monton de subcomandos para realizar distintas tareas. Por ejemplo cuando mapeamos una unidad de red en un servidor remoto, donde el servidor remoto nos comparte la unidad instaladores y nosotros lo queremos acceder desde nuestro equipo, montamos esa unidad remota a una letra en nuestro equipo, entonces por ejemplo la letra f (unidad) va a ser lo que hay en el recurso compartido remoto. Entonces con *net use* podemos ver las asociaciones que tenemos hechas, o hacer nuevas si lo requerimos.

net help command

ping: verifica la conectividad a nivel IP enviando paquetes de “Echo Request” de ICMP (Internet Control Message Protocol). Envía un paquete de un origen a un destino, y el destino envía un eco o rebote de eso, y de ese modo quien envía los paquetes sabe que se puede llegar al equipo remoto y que el equipo remoto está respondiendo.

ping [-t] [-a] [-n Count] [-l Size] [-f] [-i TTL] [-v TOS] [-r Count] [-s Count] [{ -j HostList | -k HostList }] [-w Timeout] [TargetName]

_ Si ponemos ping y una dirección IP, cuando se ejecute podemos ver el tamaño de los paquetes (cuantos bytes tiene), el tiempo de respuesta y el tiempo de vida del paquete (time to live o TTL). Por defecto muestra 3 paquetes de ping, pero con *-t* podemos hacer que se manden paquetes de forma interrumpida hasta que lo cortemos con ctrl C, esto es muy útil para encontrar problemas de red donde un dispositivo no se puede conectar a otro entonces capturamos el tráfico para ver si el ping llega o no, entre otros problemas. Ademas de una dirección IP podemos enviar una URL:

ping 10.0.2.2

ping -t www.google.com

_ Dependiendo de la distancia en donde este el host va a ser menor o mayor el tiempo al hacerle ping.

route: muestra y modifica las entradas de la tabla de ruteo IP local. Muestra información de la red. Si lo ponemos sin nada muestra la ayuda.

route [-f] [-p] [Command [Destination] [mask Netmask] [Gateway] [metric Metric]] [if Interface]]

_ El comando con *print* nos muestra la tabla de ruteo que es similar a lo que nos devuelve el comando *netstat -nr*, ya que son dos comandos donde el primero es especifico solo para rutas (ver, agregar, modificar rutas), y el otro es general, donde tiene múltiples funciones de red, pero que en definitiva ambos comandos muestran la misma información que es la tabla de ruteo del sistema operativo.

route print

_ Si necesitamos agregar una ruta por el motivo que sea hacemos lo siguiente, y luego mostramos la lista actualizada veremos la ruta que agregamos:

route add direccion_IP *mask* mascar_IP

route print

telnet: permiten la comunicación con un host remoto utilizando el protocolo Telnet. También puede utilizarse para establecer conexiones TCP en distintos puertos. Es decir establece una conexión TCP de un origen a un destino, y de esta forma sabremos si nos estamos pudiendo conectar a un determinado puerto de un determinado destino.

telnet RemoteServer [port]

_ Con el siguiente ejemplo establecemos una conexión a ese servidor remoto en el puerto 80, donde si obtenemos una pantalla negra quiere decir que estamos conectados, por default este comando no muestra lo que uno escribe, pero si lo que responde el host entonces le mandamos un mensaje para ver que responde.

telnet www.google.com 80

GET/http/1.1

_ Entonces con este comando podemos ver si el sitio responde o no, o si nos pudimos conectar al responder significa que si pero de lo contrario significa que hay otro error. Este comando es una forma de bajo nivel de abstracción de poder probar una conexión y así podemos encontrar problemas si es que los hay.

tracert: determina el camino tomado a un destino enviando mensajes “Echo Request” de ICMP. Es decir, muestra como es la ruta para llegar desde donde estoy hasta un destino. Como resultado tenemos la dirección IP del destino.

tracert [-d] [-h MaximumHops] [-j HostList] [-w Timeout] [TargetName]

_ Sabemos que una red IP o internet es una gran telaraña de conexiones ruteadas entre distintos proveedores, equipos o routers, y para llegar de un origen a un destino voy saltando entre varias direcciones hasta llegar. Con el siguiente ejemplo podemos ver todos los saltos que ocurren desde un origen hasta un destino y el tiempo en cada uno:

tracert www.google.com

Estado e información del sistema

date: muestra la fecha actual del sistema. Si se ejecuta sin parámetros, permite ademas cambiarla.

date [mm-dd-yy] [/t]

_ Con */t* vemos solamente la fecha:

date -t

time: muestra la hora actual del sistema. Si se ejecuta sin parámetros, permite ademas cambiarla.

time [/t] [/time] [hours: [minutes [:seconds [.hundredths]]] [{ A|P }]]

ver: muestra la versión de Windows.

Referencia comandos Linux

Shells Linux/Unix

_ En Linux tenemos muchos shells utilizables. Recordamos que Linux en si es el kernel, o sea el núcleo, y todo lo que lo rodea es lo que llamamos una distribución de Linux, entonces así tenemos Ubuntu, Debian, Fedora, Oracle, Xubuntu, o la distribución que sea. Entonces cada distribución puede optar por rodear el kernel de distintas herramientas o shells de distintas librerías y así poder ir armando toolsets. Cualquier usuario de Linux puede elegir que shell usar.

_ Con respecto a los shells, hay distintos tipos con distintas características que lo hacen más adecuado para un uso u otro, donde la sintaxis de uno cambia respecto a otro. Algunos de los más comunes son:

Sh (Bourne Shell): es el originario, introducido en 1979 implementa las funciones mínimas comunes a cualquier shell Unix. Es como el más básico y de este surgen todos los otros. Este se usa en Unix antiguos o aquellos que no son Linux.

Bash (Bourne-Again Shell): surge como una implementación GNU más completa que sh, y es la más estándar y la más utilizada en distribuciones Linux. Además, es compatible con lo que es sh.

dash: subconjunto de bash utilizado por Debian. Es un shell acotado, más pequeño y por ende más portable y más eficiente, entonces por ejemplo Debian lo utiliza para todos sus scripts del sistema operativo de arranque, etc.

Ksh (Korn Shell): este es un shell que tiene su propia sintaxis, y este es utilizado en algunos DSD por ejemplo Open DSD.

Csh (C Shell): este es un shell que se caracteriza por tener la sintaxis de scripting muy similar a lo que es el lenguaje de programación C.

Zsh (Z Shell): este es un shell que tiene un montón de utilidades extras, es como un shell recargado con funciones de autocompletado, color, etc, que los usuarios avanzados usan para tunear su sistema.

Manejo de archivos y directorios

_ En Linux, el resultado de la ejecución del comando anterior queda siempre guardado en una variable, que es ?, pero para ver su valor se pone $?, y con *echo* vemos los valores de la variable. Entonces con echo $? recibimos la salida del último comando, o sea que devolvió, en donde si da 0 es porque se ejecutó bien, si ejecutamos algo que de error al aplicar este comando nos va a dar un numero distinto de 0. Esto es muy importante ya que cuando uno programa scripts donde combina varios comandos, muchas veces se pregunta por si el resultado de la ejecución del comando anterior fue correcto o no. Cada número tiene un significado y pueden ser configurables.

_ Como vimos en Windows que tenemos el /? O /h para buscar ayuda de la sintaxis de algún comando, pero en Unix y Linux tenemos manuales de uso de todos los comando. Entonces con el comando *man* (de manual) podemos ver el manual de cada uno de los comandos, donde tenemos el nombre, la sintaxis, y cada uno de los parámetros con sus significados y sinónimos, también hay una sección de ejemplos para ver cómo se utilizan.

_ Linux y Unix son sistemas operativos key sensitive donde no es lo mismo mayúscula que minúscula.

cat: muestra los contenidos de un archivo a stdout. Recibe como argumento el archivo que se quiere ver. Es decir muestra el contenido de un archivo. Es equivalente al type de Windows.

cat archivo [>|>] [archivo de destino]

cd: significa change directory y permite cambiar de un directorio para ir a una nueva ubicación. Podemos movernos de un directorio a otro. Ponemos *cd* y la dirección del directorio donde nos queremos posicionar.

cd ruta

_ En Linux, para indicar las rutas y los directorios hay caracteres especiales que tienen un determinado significado. Todos los archivos que empieza con punto "." es por default oculto. En cuanto a los directorios el punto "." indica el directorio en el que estamos parados, y doble punto ".." indica el directorio de nivel superior. Entonces con *cd ..* volvemos al directorio de nivel superior. Con *cd -* volvemos al último directorio en el que estábamos . Y con *cd ../../* volvemos al directorio raíz o retrocedemos los directorios que queramos. Ahora / es el directorio de raíz para abajo.

cmp: compara los contenidos de dos archivos. Si no hay diferencias entre los archivos, no devuelve nada. De otro modo, se muestra en que difieren uno de otro.

cmp [-ls] arch1 arch2

cp: viene de copy y permite copiar dos archivos, similar al comando *copy* de Windows, donde uno pone el origen y el destino del archivo que uno quiere copiar, donde el origen y el destino puede ser o un archivo o la ruta completa para llegar a ese archivo dependiendo si usamos, una referencia absoluta o del directorio donde estamos parados.

cp [-R] origen destino

_ Con la opción *-R* hacemos que la copia sea recursiva, entonces eso nos sirve para copiar un directorio porque si no nos daría un error de que el directorio no es un archivo y que no lo puede copiar.

cut: extrae columnas de datos, que pueden ser bytes, caracteres o campos de una línea en un archivo. Este se usa bastante para procesar archivos de texto, donde tenemos campos separados por algún delimitador, por ejemplo un archivo CSV, donde con el comando *cut* podemos tomar de ese archivo la información que nos interese.

cut [-cdf lista] archivo

_ Con la opción *-d* (delimiter), uno especifica cual es el carácter que va a separar los campos, por ejemplo si fuese un CSV podría ser la coma ",", en otro ";", tab o el carácter que nosotros definamos. Y después *-f* (field) cuales son los campos que nos interesan saber. Por ejemplo, el siguiente comando muestra de ese archivo solamente los dos campos que nos interesan.

cut -d ":" *-f* 1,3 /etc/group

_ A este comando lo podemos combinar con otros y también podemos hacer consultas.

diff: como su nombre lo indica, se utiliza para determinar las diferencias entre archivos o directorios. Este comando es muy utilizado para generar parches, ya que por ejemplo cuando exportamos parches de GitHub o de cualquier sistema de control de versiones de software que utilicemos, normalmente usamos una sintaxis donde cada línea comienza con un "+" o un "-" dependiendo si hay que agregar o quitar esa línea al aplicar el parche,

entonces con el comando *diff* podemos ver si los archivos son diferentes, en que se diferencian y generar ese tipo de archivo que luego van a ser usados por el parche.

diff [-iqb] arch1 arch2

_ Por default si no ponemos ninguna opción, me muestran en que difieren los archivos, y con el signo > a modo de flecha me indica que el primer archivo es diferente al segundo en tal campo. Con el parámetro *-u* podemos ver más información en detalle. El *-q* (quiet) lo hace de forma silenciosa y pregunta si los archivos son iguales o no, donde no me muestra las diferencias sino que simplemente cambia el valor de salida del comando especificando si es igual o distinto, y si luego de esto ejecutamos echo $?, si son distintos, vemos que nos da distinto de 0, de lo contrario si son iguales da 0.

du: significa disk usage y muestra el uso de disco de archivos y directorios. Es decir, muestra la capacidad que esta usada por un determinado archivo o directorio o ubicación. Este es muy útil cuando uno necesita saber cuál es la distribución de espacio dentro de nuestro árbol de directorios. También con *du* podemos encontrar que es lo que se está comiendo todo el espacio.

du [-askh] archivos

_ Si al comando lo ponemos solo, tanto en el directorio en el que estamos para abajo nos muestra, para cada uno de los archivos, la utilización de datos de cada uno en bytes y finalmente nos indica el resumen del total ocupado en ese directorio donde estamos parados. Como el formato en bytes no es muy amigable para humanos, tenemos la opción *-h* (human) donde lo muestra en formato humano entonces lo convierte en kilo, mega o giga según corresponda. Si queremos saber el total de donde estamos parados ponemos la opción *-s* (sumary), y así con *-sh* vemos un resumen total del directorio en formato humano. Ahora si queremos que en un solo nivel nos muestre el espacio ocupado para saber dentro de los directorios que hay cuanto ocupa cada uno tenemos la opción:

du –h --max-dpth=1 .

_ *dpth* es de profundidad y en este caso es de profundidad máxima 1 entonces desciende un solo nivel de directorio y de ese modo nos va a dar un resumen por cada directorio. Por ejemplo, una combinación de comando para solo mostrar los directorios que ocupan Mb mediante *grep*, y además los ordenamos con *sort* de menor a mayor:

du –h --max-dpth=1 | *grep* M | *sort* –n

_ Si por ejemplo queremos saber en una sola línea cual es el directorio, usamos *tail* para ver el final de la lista de archivos que viene como entrada, por ende de este modo vamos a ver la última entrada y tenemos en una línea el resultado:

du –h --max-dpth=1 | *grep* M | *sort* –n | *tail* -1

file: como su nombre lo indica, determina el tipo de archivo en cuestión y muestra en la pantalla la información del mismo. Recordamos que en Windows, el tipo de archivo está determinado por su extensión, entonces si a un archivo lo nombro como .exe es de tipo ejecutable, y así, pero en Linux la extensión no tiene nada que ver con el tipo de archivo, sino que es una nemotécnica para el usuario para saber de qué se trata el archivo, pero podría tener cualquier extensión o ninguna. Lo que hace Linux es, en el comienzo del archivo, los primeros bytes tienen especificado que tipo de archivo es. Entonces con el comando file aplicado a un archivo, y me extrae esa información y me dice que tipo de archivo tenemos.

file archivo

_ Entonces dependiendo del tipo de archivo, es la información que va a mostrar.

find: sirve para buscar y encontrar determinados archivos o directorios. Este tiene una sintaxis particular porque lleva en un comienzo el punto a partir del cual quiero empezar a buscar, por ejemplo si estamos en el directorio /usr, o hacemos find . o es equivalente poner *find* /usr/share.

find [ubicacion_de_origen] [-type fdl] [-name patron] [-exec comando {} \;]

_ Tenemos distintos parámetros, según el tipo de archivo podemos usar -*type*, por ejemplo buscar los archivos -*type f* (file) o directorios -*type d* (directory). También si queremos buscar archivos o directorios con un patron, por ejemplo el nombre, tenemos lo que es -*name* y le agregamos * para escapar el asterisco. Un ejemplo seria:

find /usr/share -*name* *.png

_ Además podemos combinar lo que encontramos con la ejecución de un comando, entonces con la opción *exec* indicamos cual va a ser el comando que queremos ejecutar por cada una de las ocurrencias del archivo que estemos buscando. Entonces, el siguiente código nos dice que para cada uno de los archivos tar o gz que encuentre, los va a listar de forma extendida con *ls*, {} representa el conjunto de archivos encontrados y agregamos un ";" al final de cada línea porque en Linux si queremos ejecutar más de un comando en una línea, lo separamos con el carácter ";".

find /usr/share -*name* *.tar.gz -*exec ls -l* {} \;

grep: permite buscar una palabra, un patrón o secuencia de caracteres en uno o más archivos. Permite también buscar palabras dentro de un archivo o lista de texto que se le pase al comando. Este también se lo suele combinar con otros comandos para encontrar y filtrar información.

grep [-viw] patrón archivo

_ Por ejemplo si combino el comando *dmesg*, que tiene mucha información de booteo del sistema operativo, con grep, podemos filtrar información, y el comando seria:

dmesg | *grep* ACPI

_ Pero además, si queremos por ejemplo encontrar la información relacionada con la palabra power independientemente de que sea mayúscula o minúscula, ponemos la opción *-i* (key insensitive). Si queremos que cuente las palabras ponemos la opción *-c* (count), donde seria *-ci*. También tenemos la opción *-w* (world) para mostrar los que tengan una palabra textual.

dmesg | *grep* ACPI | *grep -ci* power

head: muestra las primeras líneas de la salida de un archivo. Este es la contraparte del comando tail. Por default muestra las primeras 10 líneas, pero si queremos setear un número de líneas le ponemos seguido del comando *-n* n_lineas.

head [-lineas | *-n* numero] archivo

_ Un ejemplo seria del comando *dmesg* estamos interesados en ver las primeras 20 líneas:

dmesg | *head -n* 20

ln: viene de links y lo que hace es crear enlaces o links simbólicos a archivos o directorios. Un link simbólico es como una referencia o algo que apunta a otro archivo o directorio con otro nombre y ubicación, con lo cual uno puede fácilmente hacer como referencias o accesos directos a otra información usando estos links. La opción *-s* es de link simbólico. Esto sería similar a un puntero.

ln [-s] origen destino

_ Por ejemplo, si ejecutamos esto se generó un link donde ahora podemos llegar al archivo1 como este mismo o como prueba:

ln -s archivo1 prueba

ls: lista los archivos (y subdirectorios) de un directorio. Tiene un montón de opciones, por ejemplo *-a* (all) me muestra todos los archivos, no solo los visibles, *-h* lo muestra en formato humano, *-l* es para hacerlo en formato long o extendido donde nos muestra información adicional para cada archivo o escritorio (permisos, propietario, usuario/grupos, tamaño, fecha de última modificación, nombre).

ls [-la1] archivo_o_directorio

_ Cuando listamos archivos estos comienzan con “-“, cuando es directorio comienza con una “d”, y cuando es un link simbólico empieza con una “l”.

mkdir: es para crea un directorio. Es bastante simple su uso, ponemos el comando seguido el nombre o la ruta donde se va a crear el directorio.

mkdir directorio

_ Por ejemplo si queremos crear un directorio dentro de otro seria como vemos a continuación, donde el directorio “a” tiene adentro el directorio “b”:

mkdir -p a/b

mv: viene de move y se utiliza para mover o renombrar archivos y directorios. Similar al copy pero este mueve desde un origen a un destino.

mv [-if] origen destino

_ Por ejemplo si queremos mover una carpeta a un directorio superior hacemos:

mv b ../

pwd: muestra por pantalla la ruta completa del directorio actual de trabajo.

rm: significa remove y con este podemos eliminar archivos o directorios. Si queremos eliminar un directorio colocamos *-r* (recursivo), y si es un archivo por default se usa el comando solo. En el caso de que se pidan permisos usando la opción *-f* se fuerza a borrar. Si ejecutamos *rm -rf* / como usuario root borra todo el directorio raíz sin preguntar de forma recursiva para abajo.

rm [-rif] archivo_o_directorio

tail: muestra las ultimas líneas de un archivo. Este al igual que head por default muestra las ultimas 10 líneas, pero si queremos setear un número de líneas le ponemos seguido del comando *-n* n_lineas.

tail [-lineas | -fr] archivo

wc: viene de word count y permite contar las líneas, caracteres y/o palabras que tiene un determinado archivo o secuencia de texto. La opción *-l* (lines) cuenta cuantas líneas tiene la salida, la opción *-w* (words) cuenta palabras, y la opción *-c* (characters) cuenta caracteres.

wc [-lwc] archivo

_ Por ejemplo, lo aplicamos en el comando *dmesg* para contar cuantas líneas tiene:

dmesg | *wc -l*

Archivado y compresión de datos

gzip: permite comprimir archivos o directorios. Es una utilidad de compresión que tiene Linux y es la más utilizada (estándar) pero hay otras mejores. No es compatible con winzip. Con *gzip* lo que hacemos es comprimir archivos o directorios y podemos pasarle como parámetros distintas opciones como *-r* (recursivo) para directorios, la opción *-v* (verbose) en cualquier comando de Linux es por lo general mostrar una salida verborragia

y que muestre que está haciendo y si está andando bien. Luego de todo esto tenemos que pasarle el archivo.

gzip [-rv9] archivo

_ Una vez que se hace genera un nuevo archivo.gz para indicar que esta comprimido. Este comando tiene 9 niveles distintos de compresión, donde por default usa 3 pero 9 es el máximo nivel de compresión. Cuanto mayor es el nivel más se tarda en procesar, más pesado es el procesamiento y menor es el espacio del archivo generado. Normalmente con -3 que usa por default es suficiente, pero hay casos en los que debemos comprimir más.

gunzip: descomprime un archivo a su forma original. Es la contraparte de gzip, donde si lo ejecutamos generamos de nuevo el archivo original con su tamaño original. Le pasamos como parámetro el archivo comprimido.

gunzip [-v] archivo

tar: viene de tape archive (archivador en cinta). Permite archivar múltiples archivos y directorios en un único archivo. Concatena o pega un archivo o directorio detrás de otro y el resultado de eso lo manda a un solo archivo donde va a tener todo el contenido. Este comando no comprime. Pero lo podemos combinar para comprimir.

tar [t] [c] [x] [v] [z] [f destino] origen

_ Por lo general en los comandos se manda primero el origen y luego el destino, pero acá es al revés. Con la opción *-c* (create) creamos el archivo, con la opción *-f* le decimos en que archivo va a crear el archivador tar.

zip/unzip: otra utilidad para compresión/descompresión de datos que es compatible con sistemas como MS-DOS y Windows NT.

zip/unzip archivo

Estado y configuración de red

Socket: es una asociación biunívoca que me permite identificar una conexión, esta está dada por la dirección IP de origen (local adress) y destino (foreign adress), y los puertos.

_ Para instalar componentes de red hacemos:

sudo apt install net-tools

arp: permite manipular el cache ARP del sistema operativo en el que estamos. Muestra la tabla ARP, donde está el mapeo que hay entre una dirección IP y la correspondiente dirección MAC o física del adaptador. Tenemos la opción *-n* muestra sin nombres la dirección IP y la mac. Y con la opción *-s* podemos hacer una asociación estática entre una dirección IP con una mac.

ifconfig: podemos configura una interfaz de red. Además, muestra toda la información de las placas de red y nos permite modificar o cambiar los parámetros que necesitemos. El comando sin nada me muestra todos los adaptadores de red que tenemos definidos en el sistema.

ifconfig [-v] [-a] [-s] [interface]
ifconfig [-v] interface options | address

_ Si queremos agregar una dirección IP a un adaptador de red, debemos hacerlo con privilegios de root, por ende usamos el comando sudo, y seria:

sudo ifconfig nombre_interfaz direccion_IP *netmask* numero_mask

_ Y si vemos la configuración del adaptador vemos los cambios.

netstat: muestra información o status de la red, conexiones de red, tablas de ruteo, estadísticas de interfaces, etc. Con la opción *-a* (all) muestra todo uno por uno, normalmente se lo usa con la opción *-n* (no resolve) para que no resuelva, entonces seria *-an* y muestra todo de una. Muestra información del protocolo que estamos usando por ejemplo TCP, las direcciones de origen y destino y el tipo de conexión o configuración de red que tenemos (Listen, Established, etc).

netstat [-t] [-u] [-l] [-a] [-n] [-p] [-v]

_ La opción *-l* (listen) vemos solo las conexiones que tenemos escuchando, sumado a *-p* (process) vemos cual es el proceso que están escuchando esas conexiones.

ping: envía paquetes de Echo Request ICMP a hosts de la red, desde un origen a un destino. Es igual que Windows.

route: muestra y configura la tabla de ruteo IP del sistema. También permite agregar o quitar si necesitamos. Gracias a esta tabla de ruteo y nuestro destino vamos a saber el camino de la red por donde ir. Si no hay otras rutas vamos por la default gateway.

route { add | del } [-net|-host] target [netmask Nm] [gw Gw]

_ Para agregar una ruta usamos hacemos:

sudo route add -net direccion_IP *netmask* numero_mask *gw* numero_getaway

traceroute: muestra la ruta hacia un host determinado. Marcamos la ruta hacia un determinado destino, y muestra todos los saltos que hace desde nuestro origen hasta llegar al destino. Ponemos la dirección IP o el nombre al que queremos acceder, -n para que no resuelva, y nos va a dar todos los saltos desde el origen al destino.

traceroute host

Estado e información del sistema

dmesg: muestra los mensajes de estado que genera el kernel durante el proceso de arranque.

free: muestra estadísticas sobre la utilización de la memoria. Si lo ponemos sin nada lo muestra en bytes, y con la opción *-h* lo mostramos en formato humano. Muestra información sobre la memoria swap o de intercambio.

kill: sirve para matar procesos, pero en realidad se utiliza para enviar una señal a un proceso. Con *kill -SIGTERM* decimos que se cierre o termine un proceso, y a la sintaxis de esto le agregamos el process ID.

kill [-señal] *pid*

_ Con *kill -l* vemos todas las señales que hay. Y con *-SIGKILL* terminamos el proceso que está corriendo por la fuerza, sobre todo cuando el proceso este clavado.

ps: produce un reporte de todos los procesos que se corren en un sistema. Vemos el listado de los comandos. Con la opción *-ef* muestra todos los procesos del sistema operativo corriendo. La opción *-l* muestra en formato long y da más información. Con la opción *-o* podemos ver las columnas o campos que a mí me interesen.

ps [-e] [-f] [-l] [-w] [-o]

shutdown: permite cerrar el sistema para apagarlo o reiniciarlo. La opción *-r* (reboot) es para reiniciar el equipo, *-h* (halt) es para apagar el sistema. Para hacerlo ahora colocamos *now* o ponemos una fecha para programar lo que queramos hacer.

shutdown [-r] [-h] [-c] [-k] [-t segundos] time

top: muestra una lista en tiempo real de los procesos corriendo en el sistema. A diferencia de *ps* que muestra como una foto de los procesos, *top* muestra un refresco permanente de 3 segundos por default, de los procesos. Nos muestra información como la hora actual, hace cuanto que esta encendido el equipo, cuantos usuarios hay conectados, la carga del sistema o servidor, numero de tareas corriendo y en que estados están los procesos, porcentaje de CPU que estamos usando, muestra la información de la memoria e información sobre la memoria de intercambio o swap, y al final la lista de procesos ordenados según el uso de CPU que se muestra con refresco constante.

uname: muestra información del sistema operativo actual. El comando solo nos dice el sistema operativo, con la opción *-m* (machine) vemos la arquitectura de la máquina, con la opción *-n* (name) nos muestra el nombre del sistema, con la opción *-r* (realase) nos muestra la versión del kernel, *-s* (system) nos muestra el sistema, y *-a* (all) muestra toda la información junta.

uname [-m] [-n] [-r] [-s] [-v] [-a]

uptime: muestra información generada desde el arranque del sistema. Por ejemplo muestra la hora actual, muestra el tiempo que hace que esta encendido un equipo, además de los usuarios y la carga.

Acceder al disco D en Windows

_ En la consola cmd los pasos son los siguientes:

```
diskpart
list volume
select volume *numero
assign letter D
remove letter D
```

Modelo TCP/IP

_ Este modelo es sobre el cual se cimienta y está basada toda la internet y la red como las utilizamos hoy en día.

Suite de protocolos de internet

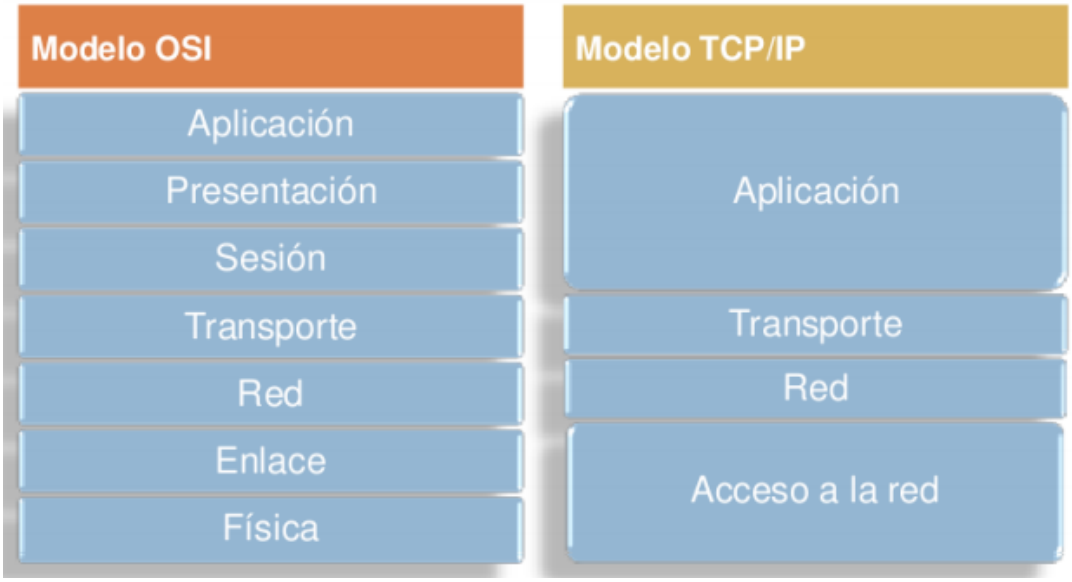
_ El protocolo TPC tiene distintas versiones, la primera que surgió fue la versión 4 que utiliza direcciones IP de 32 bits de longitud, y esto tiene una cantidad acotada de direcciones IP que ahora con el crecimiento desmedido de internet se está agotando, entonces surgió una evolución del protocolo TCP que es el protocolo TCP 6 que utiliza direcciones IP de 128 bits, entonces con eso hace muchísimo más grande el universo de direcciones IP posibles. Por ende tenemos IPv4 e IPv6, lo que usamos nosotros en la diaria es la 4 pero los grandes sitios, las nubes y otros desarrollos ya utilizan versión 6 con direcciones IP más largas con hexadecimales.

Comparación OSI y TCP/IP

_ Originalmente se creó, por una asociación de empresas y entidades, el modelo o estándar OSI del stack de red, que presentaba siete niveles o capas, yendo de abajo hacia arriba la capa de física, enlace, red, transporte, sesión, presentación y aplicación. Este fue un modelo con el que se inició, es un poco más teórico, pero en la práctica los hackers o padres de la informática, simplificaron este modelo y establecieron el modelo TCP/IP donde básicamente se cumplen las mismas funciones que en el modelo OSI pero de una forma simplificada y se reduce a cuatro capas, donde la primera es la capa de acceso a la red, luego red, transporte y por último aplicación.

_ El que gestiona todo esto es el sistema operativo. Cada sistema operativo tiene su stack TCP/IP, que es este conjunto de funcionalidades. Donde a su vez cada sistema operativo tiene su implementación, por ejemplo Unix, Windows, Mac OS, pero todas hablan el

mismo idioma, por eso pueden entenderse entre distintos sistemas, pero cada uno lo implementa con sus herramientas.



Capa acceso a la red: la función de esta capa es de independizar el acceso al hardware para las capas superiores. En esta capa donde tenemos la parte física, podemos tener distintas tecnologías de distintos medios de comunicación, por ejemplo un medio de cobre, ethernet, óptico, radial (señales de radio, wifi). Las capas superiores van a saber enviar bits, y esta capa se va a encargar de interpretar los bits según el medio que utiliza, entonces además de independizar la parte física, hace el direccionamiento a nivel capa dos del modelo OSI que sería la capa de enlace, que es donde está la dirección MAC donde cada dispositivo físico o placa de red tiene una dirección única que lo identifica.

Capa de red: es la encargada del direccionamiento y ruteo de los paquetes, es decir, cada dispositivo conectado a la red tiene un identificador que es su dirección IP, donde esta es la función de direccionamiento, y también tiene una función de ruteo donde tenemos todas las rutas cargadas y saber cómo llegar desde un extremo al otro de una comunicación pasando o saltando por routers y así se va armando el path o camino hasta llegar al destino. En esta capa conectamos los dispositivos.

Capa transporte: la función que tiene es conectar las aplicaciones de ambos extremos, es decir, como en la placa de red conectamos los dispositivos, pero dentro de cada dispositivo podemos tener distintas aplicaciones funcionando, entonces la capa de transporte se encarga de conectar aplicaciones. Por ejemplo el navegador web que tenemos por un lado y el servidor web que brinda o publica contenido en el otro extremo. Esta capa tiene otras funciones como control de errores, recuperar paquetes, tiene la función de segmentación (cuando un paquete es más grande que la unidad máxima que soporta el medio, esta capa parte la información para que quepa en el medio y luego rearmarlo en el extremo remoto). Cumple las funciones de:

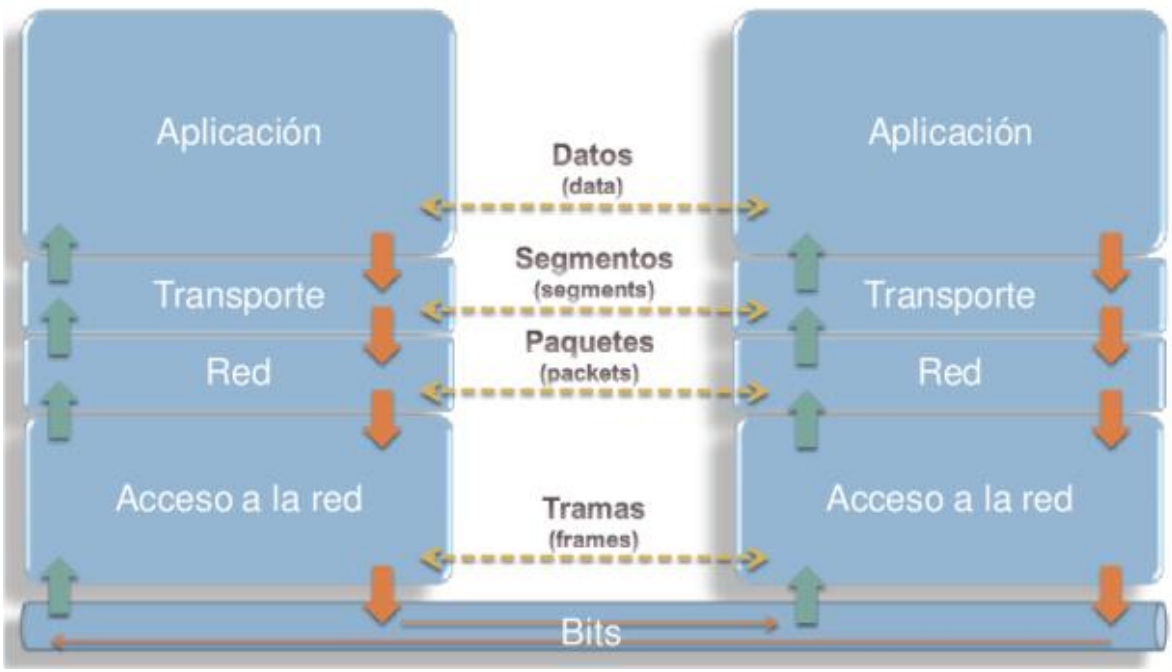
- Control de flujo: cuando hay disparidad entre las velocidades de acceso de un extremo al otro, esta capa negocia las velocidades.
- Direccionamiento de puertos: se definen los puertos TCP y UDP que se asocian a una determinada aplicación corriendo en el dispositivo.

Capa de aplicación: es donde se están ejecutándose las aplicaciones. Esta capa es la encargada de manejar la interfaz con el usuario o los sistemas que estén interactuando, y enviarlo a través de la red por las capas inferiores.

Transmisión de datos

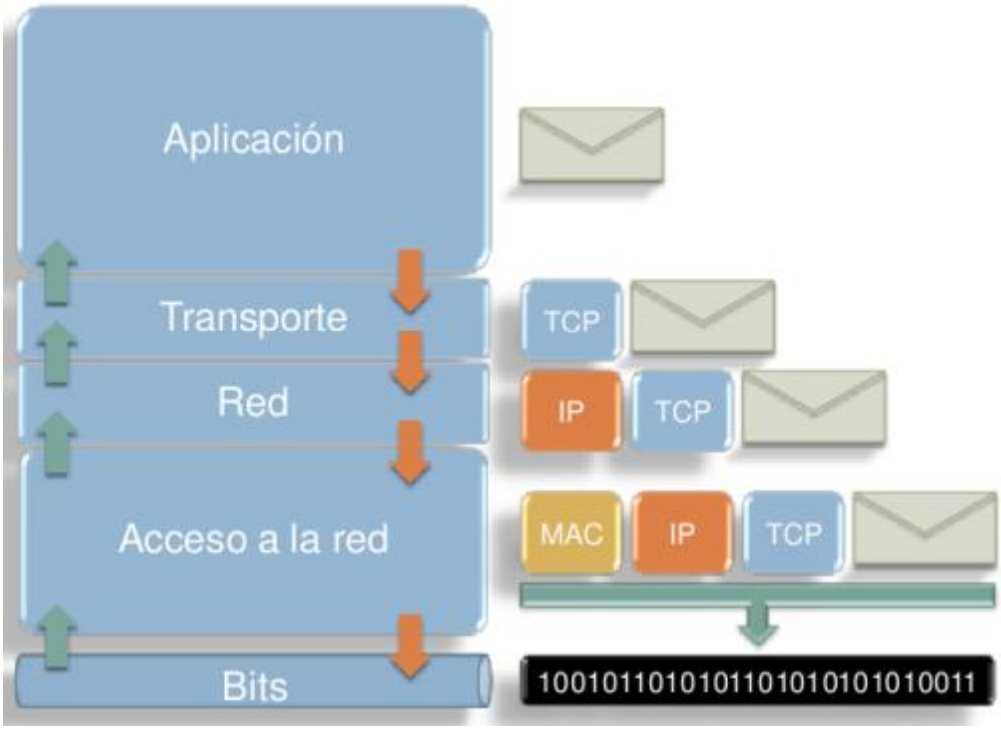
_ En cada capa de este modelo hablamos de distintas unidades de información de datos. En el medio físico en definitiva lo que siempre se transmite son bits representados según sea el medio que utilicemos. En la capa de acceso a la red lo que se trafica son tramas o frames que es un paquete o la unidad de esta capa. En la capa de red decimos que enviamos paquetes de una capa a otra del extremo remoto. En la capa de transporte se envían segmentos, y finalmente en la capa de aplicación se envían datos.

_ Entonces la información va fluyendo de una capa hacia las capas superiores y viceversa, donde cada capa va cumpliendo una función para llegar al destino y que todos se entiendan y hablen el mismo idioma. Cada capa habla en el mismo idioma con la misma capa del extremo remoto. Todo esto funciona mediante un proceso que se llama encapsulamiento.



Encapsulamiento

_ Cada capa va encapsulando la información que proviene de la capa superior y de ese modo le va agregando la funcionalidad. Si tomamos como ejemplo el correo electrónico, podemos tener una aplicación que sea un cliente de correo con el cual generan un mail, donde ese mail es la información que queremos transmitir, pero para que llegue al extremo remoto tiene que pasar por todas las otras capas, entonces esa información se va encapsulando en las capas inferiores, y cada capa le va agregando un encabezado. Entonces el mail cuando llega a la capa de transporte se le agrega el encabezado TCP, cuando pasa a la capa de red se le agrega el encabezado I, y luego a todo eso se lo encapsula en la capa de acceso a la red donde se le establece un encabezado que llamamos MAC y finalmente todo eso se termina siempre traduciendo a bits.

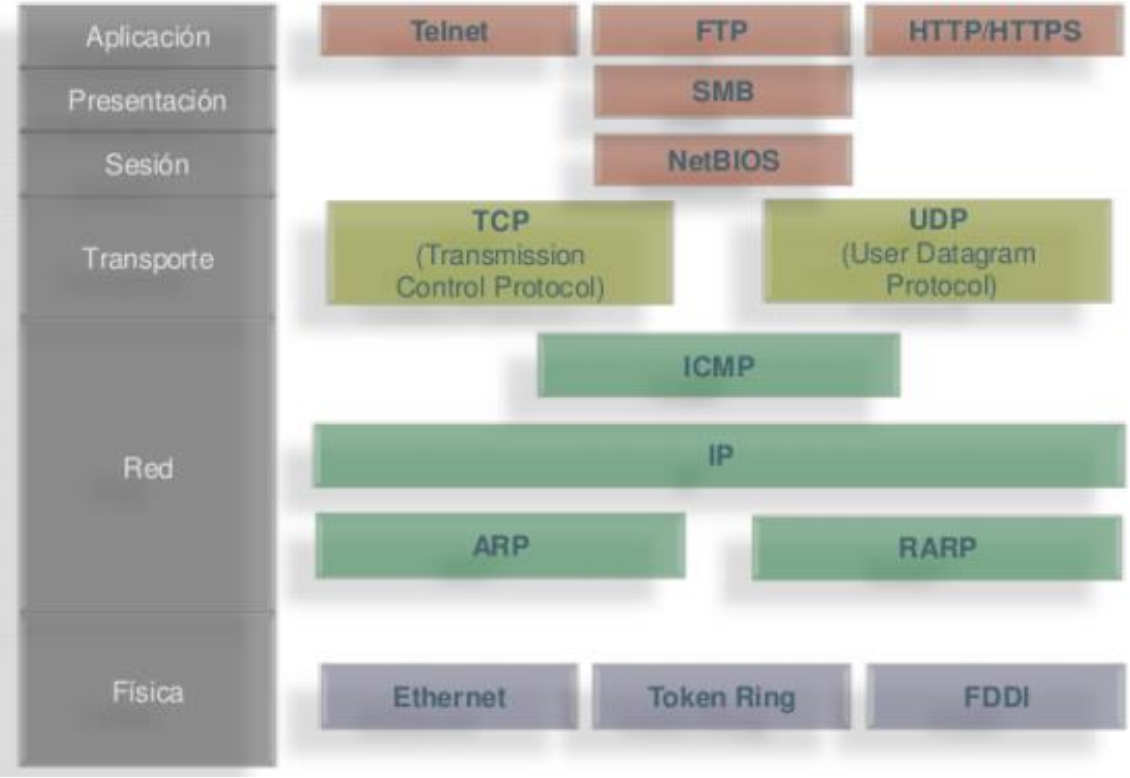


Protocolos TCP/IP

_ En la imagen vemos algunos ejemplos de protocolos TCP/IP y vemos a que capa del modelo corresponden cada uno.

- En la capa física o de acceso al medio tenemos protocolos como el ethernet que usa el 90% del mundo, tenemos Token ring que usaba IBM en equipos viejos, y FDDI que es para fibra óptica. Cada medio tiene su protocolo correspondiente.
- El capa de red el más importante y conocido es IP que es el encargado de tener la dirección que identifica a cada host, y después hay protocolos complementarios de IP en esta capa. También el protocolo ICMP que por ejemplo usa el comando ping, pero tiene un montón de funciones además del echo request, entre otras.

- En la capa de transporte tenemos TCP y UDP que son los que más se utilizan, donde dependiendo de la aplicación que se le vaya a dar a la comunicación, va a ser más conveniente utilizar UDP o TCP.
- En la capa de aplicación donde tenemos aplicación, presentación y sesión, tenemos protocolos como NetBIOS en sesión, que es el protocolo de Windows que permite la resolución de nombre o la publicación de equipos Windows que están en la red. El protocolo SMB en presentación, lo utiliza también Windows para la compartición de archivos e impresoras, en Linux este se llama SAMBA. Y en aplicación tenemos protocolos como TELNET para establecer conexiones TCP a un puerto, FTP para transferir archivos entre dos puntos de internet, HTTP/HTTPS que utilizamos todos los días para navegar en internet, entre otros.

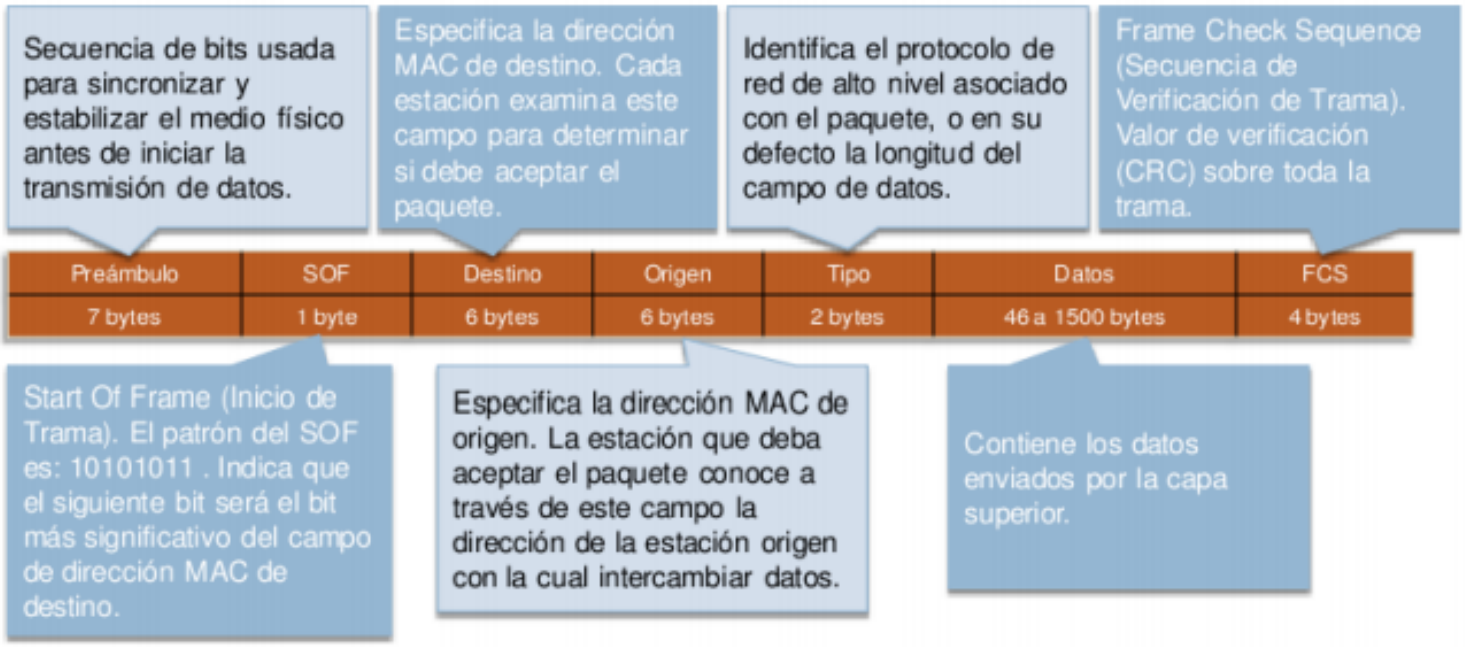


Nivel de acceso a la red

Trama ethernet

_ De los muchos protocolos que hay en la capa de acceso a la red, elegimos Ethernet porque es el más difundido. La trama ethernet es el único que nos solo tiene el encabezado sino un tráiler (cierre de la trama). En el siguiente esquema vemos que cada campo está definido según su longitud en bits, y cumple una función determinada. En los primeros siete byte tenemos lo que llamamos preámbulo, donde el medio físico es algo que puede variar o verse interferido, dañado, etc, entonces el preámbulo establece un

reseteo a partir de un punto conocido para comenzar la transmisión. Luego tenemos un byte que llamamos Start of Fram (SOF), luego en los próximos seis bytes vamos a tener la dirección MAC de destino de esta trama, y en el campo siguiente tenemos la dirección MAC del equipo de origen que este transmitiendo. A continuación tenemos un campo de dos bytes que identifica el nivel del protocolo de red del nivel superior del cual está viniendo información. Siguiente tenemos el capo de datos de 1500 bytes máximo donde encapsulamos los datos a transmitir, que en realidad esto va a ser el resultado de la capa inmediatamente superior, es decir, la capa IP. Por último tenemos la única unidad de transmisión que tiene un campo de cierre que lleva una suma de verificación de los campos anteriores para corroborar en el extremo remoto que no se ha alterado durante la transmisión la información de los otros campos.

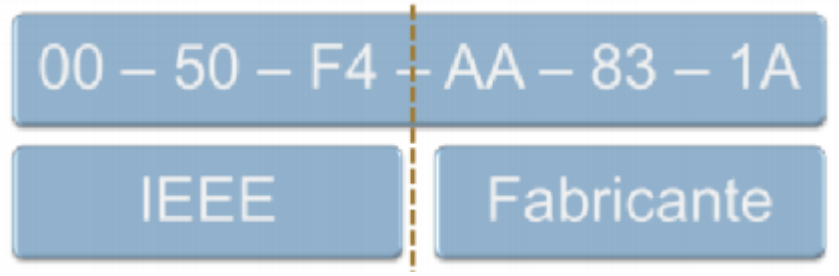


Direccionamiento MAC

_ En los campos de origen y destino de la trama ethernet, tenemos la dirección MAC (Medium Access Control) que proporciona un identificador único asignado a las placas adaptadoras de red por parte del fabricante.

- Tiene un numero de 48 bits o 6 bytes, donde se suele representar con 12 dígitos hexadecimales.
- Los primeros 24 bits representan al fabricante de la placa de red en cuestión: OUI (Organizational Unique Identifier), a esta la asigna la IEEE.
- Los segundos 24 bits son un numero serial puesto por el fabricante a sus N dispositivos: Serial Number.

_ En el sitio http://hwaddress.com/ si ponemos los primeros tres bytes de la dirección MAC podemos saber cuál es el fabricante correspondiente.



Nivel de red

Protocolo ARP/RARP

ARP (Address Resolution Protocol): protocolo de nivel de red responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP. Para que nuestro equipo pueda conectarse a otro que tenemos al lado en la red, este protocolo envía un broadcast (transmisión masiva que llega a todos los hosts de la red local), cuando se emite un mensaje de este tipo, ese paquete llega a toda la red lógica local y con este se pregunta a los equipos quien tiene la IP tal y cual es su dirección MAC, en donde solamente el equipo que tenga esa dirección IP contesta quien es y dice cuál es su dirección MAC. A partir de eso el equipo que hizo la petición (ARP request) deja cacheado en la tabla ARP del sistema operativo esa correspondencia entre la dirección MAC y la dirección IP.

RARP (Reverse Address Resolution Protocol): protocolo utilizado para resolver la dirección IP de una dirección hardware dada. Esta es la contracara de ARP ya que hace la función inversa, donde averigua cual es la dirección IP que corresponde a una dirección MAC dada. Como a esta información no podemos averiguarla automáticamente, RARP requiere de un servidor que tiene una tabla donde carga para tal IP tal MAC y así sucesivamente.

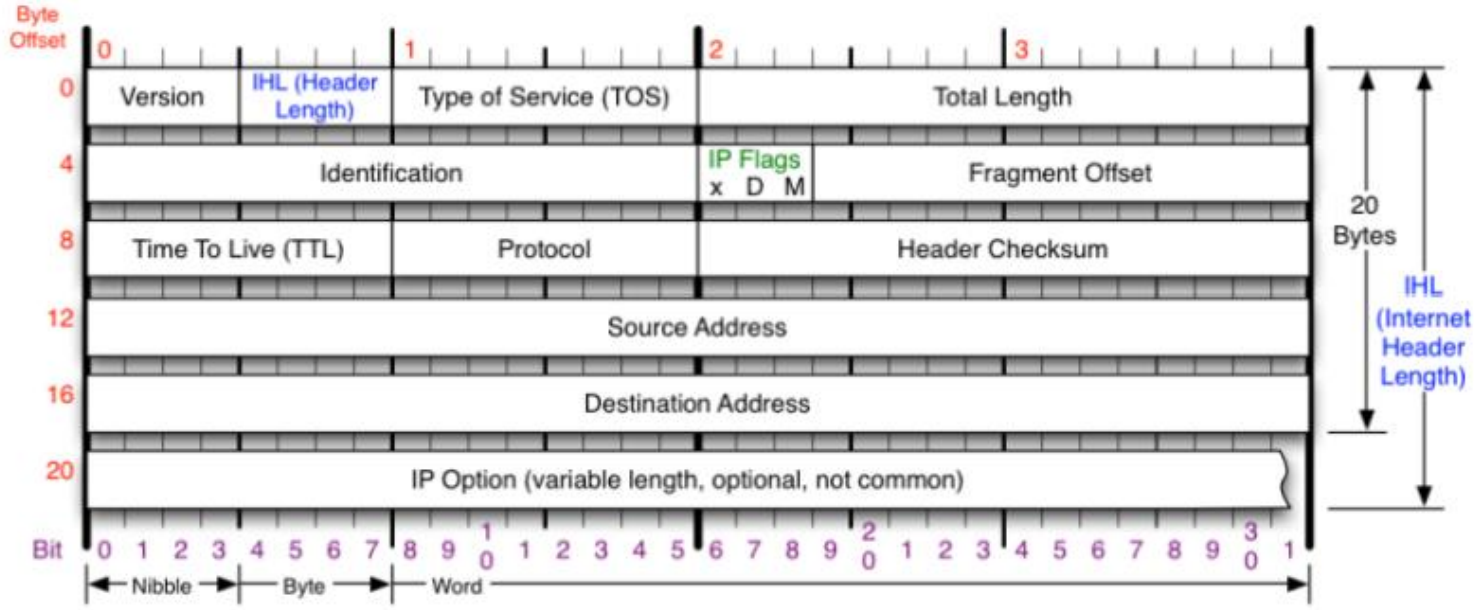
Protocolo IP

_ El IP (Internet Protocol) es un protocolo no orientado a conexión para comunicar datos a través de una red de paquetes conmutada.

Encabezado IP: cada protocolo tiene su encabezado IP y campo con distintas funciones:

- Campo versión: habla de la versión del protocolo, donde dice que es IPv4 e IPv6.
- Campo header length: dice la longitud del encabezado, que longitud va a tener la totalidad de sus campos y que está dado en el número de words (4 bytes = 32 bits).
- Campo tipo de servicio: se utiliza por ejemplo para priorizar cierto tráfico en las comunicaciones.

- Campo total length: es la longitud total del paquete IP, no solo encabezado sino también paquetes.
- Campo identificación del equipo (identification).
- Campo IP flags: tenemos una serie de flags donde podemos definir distintos valores.
- Campo time to live: identifica el tiempo de vida de un paquete.
- Campo protocol: donde se especifica el protocolo de capa superior que estamos transportando.
- Campo header checksum: suma de verificación para asegurarse de que no ha sido alterada o corrompida la información que hay en el header.
- Finalmente los campos más importantes que son la dirección de origen (source adress) donde va la dirección IP del equipo que transmite, y la dirección de destino (destination adress) donde va la dirección del equipo receptor.
- Después tenemos campos opcionales.



Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Protocol

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

IP Flags

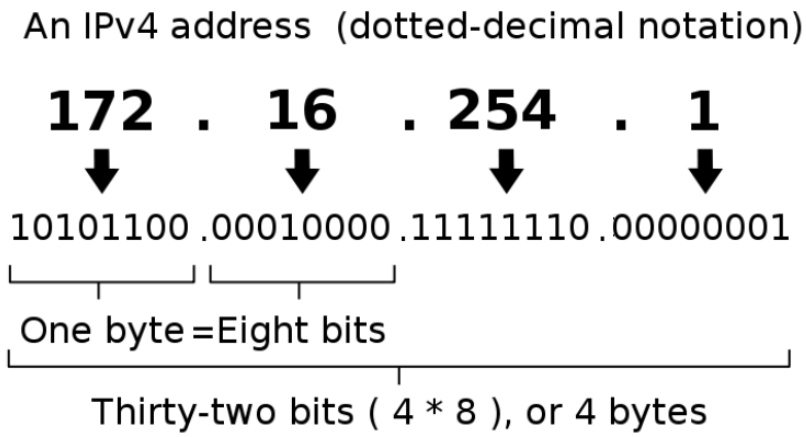
x D M

x 0x80 reserved (evil bit)
D 0x40 Do Not Fragment
M 0x20 More Fragments follow

RFC 791

Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Notación IP: esto es como se representan las notaciones IP. Los 32 bits de la dirección IP se representan en cuatro octetos o cuatro bytes donde utilizamos un numero decimal para representar cada uno. Y así representamos de una forma más entendible lo que es para nosotros inteligible. Las direcciones las representamos con números decimales, y cada dirección IP tiene más información que solamente el identificador del equipo, tiene una parte que identifica a la red y otra que identifica al host (equipo puntual dentro de esa red).



Prefijo de red: lo que determina que parte de la dirección IP se refiere a la red y que parte al host es lo que llamamos mascara de subred (subred mask), que en definitiva son bits seteados en uno que me indican que parte es red y que parte es host. La notación de la máscara de red puede estar en notación decimal separada por puntos, o la notación CIDR que se representa con /n_bits donde n_bits es el número de bits. Entonces como vemos en el ejemplo los tres primeros octetos representan a la red, y el octeto final representa al equipo dentro de la red. Con la dirección IP y con la máscara si hacemos un AND binario con estas dos, obtenemos un prefijo de la red (network prefix), es decir, me da que parte es la red y cuál es el host (host identifer).

	Binary form	Dot-decimal notation
IP address	11000000.00000000.00000010.10000010	192.0.2.130
Subnet mask	11111111.11111111.11111111.00000000	255.255.255.0
Network prefix	11000000.00000000.00000010.00000000	192.0.2.0
Host identifier	00000000.00000000.00000000.10000010	0.0.0.130

Subredes: con las notaciones pasadas podemos hacer particiones más chicas para obtener subredes. Lo que se busca en una red es lograr un tamaño optimo y se perdería mucha performance si todo estuviera en una misma red. En definitiva podemos tener una red

completa y subdividirla en subredes, jugando con la máscara, que en definitiva son un numero de bits que usamos para representar la red.

	Binary form	Dot-decimal notation
IP address	11000000.00000000.00000010.10000010	192.0.2.130
Subnet mask	11111111.11111111.11111111.**11**000000	255.255.255.192
Network prefix	11000000.00000000.00000010.10000000	192.0.2.128
Host part	00000000.00000000.00000000.00000010	0.0.0.2

Clases de direcciones IP: estas direcciones son según distintos rangos, donde dependiendo del tamaño de la red y de la organización puede convenir usar una clase u otra. También, en estas hay un rango de IPs privadas en determinados rangos en redes locales para que se distingan de las públicas y no afecten la comunicación, y que por convención son las que deben utilizar las organizaciones dentro de sus redes LAN ya que todo el resto de direcciones IP se utilizan en internet y son públicas, por lo que en el caso de tener el mismo rango en internet y en nuestra red local no nos podríamos comunicar porque no sabría si ir a la red local o la remota, por eso se establece que para redes privadas se usa tal rango:

Clase de Dirección IP	Bits de mas peso	Intervalo 1er Octeto	Bits de dir. de red	Mascara por defecto	Parte de Red y de Host	IPs Privadas
Clase A	0	0-127	8	255.0.0.0	RRR.HHH.HHH.HHH	10.0.0.0/8
Clase B	10	128-191	16	255.255.0.0	RRR.RRR.HHH.HHH	172.16.0.0-127.31.0.0
Clase C	110	192-223	24	255.255.255.0	RRR.RRR.RRR.HHH	192.168.0.0-192.168.255.0

_ La clase A es la que en general se usa para las empresas en donde dependiendo del tamaño de la organización convendrá usar un rango u otro. Y la clase C es la que en general se usa en los hogares donde tenemos un rango pequeño.

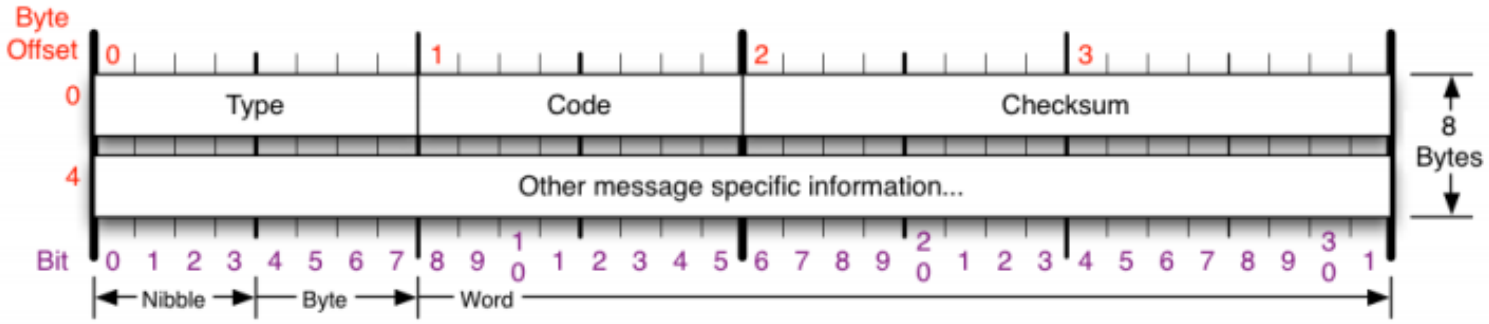
_ Con la herramienta o comando *ipcalc* podemos calcular toda la información referida a la red dándole la dirección IP y los bits de la máscara, donde si lo ejecutamos nos da información como la dirección, la máscara (netmask) expresada en sus dos formatos, tenemos un wildcard que es la inversa de la netmask, tenemos la dirección de la red (network), cual es el primer host que puede haber en la red (HostMin) y cuál es el máximo host (HostMax), además cual es la dirección del broadcast, donde también nos dice cuál es la cantidad de hosts que podemos tener en la red (Hosts/Net), sumado a la clase o tipo de red y si es privada o pública.

Protocolo ICMP

_ El protocolo ICMP (Internet Control Message Protocol) es el sub protocolo de control y notificación de errores del Protocolo IP. E un protocolo de ayuda a lo que es IP y se usa para enviar mensajes de error y de control, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado. Este es un protocolo muy útil, una de sus funcionalidades es el *ping*, pero debemos saber que puede ser utilizado malignamente. Muchas veces este protocolo entre redes se bloquea o se permite determinado tipo echo request o echo reply, porque puede utilizarse como un medio encubierto para enviar información entre dos software malignos.

Encabezado ICMP: el protocolo ICMP tiene un encabezado, como vemos es muy pequeño y simple:

- Campo type: muestra el tipo.
- Campo code: tiene un campo para el código.
- Campo chekcsum: una suma de verificación.
- Luego tiene datos en un campo de longitud variable.



ICMP Message Types

Type	Code/Name
0	Echo Reply
3	Destination Unreachable
	0 Net Unreachable
	1 Host Unreachable
	2 Protocol Unreachable
	3 Port Unreachable
	4 Fragmentation required, and DF set
	5 Source Route Failed
	6 Destination Network Unknown
	7 Destination Host Unknown
	8 Source Host Isolated
	9 Network Administratively Prohibited
	10 Host Administratively Prohibited
	11 Network Unreachable for TOS
3	Destination Unreachable (continued)
	12 Host Unreachable for TOS
	13 Communication Administratively Prohibited
4	Source Quench
5	Redirect
	0 Redirect Datagram for the Network
	1 Redirect Datagram for the Host
	2 Redirect Datagram for the TOS & Network
	3 Redirect Datagram for the TOS & Host
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
	0 TTL Exceeded
	1 Fragment Reassembly Time Exceeded
12	Parameter Problem
	0 Pointer Problem
	1 Missing a Required Operand
	2 Bad Length
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
30	Traceroute

Checksum

Checksum of ICMP header

RFC 792

Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.

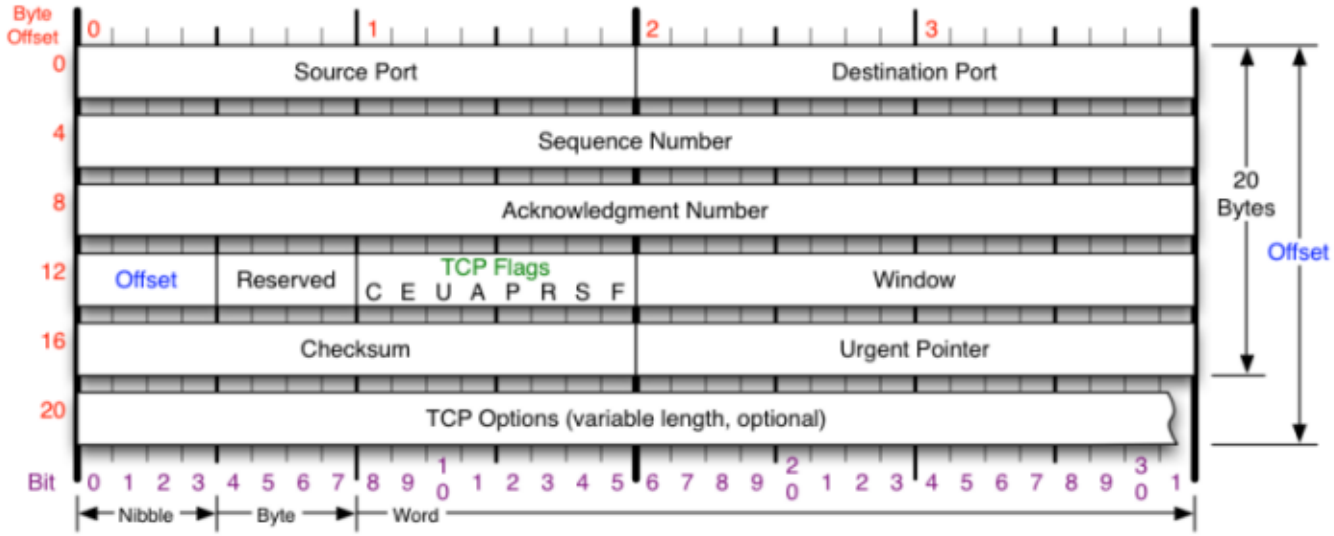
Nivel de transporte

Protocolo TCP

_ El protocolo TCP (Transmission Control Protocol): provee una entrega ordenada y confiable de un flujo de bytes desde un programa en una computadora hacia otro programa en un equipo remoto (cliente-servidor). Además es el encargado de ordenar y regular el flujo de información entre ambos equipos.

Encabezado TCP: este tiene un encabezado similar al de IP donde tenemos:

- Campo de origen y destino: source port y destination port, estos son muy importantes, ya que cada aplicación tiene normalmente un puerto asociado.
- Campo sequence number: número de secuencia.
- Campo acknowledgment número: número de reconocimiento.
- Campo TCP flags: es importante porque se definen distintas flags (valores binarios seteados en 0 o 1 para decir que está presente o no el mismo flag), y tenemos distintos tipos.
- Campo window: ventana.
- Campo checksum.
- Campo urgent pointer: para paquetes que son urgentes.
- Luego información de tamaño variable, que es lo que encapsulamos del protocolo de la capa superior.



TCP Flags

C E U A P R S F

Congestion Window
C 0x80 Reduced (CWR)
E 0x40 ECN Echo (ECE)
U 0x20 Urgent
A 0x10 Ack
P 0x08 Push
R 0x04 Reset
S 0x02 Syn
F 0x01 Fin

Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0
No Congestion	0 1	0 0
No Congestion	1 0	0 0
Congestion	1 1	0 0
Receiver Response	1 1	0 1
Sender Response	1 1	1 1

TCP Options

0 End of Options List
1 No Operation (NOP, Pad)
2 Maximum segment size
3 Window Scale
4 Selective ACK ok
8 Timestamp

Checksum

Checksum of entire TCP segment and pseudo header (parts of IP header)

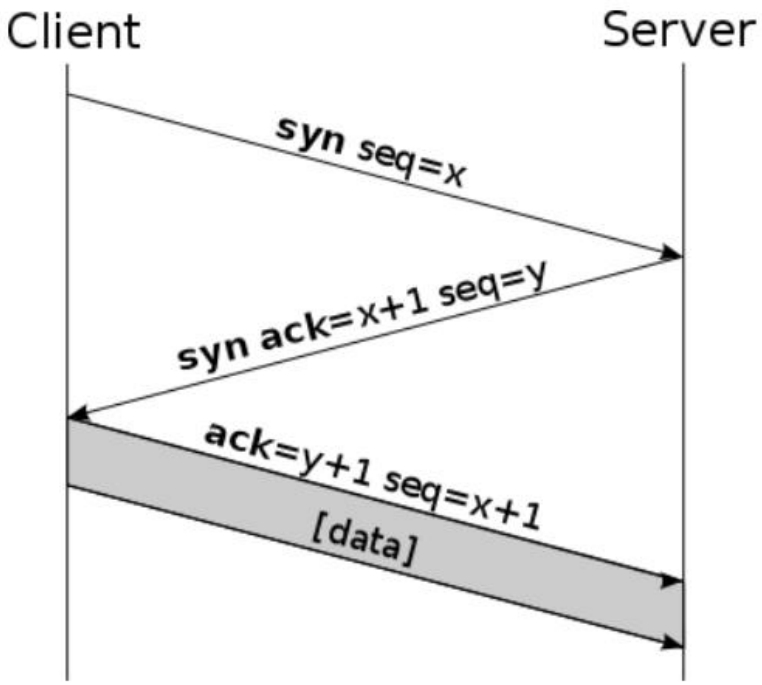
Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

RFC 793

Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.

Negociación de 3 vías: para establecer por primera vez la comunicación entre un origen y un destino se hace a través de lo que llamamos negociación de 3 vías (three way handshake). Lo que se hace básicamente es, quien inicia la conexión que es el cliente, envía un paquete con la flag de sincronización seteada y un numero inicial de secuencia cualquiera que se defina (syn seq = x), quien recibe esta conexión y la acepta, que es el server, responde con la flag de sincronización y acknowledge, y en el número de acknowledge pone lo que recibió del cliente incrementado en uno y genera un numero de secuencia nuevo que en este caso llamamos "y" (syn ack = x+1 seq = y), luego el cliente al recibirlo responde con el acknowledge seteado en el "y" que envió el servidor pero incrementado en uno y el número de secuencia original seteado en uno (ack = y+1 seq = x+1). A partir de que se establece esto, es que se establece la conexión y se puede empezar a transmitir libremente los datos, pero siempre existe esta negociación de 3 vías. En toda comunicación TCP, sea el protocolo que sea, utiliza esto.



_ La herramienta que utilizamos en este caso para realizar el laboratorio se llama Wireshark.

Algunos puertos TCP: muchas veces por convención y por estándares se definen cuáles son los puertos que utilizan determinados protocolos. Normalmente del 1024 para abajo son puertos fijos, reservados y estandarizados, y del 1024 para arriba son puertos más libres que uno puede utilizar. Entonces en base al puerto, podemos saber cuál es el servicio que esta atrás escuchando. Por ejemplo:

Puerto	Descripción
20	FTP Data
21	FTP Control
22	SSH
23	Telnet
25	SMTP
53	DNS
80	HTTP
110	POP3
123	NTP
143	IMAP
443	HTTPS

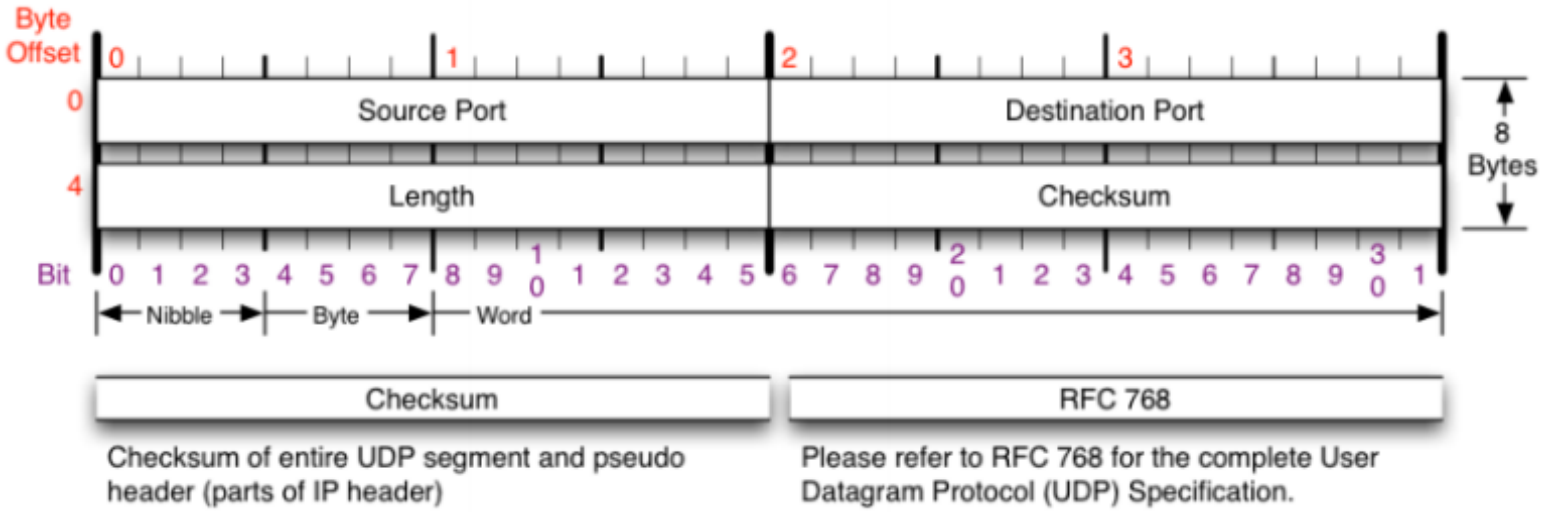
Protocolo UDP

_ El protocolo UDP (User Datagram Protocol) permite a las aplicaciones enviar mensajes (datagramas) a otros equipos en una red IP sin requerir el establecimiento de canales especiales de comunicación. A diferencia de TCP, este es un protocolo mucho más simple, liviano y eficiente. Estas no requieren del procesamiento propio de TCP. Algunos protocolos que se usan son DNS, NTP, entre otros.

Encabezado UDP: en el encabezado solo tiene:

- Campo de origen y destino: source port y destination port.
- Campo lenght: longitud.
- Campo checksum

_ Acá no tenemos un montón de herramientas que tiene TCP, pero si tenemos un stack muy pequeño y eficiente, haciéndolo apropiado para determinadas aplicaciones. Entonces si asumimos que la red que tenemos es bastante confiable y vamos a poder ignorar los mecanismos de control que tiene TCP, en muchas casos es aplicable el protocolo UDP.



Nivel de aplicación

Algunas aplicaciones de Internet

Protocolo	Puerto	Descripción
FTP	20/TCP 21/TCP	File Transfer Protocol: se utiliza para intercambiar archivos a través de una red basada en TCP/IP.
SSH	22/TCP	Secure SHell: permite el intercambio de datos a través de un canal seguro entre dos dispositivos de red.
Telnet	23/TCP	Proporciona una comunicación bidireccional orientada a texto a través de una conexión de terminal virtual.
SMTP	25/TCP	Simple Mail Transfer Protocol: estándar para la transmisión de correo electrónico a través de redes IP.
DNS	53/TCP 53/UDP	Domain Name System: servicio distribuido que traduce nombres de dominio en direcciones numéricas IP.
DHCP	67/UDP 68/UDP	Dynamic Host Configuration Protocol: es utilizado por los equipos para obtener en forma dinámica una dirección IP y otros parámetros de configuración.
HTTP HTTPS	80/TCP 443/TCP	Hyper Text Transfer Protocol: servicio para sistemas de información distribuida y colaborativa basada en hipertexto.

POP3 POP3S	110/TCP 995/TCP	Post Office Protocol: permite a clientes recibir correos desde un servidor remoto.
IMAP	143/TCP 993/TCP	Internet Message Access Protocol: permite la publicación y acceso al servicio de correo electrónico.

Criptografía

Definiciones y conceptos

Criptografía: es una ciencia que utiliza técnicas de distinto tipo para ocultar (cifrar) información de modo que solo pueda leerse (descifrarse) mediante la utilización de una llave o clave. Hay múltiples objetivos, pero fundamentalmente es hacer que nadie que no sea el destinatario correcto pueda acceder a la información que queramos proteger. Algunos puntos que debe cubrir:

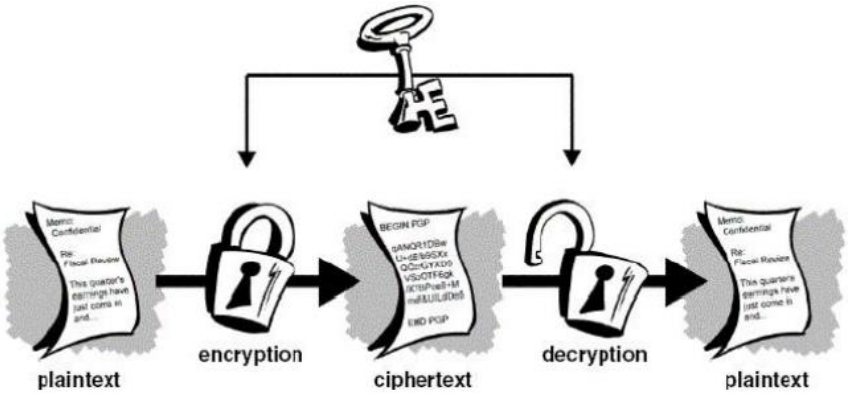
- Confidencialidad: al convertir un mensaje en otro cuyo contenido de información solo puede ser accedido por las personas o sistemas autorizados para luego descifrarlo y ver su contenido.
- Integridad: permite detectar si el mensaje ha sido modificado en su totalidad o en parte. Es decir, asegurarse de que esa información no solo sea confidencial sino que no se haya modificado en el proceso de almacenarla o transmitirla de un punto a otro.
- Autenticidad y no repudio: permite establecer en forma fehaciente la identidad del emisor. Es decir, autenticidad es que la información provenga de quien nosotros creamos que proviene, y no haya nadie que se esté haciendo pasar por el emisor por ejemplo, y el no repudio significa que nadie puede aducir que no fue el quien hizo determinada cosa.

Criptosistema: lo definimos como una quíntupla (M, C, K, E, D), es decir, cinco elementos que intervienen, donde:

- M: representa el conjunto de todos los mensajes o información sin cifrar (lo que se denomina texto plano, o plaintext) que pueden ser enviados. Cuando hablamos de mensaje hablamos de un conjunto de información, en donde cualquier cosa sin cifrar o que uno quiere cifrar es lo que llamamos texto plano, antes del proceso de cifrado.

- C: es la contraparte de M, y representa el conjunto de todos los posibles mensajes ya cifrados.
- K: representa el conjunto de claves que se pueden emplear en el criptosistema. Para poder producir el cifrado de una información en texto plano a su salida cifrada, se utiliza siempre una llave, clave o key.
- E: es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E_k para cada valor posible de la clave k con la cual se realiza el cifrado.
- D: es el conjunto de transformaciones de descifrado, análogo a E. O sea partiendo del texto cifrado aplicamos una función de descifrado usando también una llave y de ese modo tenemos el mensaje original de texto plano.
  - $c_i = E_k (m_i)$ (cifrado)
  - $m_i = D_k (c_i)$ (descifrado)
  - $m_i = D_k (E_k (m_i))$ (propiedad texto plano)

Criptosistema



_ Tenemos el mensaje original en texto plano, lo cual se le realiza una operación de cifrado, y con eso obtenemos el mensaje cifrado correspondiente al original, y en el otro extremo podemos hacer la operación inversa, aplicando el descifrado con la key y así obtenemos el mensaje original. El candado representa el algoritmo de cifrado y descifrado. Lo que vemos en candado hace referencia a esquemas, en donde hay distintos esquemas en cuanto a lo que son los algoritmos, y son cerrados y abiertos, análogamente seria:

Esquema cerrado: seria como hacer un candado super seguro, haciendo ocultas las especificaciones con las cuales está hecho el mismo porque eso le daría más seguridad y fortaleza. Este esquema se llama seguridad por ocultamiento. En los hechos este no da buenos resultados debido a la ingeniería reversa o hackers que terminan de entender cómo funciona esa caja negra y se encuentran las vulnerabilidades.

Esquema abierto: donde hacemos un candado abriendo todas las especificaciones de cómo está hecho para que cualquiera pueda verlo y encontrarle fallas, para que entre todos armar un candado más seguro si vemos que se puede mejorar.

_ La fortaleza de un esquema va a estar dado por la fortaleza de la llave criptográfica. La ”Clave” es la longitud de la ”Llave”, y esto es lo más importante, donde por ejemplo si tomamos una llave de 8 bits la cantidad de combinaciones posibles de la llave son 2^8 = 256, mientras que incrementamos la longitud de la llave por ejemplo a 40 bits seria 2^{40} = 1.099.511.627.776. La llave es un esquema matemático, y es lo único que puede utilizarse con tal algoritmo para cifrar o descifrar el mensaje. Entonces teniendo un mecanismo de cifrado aunque sea abierto pero sólido y una llave con la longitud correcta, podemos lograr la mayor de las seguridades.

Criterios de clasificación

Primer criterio

_ Podemos clasificar la criptografía de acuerdo a las técnicas empleadas en los algoritmos:

Criptografía clásica: es utiliza principalmente técnicas basadas en las operaciones de sustitución y transposición de caracteres. Su seguridad está basada solamente en el secreto de la transformación o del algoritmo empleado. Tenemos algunas operaciones de operaciones de criptografía clásica:

- Sustitución: consiste en el reemplazo de las unidades del mensaje original según una determinada transformación. Por ejemplo: El cifrado del Cesar, donde tenemos una tabla con el significado de cada unidad para descifrar.
- Transposición: consiste en el reordenamiento de las unidades del mensaje original según una determinada transformación, donde por ejemplo la letra A del mensaje original es la Z del mensaje cifrado. Ejemplo: La Escitala.

Criptografía moderna: es la que se utiliza hoy en día. Prácticamente toda la criptografía actual se basa en las teorías de la información y grandes números, la matemática discreta y la complejidad de los algoritmos, además de utilizar las operaciones clásicas de sustitución y transposición.

Segundo criterio

_ También de acuerdo al tipo de llaves utilizadas en los algoritmos:

Criptografía simétrica: utiliza la misma llave tanto para cifrar y como para descifrar la información. Hablamos también de un algoritmo de cifrado simétrico o de clave privada. En el esquema tenemos una sola llave, que utiliza el mismo algoritmo para cifrar o descifrar el mensaje original.

_ Cada uno de estos candados o mecanismos con los cuales uno produce un mensaje cifrado, en base a un mensaje de texto plano, utilizan lo que llamamos algoritmos, donde hay de distintos tipos con distintas características de funcionamiento general e interno, creados por distintos criptógrafos, grupos de estudios de seguridad, ejercito, entre otros, y cada uno de estos algoritmos tienen sus particularidades. Dada a la característica de como estén formados los algoritmos los va a hacer más eficientes o deficientes, robustos o débiles, livianos o pesados, patentados (cerrados) o libre uso (abiertos). Los algoritmos más utilizados son:

DES: es un algoritmo cifrador de bloques de 64 bits. La longitud de la llave es fija y de solamente 56 bits lo que permite encontrar la misma mediante fuerza bruta en cuestión de horas. No posee patentes. Este en la actualidad ya está superado y en desuso ya que hace posible un ataque de fuerza bruta.

Triple DES: consiste en la aplicación del algoritmo DES 3 veces con 3 llaves distintas. Al emplear 3 llaves DES distintas, la longitud de la llave en TDES es fija y de 168 bits. En la práctica hay algunos problemas de superposición que hace que la longitud efectiva de la misma sea de 112 bits. No posee patentes. Tampoco es apto para el uso actual.

AES (Advanced Encryption Standard): es un algoritmo cifrador de bloques de 128 bits con una llave de longitud variable de 128, 192 o 256 bits. Deriva de un algoritmo llamado Rijndael. No posee patentes. Este se utiliza actualmente.

IDEA (International Data Encryption Algorithm): es un algoritmo cifrador de bloques de 64 bits, que utiliza una llave de longitud fija de 128 bits. Las características que tiene es que es un algoritmo sencillo, rápido y fácil de programar, solo es de utilización libre para uso no comercial.

Blowfish: este algoritmo cifrador de bloques de 64 bits se diferencia de los anteriores en que utiliza una llave de longitud variable desde 32 hasta 448 bits (en saltos de 8 bits). La longitud por defecto es de 128 bits. Es más rápido que DES e inclusive IDEA. No posee patentes.

Twofish: cifrador de bloques de 128 bits que también utiliza llaves de longitud variable de 8 a 256 bits en múltiplos de 8 bits (128 bits por defecto). No posee patentes.

CAST-256: cifrador de bloques de 128 bits que utiliza una llave de longitud variable de 128 a 256 bits en múltiplos de 32 bits. Esta patentado y es de uso libre.

RC2: cifrador de bloques de 64 bits que soporta de 0 a 1024 bits de longitud de llave (128 bits por defecto) en múltiplos de 8 bits.

RC4: cifrador de flujo con longitudes de llave 8 a 2048 bits (128 bits por defecto) en múltiplos de 8 bits. Puede ser inseguro dependiendo del uso que se le dé.

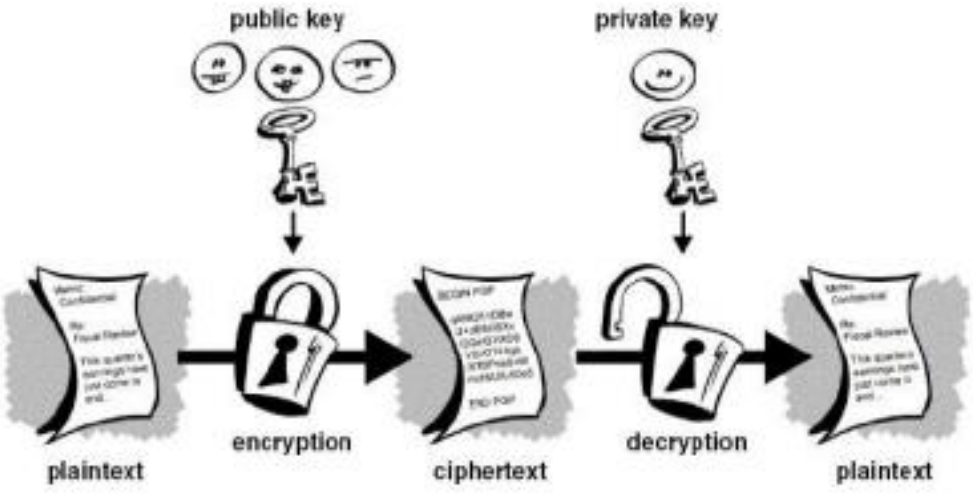
ChaCha20: cifrador de bloques de 256 bits. Hasta 3 veces más rápido que AES, utilizado por Google, OpenSSH, etc. Disponible en el dominio público.

_ Ventajas y desventajas de la criptografía simétrica:

- Ventajas: los algoritmos simétricos son más rápidos. El tamaño del mensaje cifrado es prácticamente el mismo que el del mensaje original. Ideal para cifrar información que permanece almacenada y no necesita ser transmitida.
- Desventajas: intercambio de llaves en forma segura ya que al ser la misma llave para cifrar como descifrar, el hecho de pasarle la llave al destinatario de forma segura se complica . La cantidad de llaves necesarias cuando intervienen muchas personas es n(n-1)/2 llaves para cada par de personas por ende crecen desmedidamente. No garantizan autenticidad y no repudio ya que al ser una sola llave no sabemos quién cifro y quien descifro, ya que cualquiera que tenga la llave lo hizo.

Criptografía asimétrica: se utiliza una llave para descifrar distinta de la empleada para cifrar. Ambas llaves se encuentran relacionadas matemáticamente, pero es computacionalmente imposible obtener una a partir de la otra. Lo que se cifra con una se descifra con la otra y viceversa. También se la llama criptografía de llave pública. En esta tenemos un par de llaves, una que llamamos llave publica y otra llave privada. La llave publica la puede tener cualquiera y a la privada la tenemos nosotros de forma propia, segura e intransferible, donde todo lo que se cifre con la llave publica solo puede ser descifrado con la llave privada, y si lo hacemos de forma viceversa cualquiera va a poder ver la información. Los usos son:

- Confidencialidad: utilizamos la llave pública del receptor para cifrar y luego el utilizara su propia llave privada para descifrarlo. De esta forma nos aseguramos que solo el receptor autorizado pueda acceder nuevamente al texto plano.
- Autenticación: ciframos el mensaje utilizando la llave privada del emisor y cualquiera que posea la llave publica correspondiente podrá descifrarlo. El receptor se asegura de la integridad, autenticidad y no repudio del mensaje recibido.



_ En esta criptografía, los algoritmos más utilizados son:

DH (Diffie-Hellman): fue el primer algoritmo asimétrico (1976). Es un método para intercambio de llaves que permite a dos extremos que no se conocen previamente, establecer una llave secreta compartida a través de un canal de comunicaciones inseguro. Esta llave puede ser luego utilizada para cifrar comunicaciones subsiguientes usando un algoritmo de cifrado simétrico.

RSA (Rivest, Shamir y Adelman): data del año 1977. Este algoritmo sirve tanto para cifrar como para autentificar. Permite utilizar llaves de longitud variable, pero actualmente se aconsejan llaves mayores a 1024 bits.

Elgamal: desarrollado por el Dr. Taher Elgamal en 1985. Este algoritmo se utiliza tanto para cifrar un mensaje como para firmar. El mensaje cifrado puede llegar a ocupar el doble de espacio que el mensaje original por lo que no se lo suele utilizar. No esta patentado.

ECC (Elliptic Curve Cryptography): algoritmo que utiliza puntos en una curva elíptica para definir las llaves publica/privada. Una llave de 256 bits ECC equivale a una de 3072 bits de RSA. Ideal para conexiones rápidas, seguras y con bajo uso de poder de cómputo (smartphones, tablets, etc).

_ Ventajas y desventajas de la criptografía asimétrica:

- Ventajas: los algoritmos asimétricos simplifican el intercambio de llaves en forma segura. Se garantiza la integridad, autenticidad y no repudio.
- Desventajas: son mucho más lentos y pesados que los algoritmos simétricos. Un mensaje cifrado con un algoritmo asimétrico ocupa más espacio que el original.

Tercer criterio

_ Y también de acuerdo al procesamiento del mensaje:

Cifradores de bloque: aplicamos la operación de cifrar (o descifrar) sobre bloques de tamaño fijo del mensaje. Por ejemplo, estos pueden ser bloques de 32, de 64 o de 128 bits. Algunos algoritmos son:

Electronic Code Book (ECB): los mensajes se dividen en bloques y cada uno de ellos es cifrado por separado. La desventaja de este método es que a bloques de texto plano idénticos les corresponde bloques idénticos de texto cifrado, de manera que se pueden utilizar estos patrones como guía para descubrir el texto plano a partir del texto cifrado.

Cipher Block Chaining (CBC): a cada bloque de texto plano se le aplica la operación XOR con el bloque cifrado anterior y luego es cifrado. De esta forma, cada bloque de texto cifrado depende de todo el texto plano procesado hasta este punto. Esto evita la sustitución de un bloque individual dentro del mensaje.

Cifradores de flujo: aplicamos la operación de cifrar (o descifrar) sobre cada elemento o carácter del mensaje. Normalmente por cada bit.

RC4: cifrador de flujo que soporta de 8 a 2048 bits de longitud de llave (128 bits por defecto) en múltiplos de 8 bits. Dependiendo del uso que se le dé, puede presentar algunas vulnerabilidades.

SEAL: cifrador de flujo muy veloz diseñado para equipos de 32 bits. Se encuentra patentado.

Conceptos complementarios

Función Hash: una función criptográfica de hash es un procedimiento determinístico (siempre produce el mismo resultado) que toma un bloque arbitrario de datos y mediante el procesamiento devuelve una cadena de longitud fija (en bits), llamada valor hash, de modo tal que cualquier modificación a los datos de entrada hará que el valor hash cambie. El resultado que obtenemos puede tener diferentes nombres como digest, hash, resumen, etc. Una función de Hash no tiene por objetivo proteger la información, ni la autenticidad, ni no repudio, ya que la salida es una cadena de longitud fija y lo que se busca es saber si fue modificado o no, pero no todos los otros usos de lo que uno espera de la criptografía.

_ Las propiedades de las funciones hash son:

- Unidireccionalidad: dado un resumen $H(m_i)$, o sea teniendo la salida, debe ser computacionalmente imposible encontrar m_i o la entrada, a partir de dicho resumen.
- Compresión: a partir de un mensaje de cualquier longitud, el resumen $H(m_i)$ debe tener una longitud fija. Lo normal es que la longitud de la salida $H(m_i)$ sea menor que el mensaje m_i original.
- Facilidad de cálculo: debe ser fácil calcular H(mi) a partir de un mensaje determinado m_i.
- Difusión: el resumen $H(m_i)$ debe ser una función compleja de todos los bits del mensaje m_i. Si se modifica un solo bit del mensaje m_i, el hash $H(m_i)$ debería cambiar la mitad de sus bits aproximadamente. Básicamente cualquier modificación que haya en la entrada se va a arrastrar y manifestar en la salida.

_ Las funciones o algoritmos de Hash más utilizados son:

MD2 (Message Digest 2): devuelve una salida fija de 128 bits para cualquier longitud del mensaje de entrada. La implementación de MD2 esta optimizada para equipos de 8 bits. Es el más lento y no se recomienda su utilización en nuevas implementaciones.

MD4 (Message Digest 4): devuelve una salida fija de 128 bits para cualquier longitud del mensaje de entrada. Es el más rápido de la familia. Es considerado actualmente inseguro y no se recomienda su utilización.

MD5 (Message Digest 5): también devuelve una salida de longitud fija de 128 bits. Procesa los datos de entrada en bloques de 512 bits. Es uno de los algoritmos más utilizados aunque actualmente está en duda su seguridad y no se recomienda su utilización en nuevas implementaciones.

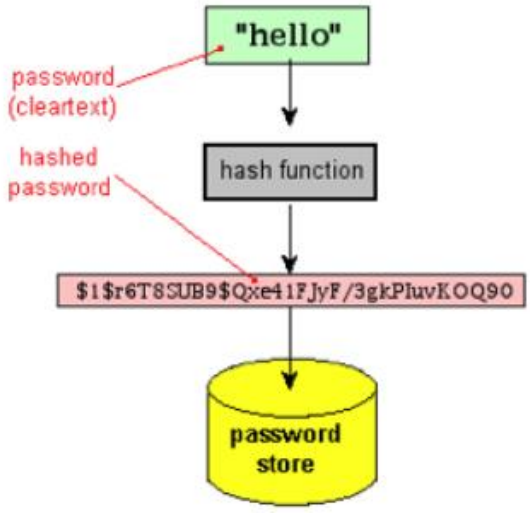
MDC2 (Modification Detection Code 2): desarrollado por IBM. Devuelve un compendio de 128 bits.

SHA-1 (Secure Hash Algorithm): es un algoritmo que se utiliza en el SHS (Secure Hash Standard) y devuelve un compendio de 160 bits. Procesa los datos de entrada en bloques de 512 bits. Actualmente su seguridad esta puesta en duda. Otros algoritmos de la misma familia son SHA-224, SHA-256, SHA-384 Y SHA-512, con salidas de 224, 256, 384 y 512 bits respectivamente.

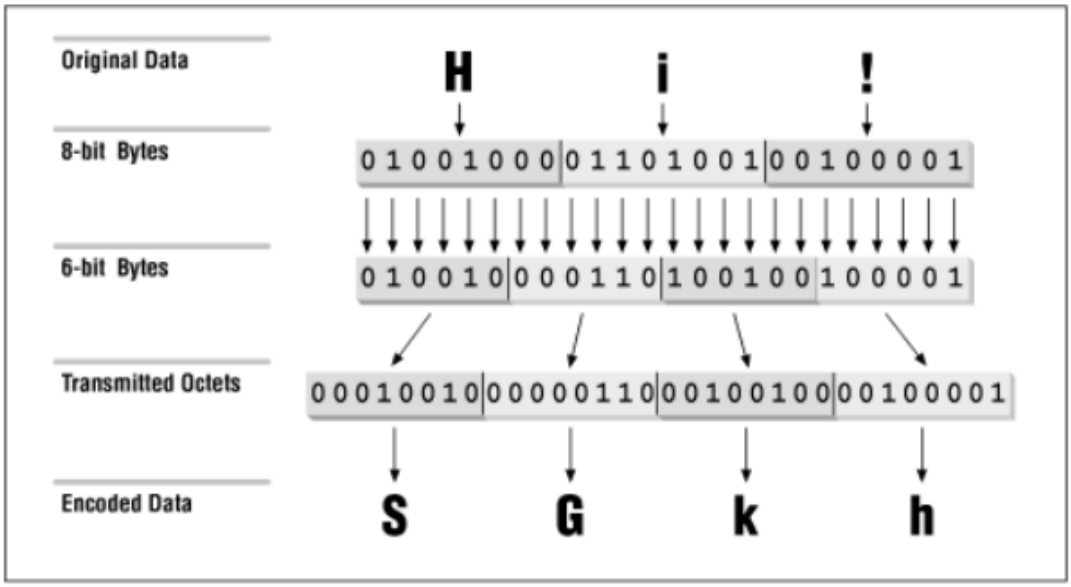
RIPEMD-160 (RACE Integrity Primitives Evaluation Message Digest): este es un algoritmo europeo que devuelve una salida de 160 bits. Existe otras versiones de este algoritmo de 128, 256 y 320 bits. RIPEMD-160 es más lento que SHA-1. No posee ninguna patente.

Tiger: devuelve una salida de 192 bits. Este algoritmo está diseñado para ser utilizado en plataformas de 64 bits.

_ Las funciones de Hash se pueden utilizar para evitar que una información sea descubierta o deducida dada la propiedad de que estas unidireccional, es decir, desde la entrada se produce la salida pero no se puede hacer el proceso inverso, y eso hace que sea muy adecuado para representar contraseñas en sistemas operativos, bases de datos, aplicaciones, etc. Esto funciona de la siguiente manera, el usuario coloca la contraseña en texto plano, a esa contraseña se le calcula la correspondiente función de Hash y eso es lo que se almacena en el disco, es decir, nunca se almacena la información en texto plano sino que se almacena el Hash de esa información. Y para saber si el usuario puso bien la contraseña, tomamos lo que puso el mismo y le calculamos el Hash, luego comparamos el Hash calculado con el que tenemos guardado, en donde si son iguales entonces significa que puso bien la contraseña.



Codificación Base64: esto no es criptografía, no es funciones de hash, esto es un esquema de codificación para datos binarios que puede ser representado usando únicamente los caracteres imprimibles de ASCII. El funcionamiento de esta consiste en tomar la información original, que viene en bloques de bytes (8 bits) y se toma de a seis bits la información original, donde luego cada grupo se representan en el octeto correspondiente del carácter ASCII. De este modo traducimos la información binaria en información en caracteres ASCII o de texto. Cualquiera que tenga información producida con base64 puede decodificarla y obtener el mensaje original, pero lo que se busca acá no es ocultar nada, simplemente representar de una forma distinta la información.



Implementación

OpenSSL: es un proyecto colaborativo para desarrollar un kit open source de herramientas robusto, completo y de calidad empresarial que implemente los protocolos SSL (Secure Socket Layer) y TLS (Transport Layer Security) y un conjunto de librerías criptográficas de propósito general. Este kit sirve para la gestión y manipulación de criptografía. Los algoritmos y funciones implementados son:

- Algoritmos simétricos: AES128, AES192, AES256, Blowfish CAST, CAST5, DES, 3DES IDEA (Patentado), RC2, RC4, RC5 (Patentado).
- Funciones de HASH: MD2, MD5, MDC2 (Patentado), SHA, SHA-1, RIPEMD-160 .

_ Algunos ejemplos, la herramienta base es openssl pero dentro de esta le indicamos que tipo de operación vamos a realizar. Con *enc* de encript indicamos que vamos a realizar una operación de cifrado, luego tenemos en cuenta que *-e* es de encriptar, *-d* es de desencriptar, *-in* es de input e indicamos el archivo, *-out* es donde genera la salida, y finalmente colocamos el algoritmos que vamos a usar. Al ejecutarlo debemos poner la password.

- Cifrar un archivo utilizando el algoritmo simétrico DES:
  $ openssl enc -e -in mensaje.txt *-out* mensaje.des *-des*

- Cifrar un archivo utilizando el algoritmo simétrico AES128:
  $ openssl enc -e -in mensaje.txt *-out* mensaje.aes *-aes128*

- Descifrar un archivo cifrado con el algoritmo DES:
  $ openssl enc -d -in mensaje.des *-out* mensaje.txt *-des*

- Calcular el resumen (Hash) de un archivo empleando el algoritmo MD5:
  $ openssl dgst -md5 mensaje.txt

- Codificar en base64 un archivo:
  $ openssl base64 -in mensaje.txt

_ Si ponemos *cat* en un archivo cifrado, vemos caracteres binarios.

_ Para comparar dos archivos uno encriptado y el original conviene usar el comando du para saber cuánto pesan y no wc porque el archivo encriptado no tiene líneas.

_ Con la opción *-pass pass:*somepassword, donde es el comando *-pass* y luego *pass:* con la contraseña que queremos poner. Pero con *-passin pass:*somepassword damos una contraseña para el encriptado y con *-passout pass:*somepassword damos una contraseña para el desencriptado.

Firmas y certificados digitales

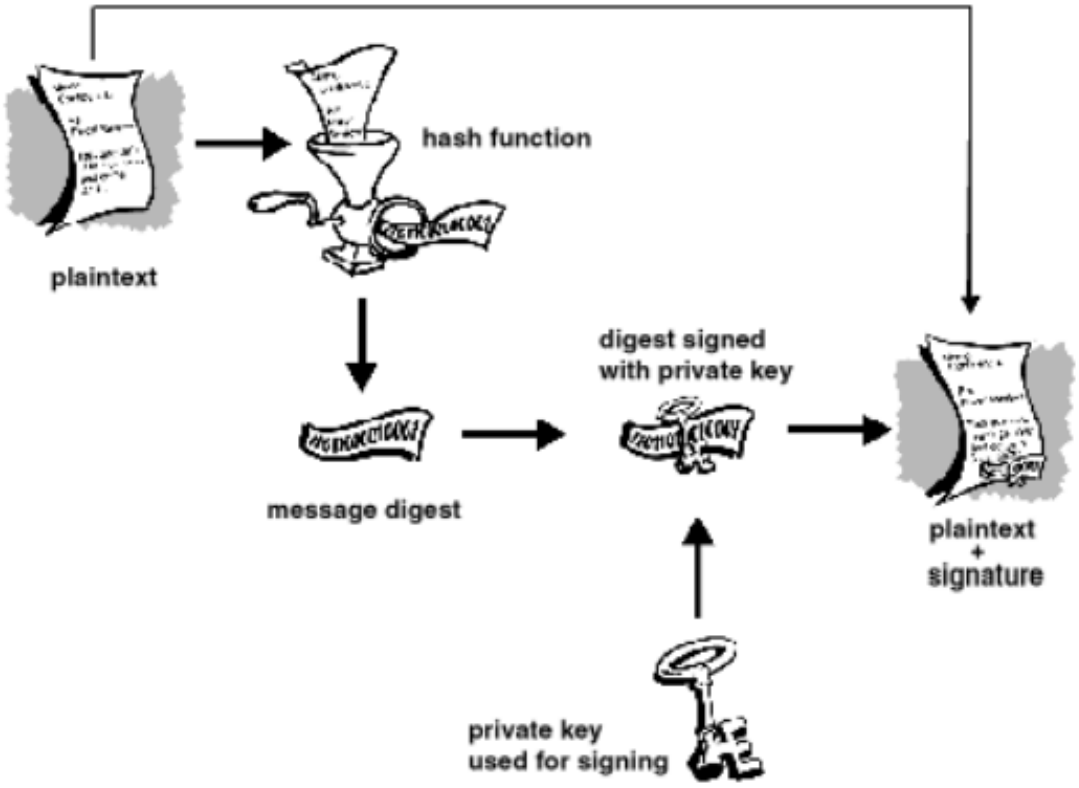
Firmas digitales

Firma digital: es un esquema criptográfico que sirve para demostrar y asegurar la autenticidad e integridad de un mensaje digital o documento electrónico. Mediante esta se asegura:

- La integridad del documento, es decir, que no haya sido alterado.
- La autenticidad o identidad del emisor, que es una forma univoca e irrefutable para demostrar quien ha sido el emisor.
- Y garantiza el no-repudio, porque como sabemos quién fue el emisor, el emisor no podrá aducir que no fue el quien genero esa firma.

_ Los procedimientos del emisor son bastantes simples y combina los conceptos de criptografía asimétrica y funciones de Hash. Entonces, imaginamos una situación en la que tenemos que enviar un documento y queremos firmarlo:

1. Partimos de un documento de texto plano, y a ese se le calcula mediante una función de hash el resumen del documento digital original.
2. A este resumen se lo cifra con la llave privada del emisor que no la comparte con nadie. Donde solo la llave publica correspondiente al par de llaves será el único que puede descifrar esto.
3. Entonces el resultado es la firma digital, y esta se adjunta al final del documento produciendo el documento firmado digitalmente. Donde lo que se envía es el documento más el Hash cifrado (firma) al final del documento.



_ Ahora, los procedimientos del receptor son es lo inverso:

1. Separamos el documento digital de la firma. O sea por un lado el mensaje de texto, que es texto plano que no está cifrado, lo podemos ver y es accesible para todos, y por otro lado la firma que es el Hash cifrado.
2. Calculamos nuevamente el resumen del documento digital con el mismo algoritmo empleado por el emisor.
3. Desciframos la firma digital empleando la llave pública del emisor para extraer el resumen o Hash calculado originalmente.
4. Y por último comparamos ambos resúmenes o Hashes, donde sí coinciden quiere decir que la firma es válida y que el documento no ha sido alterado.

_ Entonces mediante este esquema podemos asegurarnos de que es válida la firma que viene en el documento. No es necesario que sea un documento de texto, sino que puede ser una archivo, un flujo de datos, etc, donde cualquier información puede ser sometida a este procedimiento para calcular la firma.

Algoritmos

_ Dentro de lo que es firma digital, tenemos diferentes algoritmos o formas de implementarlos. Los más utilizados son:

DSA (Digital Signature Algorithm): basado en el esquema de firma de Elgamal, este algoritmo, junto con SHA, es el núcleo del DSS (Digital Signature Standard) y solo se utiliza para firmar digitalmente un mensaje, no para cifrarlo. Además provee la capacidad de verificar una firma digital. Esta patentado pero es de libre uso.

ECDSA (Elliptic Curve DSA): es una variante del algoritmo DSA pero basado en el problema de las curvas elípticas. Esta variante requiere tamaños de llaves de menor longitud para ofrecer el mismo nivel de seguridad. El tiempo de procesamiento es similar al DSA.

Observaciones

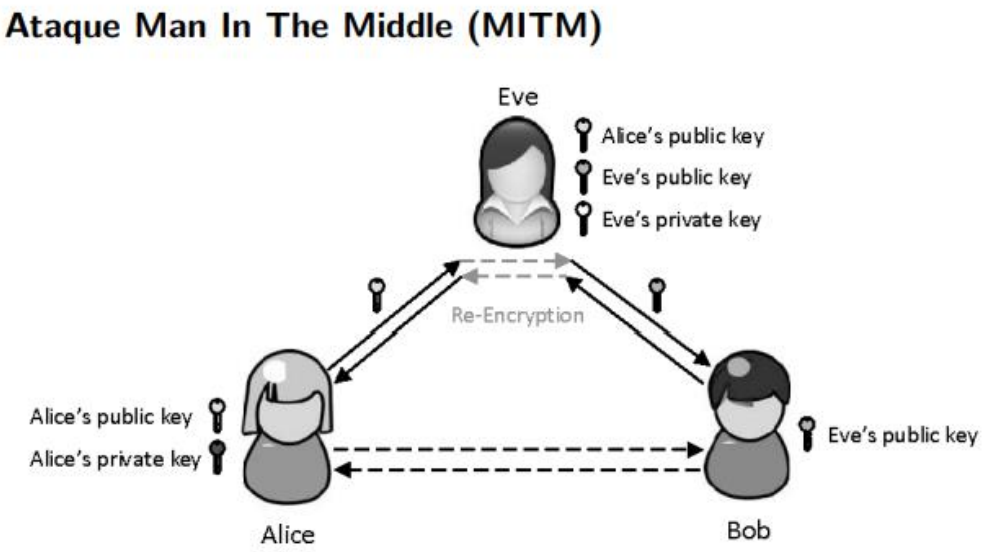
_ Tenemos que entender que firmar digitalmente NO es cifrar, lo que firmar hace es garantizar la autenticidad e integridad del documento. Podemos contar escenarios donde un documento puede estar firmado digitalmente y cifrado, puede estar firmado y no cifrado, o puede estar cifrado y no firmado. Al cifrar algo estamos garantizando la confidencialidad, y esto no es el objetivo ni la función de la firma digital. Entonces podemos jugar con estas combinaciones dependiendo de lo que necesitemos.

Problemas llave publica/privada

_ Utilizamos la criptografía asimétrica que tiene una llave privada y otra publica para cifrar el hash del documento. La seguridad de este esquema va a estar basado en que tengamos la llave pública del emisor, porque si hubiera alguien que pudiera de algún modo intervenir, engañarnos y hacernos creer que la llave publica es la del emisor, podría darse lo que se llama un ataque de hombre en el medio (man in the midle o MITM), este ataque lo estamos viendo en criptografía pero lo podemos ver en distintos rubros o conexión en los que alguien logre interceptarla y meterse en el medio para hace creer que ambos extremos están dialogando entre sí pero en realidad este actor que está en el medio está viendo toda la comunicación.

Ataque man in the middle: para explicar el ataque en nuestro contexto suponemos que Alice con su par de llaves desea interactuar con Bob, por ejemplo enviándole un documento firmado o cifrado. Es muy importante que Bob tenga la llave publica de Alice. Pero si un tercer actor, en este caso Eve, se mete en el medio y engaña de algún modo a Bob dándole su llave publica haciéndole cree que es la llave de Alice, esta persona lograría

que todo lo que cifre Bob en realidad va a estar cifrado con la llave publica de Eve, con lo cual al tener la llave privada va a poder verlo, y luego lo cifra con la llave de Alice y se la envía, donde Alice no se entera de que alguien leyó la información en el medio. Si el actor del medio mediante un ataque phishing logra por ejemplo modificar el sitio o la forma en la que Alice publica su llave publica y mete la suya, entonces va a lograr este tipo de ataque o intervención en la que va a poder acceder a toda la información e incluso modificarla si quisiera y mandarla al destinatario sin que este se entere.



_ Explicado de otra forma, suponemos que Alice le quiere mandar un documento cifrado a Bob, entonces Alice le avisa a Bob que le va a mandar el documento cifrado y le dice que se baje su llave pública. Si Eve de algún modo logra engañar a Bob, haciéndole creer que determinada llave pública, es decir, la llave de Eve, es en realidad la llave de Alice, entonces Bob va a utilizar la llave de Eve y va a estar cifrando con esa en vez de cifrar con la de Alice, por lo que al estar usando la llave pública del atacante, lo que Bob cifre va a poder ser descifrado por Eve con su llave privada y luego Eve va a usar la llave publica de Alice para enviarle el mensaje. Entonces el atacante se pone en el medio, intercepta y desencripta lo que envía Bob, lo lee y modifica, luego lo cifra con la llave de Alice y se lo manda a ella sin que se entere que paso todo esto en el medio.

_ Esta es un dificultad que tiene en general la criptografía asimétrica para este tipo de escenario, por ende había que crear un mecanismo o una forma mediante la cual se pudiera asegurar que una llave publica determinada pertenezca a una entidad, persona, empresa, etc, y para eso surge lo que son los certificados digitales.

Certificados digitales

Certificado digital: es un documento electrónico que básicamente usa una firma digital para vincular una llave publica con una identidad (el nombre de una persona física, una organización, software, etc.). Entonces a través del certificado se puede utilizar para verificar y garantizar que una llave publica pertenece a un individuo y de esta manera eliminamos el problema del ataque man in the middle.

_ A continuación vemos como se esquematiza el certificado, en donde “yo”, es decir, tal autoridad de certificación entrega tal llave pública, y por otro lado ese documento esa firmado por la misma autoridad de certificación.

Autoridad de certificación (CA): es una entidad u organización encargada de emitir certificados digitales de acuerdo a determinadas políticas, procedimientos y algoritmos criptográficos, certificando así la autenticidad y validez de las llaves públicas. Esta entidad sigue un montón de procedimientos, normas, algoritmos y mecanismos de seguridad para garantizar la autenticidad y validez de las llaves que se certifican. Una autoridad de certificación certifica la validez de los documentos que emite mediante la firma digital, y para controla la validez de una firma digital necesitamos la llave pública del emisor. Entonces mediante la llave publica de la autoridad de certificación es que podemos verificar que los certificados son válidos.

- La CA es un tercero en el que confían tanto el sujeto (dueño) del certificado como quien lo utiliza luego.
- La confianza en la CA se basa en contar con su llave publica, la cual debe ser obtenida de manera segura.
- La llave publica de la CA se suele distribuir como un certificado digital autofirmado o mediante estructuras jerárquicas de autoridades de certificación.

_ Existen un gran número de entidades de certificación, por ejemplo Google, Taubate, Comodo, etc, en el que siguen un gran número de protocolos para asegurar esto. Para generar certificados se suelen pedir datos o validaciones como DNI, pasaporte, impuesto o servicio donde figure nuestra dirección, validación del correo electrónico, validación

mediante un llamado electrónico, es decir, se toman un montón de estos recaudos para asegurarse de que la identidad sea realmente válida.

_ Tercerizamos la confianza en los certificados que estas entidades emiten y para confiar en esta es fundamental tener el certificado con la llave pública, para obtener la llave de forma segura básicamente todos los navegadores tienen un pack de certificados de todas las autoridades de certificación en los que ya están grabadas las llaves públicas, con los cuales el navegador luego al interactuar con sitios que usan certificados emitidos por estas autoridades, ya puede verificar automáticamente la validez. Lo mismo sucede en los sistemas operativos.

Estructura certificados digitales

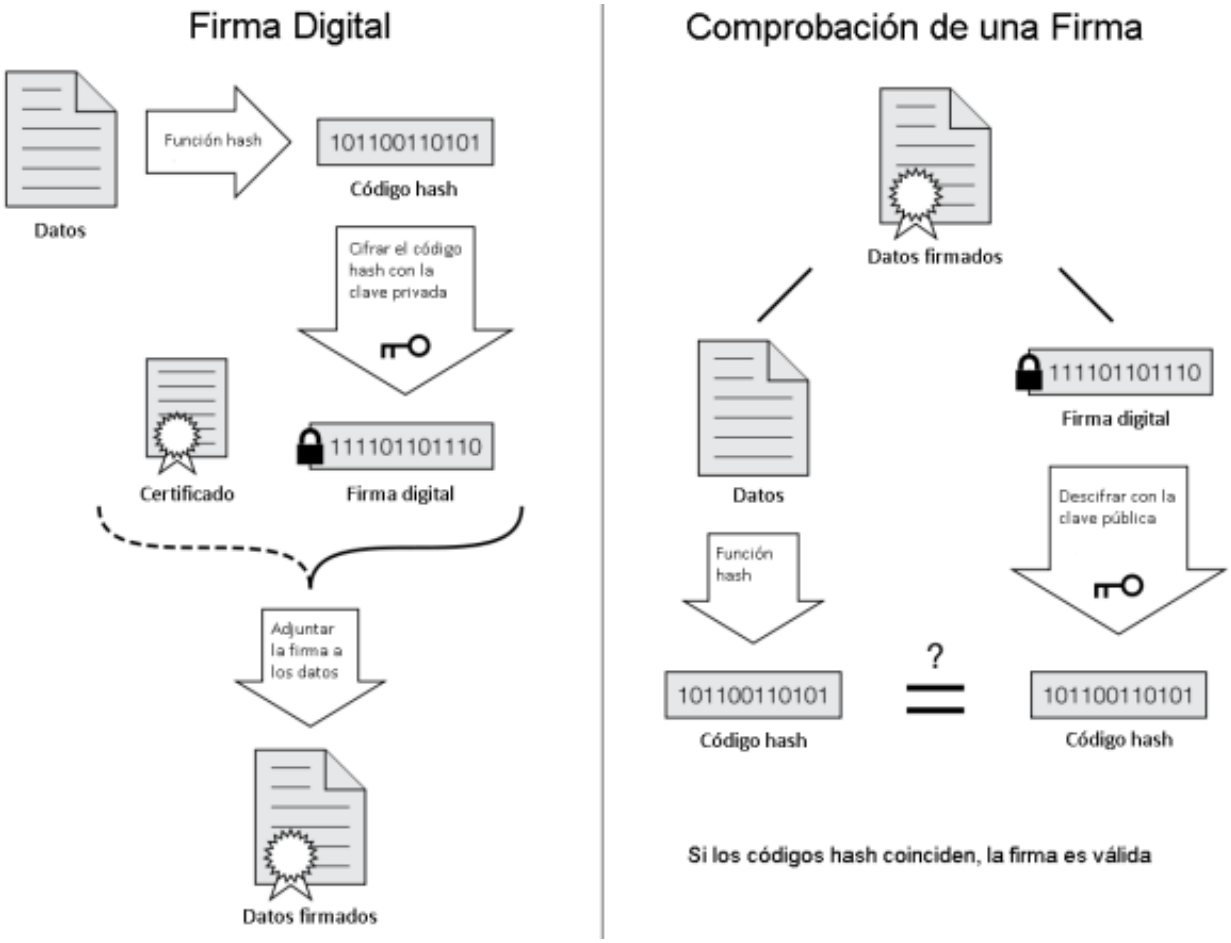
Estándar X.509: este es un estándar de ITU-T para el manejo y gestión de una infraestructura de llave pública (PKI), y define como debe ser el contenido y el formato de un certificado digital con algunos componentes como:

- Versión: puede haber distintas versiones sucesivas que vaya haciendo el certificado.
- Serial number
- Algorithm ID: el algoritmo que se usó para la firma, se usa uno u otro dependiendo la conveniencia.
- Issuer: es emisor, ya sea una empresa u organización que lo genero.
- Validity: la validez del certificado, haciendo referencia a la fecha de validez justamente (not before, not after).
- Subject: este es uno de los componentes centrales y es la identidad de la persona, empresa o cosa que se va a asociar con la llave pública. Información sobre el sujeto.
- Subject public key info: dentro de la llave publica se especifica cual fue el algoritmo (public key algorithm) que se utiliza y la llave propiamente dicha expresada en hexadecimal.
- Certificate signature algorithm: finalmente se dice que algoritmo se utilizó para la firma digital de tal documento
- Certificate signature: se firma al final de todo.

_ Todos estos campos son importantes porque a través de estos podemos generar ataques o descubrir que nos quisieron atacar vulnerando o falsificando alguno de estos puntos o podemos tener problemas con algo que no funcione por ejemplo que haya expirado la fecha de validez o por otros motivos.

Usos de certificados digitales

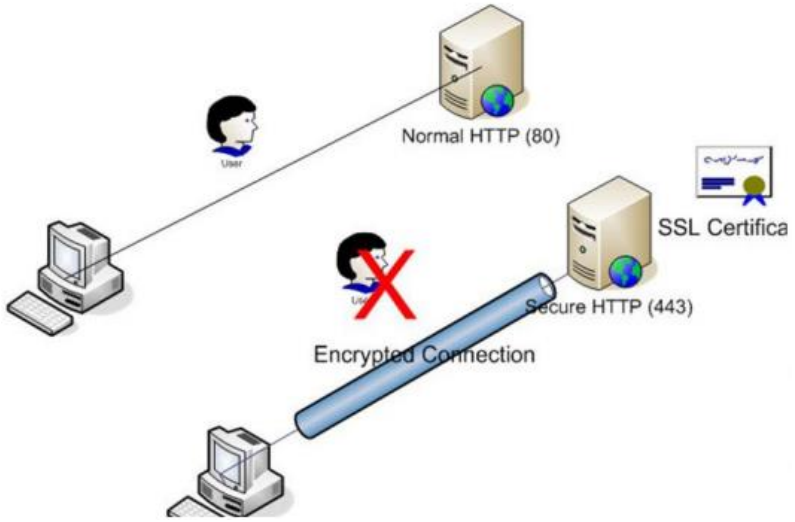
Verificación de la firma digital: el receptor del mensaje verificar a el certificado usando la llave publica de la CA, y, teniendo confianza en la llave pública del remitente, verificar a la firma del mensaje.



_ Analizando la parte de la firma digital, se toman los datos, se calcula un Hash, donde luego a ese Hash se cifra con la llave privada y se envía el documento junto con el certificado y con la firma, entonces el documento resultante va a contener el texto original, la firma y el certificado. Ahora en la parte de la comprobación de la firma, quien recibe esto separa por un lado el texto original en texto plano y por el otro toma el certificado original y valida contra la autoridad de certificación que corresponda (tercero que estaría arriba garantizando tanto al emisor como al receptor que la llave publica que utilicen va a ser válida), luego a la firma la desciframos con la llave publica que va a estar dentro del certificado obteniendo así el Hash, y por otro lado calculamos un Hash al documento para finalmente comparar ambos Hashes para ver si son iguales y así la firma será válida. De esta forma eliminaríamos el ataque de man in the middle.

Autenticación y confidencialidad: esto de los certificados tienen infinidad de usos, los usamos todos los días muchas veces sin saberlo, pero con el simple hecho de usar internet o distintos softwares que van validando comunicaciones o que la información no fue adulterada esto se usa todos los días mediante los certificados y las autoridades de certificación. Por ejemplo, mediante la instalación de un certificado digital en un servidor web, los clientes que acceden al sitio pueden verificar su autenticidad y cifrar la conexión. Este es el ejemplo típico donde la comunicación cifrada que realizamos mediante HTTPS con un navegador y un servidor web. El procedimiento es el siguiente:

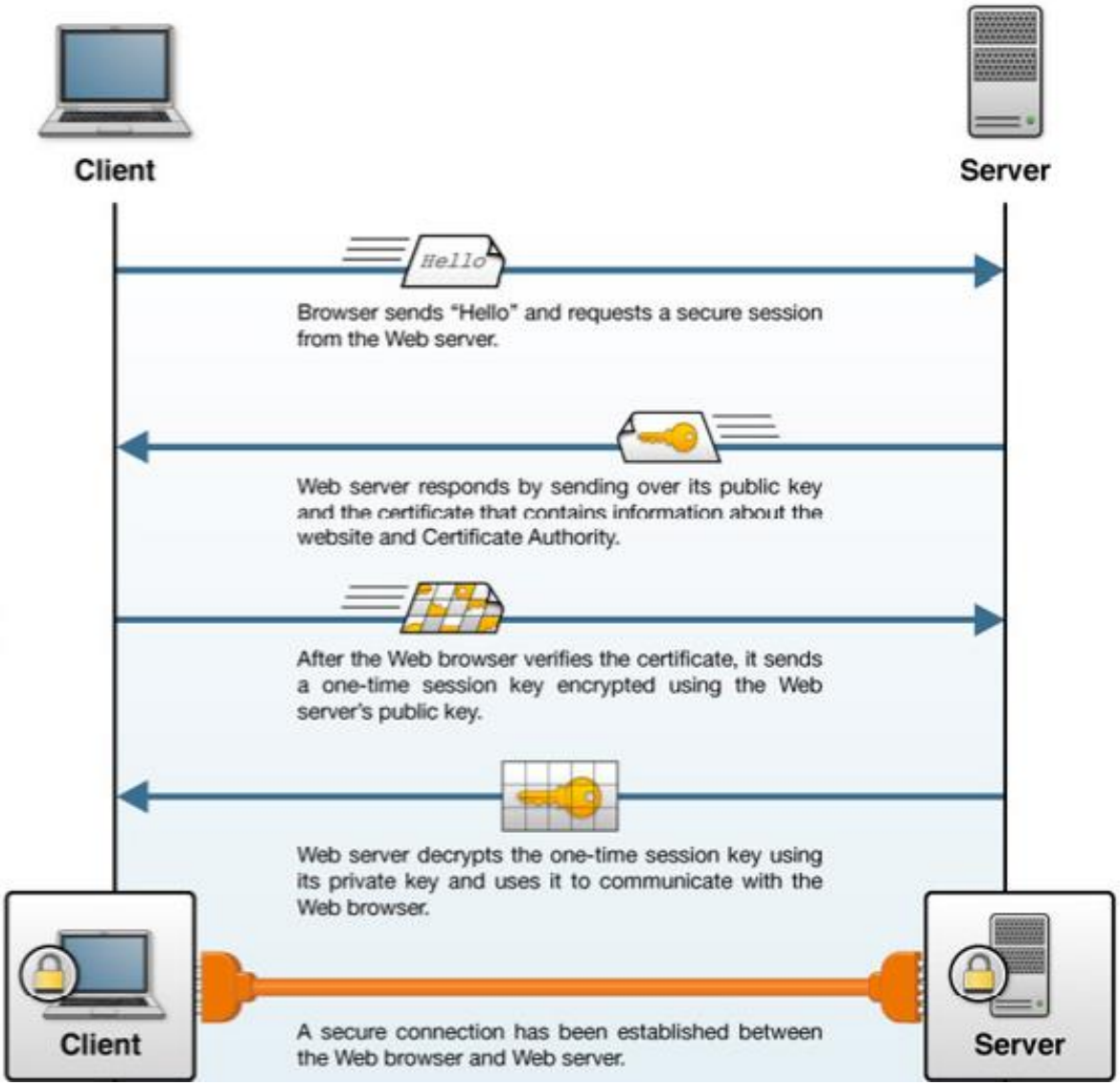
- HTTP: Cuando nos conectamos a un servicio web en forma de texto plano con HTTP en el puerto 80, esa comunicación va en texto plano y cualquiera que la interfiera va a poder ver todo lo que hay en esa comunicación, no solo información sino también contraseñas y demás.
- HTTPS: para solucionar el problema de HTTP usamos la comunicación cifrada con HTTPS en el puerto 443 el cual establece un túnel entre el cliente y el servidor y evita que alguien en el medio pueda vero o interpretar lo que va por dentro de la conexión.



_ A continuación vemos el esquema paso a paso de cómo funciona esto:

1. El cliente inicia la conexión y manda un saludo para comenzar la comunicación.
2. El servidor responde con un certificado digital donde va a estar contenida la llave pública del servidor.
3. El cliente verifica la autenticidad del certificado mediante la autoridad de certificación correspondiente y por ende del servidor. De este modo obtiene la llave publica de una forma segura. Hasta este paso se utiliza la criptografía asimétrica, es decir, para intercambiar las llaves ya que es una dificultad de las simétricas.

4. Luego de realizada la autenticación del servidor, el cliente genera una llave de criptografía simétrica de sesión aleatoria y la envía cifrada con la llave pública del certificado y la envía al servidor.
5. El servidor recibe la llave de sesión cifrada con su llave pública y la descifra empleando su llave privada. A partir de ese momento ambos extremos ya tienen una misma llave simétrica.
6. El cliente y servidor pueden desde este punto intercambiar información cifrada empleando un algoritmo simétrico porque tiene la ventaja de que la información cifrada ocupa el mismo espacio que la original. Con ello se asegura la confidencialidad e integridad de la información transmitida.
7. Se logra un esquema hibrido donde se combinan ambas criptografías



Herramientas

GnuPG

_ Tenemos GnuPG que es un set de herramientas que se diseñó para producir distintos tipos de cifrado con distintos algoritmos y sobre todo para gestionar llaves, para tener lo que se llama un anillo de llaves (keyring), con lo cual uno puede ir cargando llaves de usuarios o las propias y utilizarlos para enviar información a distintos usuarios con la llave correspondiente.

PGP: fue el primero que surgió y es un programa diseñado para proteger la información enviada a través de una red pública mediante el uso de criptografía simétrica y asimétrica.

GnuPG: reemplazo libre de PGP que puede ser utilizado sin restricciones y es compatible y cumple con el estándar OpenPGP (RFC 2440). Este además:

- Proporciona facilidades para la gestión de las llaves públicas y privadas, mediante la utilización de un "anillo de llaves" o keyring.
- Soporta entre otros los siguientes algoritmos: ElGamal, DSA, RSA, AES, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 y TIGER.

_ Esta herramienta nos permite generar las llaves, verlas, importar llaves de otros, permite cifrar, enviar y recibir llaves. Algunos ejemplos de manejo de llaves:

- Generar par de llaves pública y privada. El ejecutar este comando elegimos que algoritmo usar por default es RSA, luego pregunta la longitud de la llave, luego una fecha de validez, nuestros datos, y una confirmación. Luego pide una passfrase (contraseña para poder abrir la llave privada). Una vez generada en el home se guarda toda la información de esta:
  gpg –full-gen-key

- Listar llaves disponibles en el keyring:
  gpg --list-public-keys (muestra las llaves públicas)
  gpg --list-secret-keys (muestra características de las llaves privadas)

- Exportar llave pública. Armor manda la salida binaria en archivo de texto en base64, ya que la clave publica esta codificada en base64:
  gpg --armor --output usuario.asc *--export* \ usuario@dominio.com

- Para ver el contenido del archivo usuario.asc:
  cat usuario.asc

- Importar llave publica que alguien me mando para incorporar en el anillo de llaves:
  gpg --import usuario.asc

- Editar keyring para confiar en llaves publicas importadas:
  gpg --edit-key usuario@dominio.com

- Buscar llave publica en servidor de llaves:
  gpg --keyserver hkp://keys.gnupg.net:80 *--search-keys* 00411886

- Enviar llave publica a servidor de llaves para alojar las llaves:
  gpg --keyserver hkp://keys.gnupg.net:80 *--send-keys* 9D2FFAC8

- Importar llave publica de terceros desde servidor de llaves:
  gpg --keyserver hkp://keys.gnupg.net:80 *--recv-keys* 00411886

- Cifrar un archivo utilizando nuestra llave publica:
  gpg --recipient usuario@dominio.com *--output* \ archivo.txt.gpg *--encrypt* archivo.txt

- Descifrar archivo:
  gpg --output archivo.txt *--decrypt* archivo.txt.gpg

- Cifrar archivo utilizando un algoritmo simétrico (por defecto AES o usando –cypher-algo):
  gpg --output archivo.txt.gpg *--symmetric* archivo.txt

- Cifrar un archivo para ser enviado a otro usuario en Base64:
  gpg --armor --recipient usuario@dominio.com *--output* \ archivo.txt.asc *--encrypt* archivo.txt

- Cifrar un archivo para ser enviado a otro usuario en formato binario:
  gpg --recipient usuario@dominio.com *--output* \ archivo.txt.gpg *--encrypt* archivo.txt

- Descifrar un archivo cifrado para nosotros:
  gpg --decrypt-files archivo.txt.asc

- Firmar digitalmente un archivo:
  gpg --local-user usuario@dominio.com *--clearsign* archivo.txt

- Verificar un mensaje firmado digitalmente:
  gpg --verify archivo.txt.asc

- Cifrar y firmar al mismo tiempo:
  gpg --local-user usuario@dominio.com *--recipient*\ maliaga@uccor.edu.ar *--armor --sign --output* archivo.asc\ *--encrypt* archivo.txt

Google Hacking

_ Veremos las técnicas de reconocimiento que son como los deberes que hace un hacker o atacante, previo a realizar sus actividades, con el objetivo de recaudar toda la información posible sobre el target o el objetivo que quiere atacar. Hay distintas formas, con distintas herramientas, como técnicas de baja tecnología, técnicas que usan herramientas de software y demás, pero la primera que veremos es Google Hacking.

Conceptos

Google Hacking: es una técnica que utiliza una herramienta que siempre está disponible y a nuestro alcance además de ser gratuita y muy poderosa que es el motor de búsquedas y otras aplicaciones de Google para encontrar fallas de seguridad en la configuración y código que utilizan las páginas web. Entonces mediante esto y conociendo algunas características del objetivo que queremos buscar podemos hacer búsquedas muy precisas y avanzadas que devuelvan resultados sobre posibles objetivos. Para usar el motor de búsqueda de Google, este consta con varios elementos:

- Google bots: programas que barren o escanean la web permanentemente recopilando información, sobre páginas, enlaces, etc, indexándola y almacenándola en los servicios de Google.
- Google índex: con la información recopilada por los bots, se crea un gigantesco índice (inicialmente con 100 PB pero ahora está superado) o base que es el utilizado luego en las búsquedas.
- Google cache: los bots recopilan además el texto de las páginas y metadatos, lo que se almacena en el cache de Google para futuras búsquedas.
- Google API: existen métodos para que programas realicen búsquedas en formato XML mediante el protocolo SOAP, con el cual podemos tirar consultas en vez de hacerlo de forma manual en Google.

Operadores de búsqueda

_ Podemos usar dentro de la búsqueda una serie de operadores de búsqueda, que pueden ser símbolos o palabras claves con los cuales indicamos de una forma más precisa que cosa concreta estamos buscando:

“ ” (frase completa): colocando una frase entre comillas se buscan los sitios que la contengan textualmente.

- (signo menos): si anteponemos un "-" a un término u operador de búsqueda, los resultados que coincidan serán excluidos. Se excluye un término.

+ (signo más): si anteponemos un "+" la búsqueda devuelve resultados que tengan textualmente la palabra buscada. Se fuerza la aparición de un término.

OR: permite buscar un contenido u otro, separándolo con este operador.

_ Una página contiene distintos bloques o secciones, y uno puede específicamente buscar dentro de esas secciones con determinados parámetros. Por ejemplo vemos que tenemos un encabezado, el cuerpo de una página, tenemos textos, links, índices de la página, entonces en base a toda esta información nosotros podemos buscar dentro del body, encabezado, etc, utilizando distintos operadores.

```
<!doctype html>
<html lang="en">
        <head>
                <meta charset="utf-8">
                <title>hello, world!</title>
        </head>
        <body>
                <h1> hello, world!</h1>
                <p>My first web page</p>
        </body>
</html>
```

_ A continuación vemos más operadores, en donde cuando tenemos el prefijo *all* en algunos operadores, significa que todas las palabras tienen que estar en la búsqueda que se hace. Se recomienda no dejar espacio entre los dos puntos del operador y la palabra o frase que estemos buscando. Por convención conviene poner los operadores y no dejar espacio:

allintext: se restringen los resultados a aquellos que contengan los términos buscados en el "texto de la página". Todas las palabras tienen que estar en el texto de la página en este caso. Ejemplo:

*allintext:*relojes Rolex

allintitle: se acota la búsqueda a los resultados que contengan las palabras buscadas en el "título de la página". Es decir buscamos en el título de la página que es lo que nos figura en la ventana del navegador.

allinurl: la búsqueda muestra los resultados que contengan los términos buscados en la "URL". Podemos buscar determinados patrones y buscar de este modo lo que está en la dirección. Ejemplo:

*allinurl:*google faq -> www.google.com/help/faq.html

cache: muestra la versión guardada en el cache de Google de la URL solicitada. Buscamos un determinado patrón de palabras dentro del cache de Google y de ese modo podemos encontrar información antigua. Ejemplo:

*cache:*www.ietf.org

_ Tenemos otros operadores que no son tan útiles para las búsquedas de seguridad pero si para hacer un uso avanzado de la herramienta:

define: Google muestra definiciones del término buscado que figuren en distintas páginas web. Es decir, muestra sitios con definiciones de la palabra o termino buscado. Ejemplo:

*define:*blog

filetype: restringe la búsqueda a páginas que terminen con la extensión buscada. Es decir, devuelve solamente archivos de un tipo que estoy buscando. Ejemplo:

hp proliant g9 *filetype:*pdf

in: convierte dos unidades equivalentes, es decir, es un conversor de unidades. Ejemplo:

70 mph *in* kmh

intext: restringe los resultados a los documentos que contienen la palabra buscada en el texto de la página. Este es lo mismo que allintext pero ya sin el prefijo *all*.

intitle: muestra los documentos que contienen el termino buscado en el título.

inurl: restringe los resultados a documentos que contengan la palabra en la URL. Para acotar la búsqueda en determinadas palabras . Ejemplo:

*inurl:*indexFrame.shtml Axis

location: muestra artículos de Google News de la ubicación solicitada. Ejemplo:

messi *location:*españa

related: muestra paginas relacionadas con la URL proporcionada. Ejemplo:

*related:*despegar.com

site: este es muy útil ya que la búsqueda se acota al sitio especificado. Ejemplo:

fiat *site:*lavoz.com.ar

_ El parámetro de búsqueda que uno pone en el formulario de Google, que son los operadores anteriores, el conjunto de estos operadores se llaman dorcks. Si encontramos un sitio con un acceso fácil, podemos ver información detallada en el sitio Ipinfo.io/(ip), donde alguien puede detectar vulnerabilidades y usar el sitio, por ejemplo el acceso a una cámara de algún lugar, como un trampolín y desde ahí lanzar algún ataque por ejemplo a la red interna de esa cámara. La culpa no es de uno, ya que hacemos una búsqueda libre y gratuita, sino la del otro que no tomo los recaudos ni las medidas de seguridad para asegurar su sitio, red, dispositivo, etc.

Google Hacking Database (GHDB): es una fuente autorizada para consultar el alcance del motor de búsqueda de Google. En esta podemos encontrar términos de búsqueda para archivos que contienen nombres de usuario, servidores vulnerables e incluso archivos que contienen contraseñas. Esta tiene una colección de lo que llamamos dorcks que son el conjunto de los operadores de búsqueda de Google. Entonces, este es un sitio donde distintos usuarios suben las búsquedas que han creado y las dan ya listas y categorizadas, entonces cómodamente podemos buscar por ejemplo un dorck que nos lleve a una cámara de vigilancia. La url es: https://www.exploit-db.com/google-hacking-database

Técnicas de reconocimiento

_ Además de Google Hacking existen otras técnicas de reconocimiento, donde utilizamos otros métodos, tecnologías o procedimientos.

Reconocimiento

Reconocimiento: básicamente consiste en investigar al objetivo de ataque usando información públicamente disponible. Hay distintas formas:

- Reconocimiento de baja tecnología: utilizan ingeniería social, acceso físico, hurgar en la basura, es decir, utilizan la interacción humana, ver documentos en papel o estudiar cómo actúa determinada organización para luego entender vulnerabilidades.
- Búsquedas en Internet (Search The Fucking Web - STFW): donde tenemos Google hacking, bases de datos whois, búsquedas DNS y bases de datos de vulnerabilidades. En estas ya utilizamos una cierta tecnología.

Reconocimiento de baja tecnología

Ingeniería social: consiste en explotar el eslabón más débil o las debilidades del “elemento humano” de los sistemas de información, utilizando engaños en el usuario para obtener información confidencial que no nos debería dar. Según Kevin Mitnick se basa en estos cuatro principios:

1. Todos queremos ayudar, es decir, tenemos una predisposición de querer ayudar al otro si vemos que la necesita.
2. La primera reacción es siempre de confianza hacia el otro, donde la gente tiende a ser confiada y de entrada confía en gente que por ahí no conoce.
3. No nos gusta decir No, es más difícil decir que no que por ahí decir que si y ceder con algunas cosas.
4. A todos nos gusta que nos halaguen, donde en base a elogios o hacer sentir bien al otro, lograr determinadas cosas.

_ Un ejemplo de ingeniería social seria por ejemplo llamar a una persona y hacerse pasar por Fibertel, diciendo que hay problemas con su conexión y que necesitamos resetear su modem, para eso le decimos que para hacerlo su contraseña es 123, a lo que la otra persona corrige y dice que es 456, y así obtenemos acceso. Con engaños a veces se logra obtener información, otro ejemplo es phishing con los mails de los bancos pidiendo cambiar la clave de seguridad, por lo que debemos estar atento. Entonces todo aquello que conlleve a engañar de algún modo al cliente mediante distintas técnicas por distintos medios o tecnologías, es decir, todo lo que sea vulnerable al factor humano y encontrar la vuelta para engañar al otro es ingeniería social.

Acceso físico: consiste en obtener acceso físico a un equipo en la red o donde sea. Ya teniendo acceso el atacante puede instalar backdoors, escanear la red, hacer un reconocimiento, o sacar información importante. Cuando se obtiene el acceso físico la batalla está perdida y es muy difícil tener protección, por eso se debe ser riguroso con el control de personas al acceso físico.

Hurgar en la basura: conocido como ”dumpster diving”, se analiza la basura de la persona u organización en busca de información sensible. Pueden obtenerse diagramas de red, borradores con usuarios y claves, CD’s, contratos, etc, o hasta un papel hecho un bollo con información importante y confidencial, donde el que la recauda puede obtener información para luego realizar un ataque.

Búsquedas en Internet

_ En este caso ya estaríamos implementando algún tipo de tecnología. Tenemos distintos servicios en los cuales se basa todo el funcionamiento de la internet, que son complementarios y normalmente los usamos de forma transparente sin saber que están:

Whois: es un protocolo TCP basado en petición/repuesta que se utiliza para efectuar consultas en una base de datos que permite determinar el propietario de un nombre de dominio o una dirección IP en Internet. Es decir, este servicio lo que hace es mantener una base de datos con la información de todas las direcciones y rangos IP de internet. Proporciona información sobre: nombre de dominio, NIC handle (identificador único para esa entidad dueña de ese ip o rango), direcciones IP, teléfono de contacto, email de contacto, dirección postal, fechas de validez, servidores DNS. Por ejemplo, en la consola

de Linux podemos poner el siguiente comando para ver mis datos, en el caso de no saber la dirección IP usamos el comando host y la dirección web:

host www.lanacion.com

whois 192.XXX.X.XX | *less*

DNS (Domain Name Service): es una base de datos jerárquica distribuida alrededor de Internet que proporciona información principalmente para la "resolución" de nombres de dominio a direcciones IP. Esta es una base de datos distribuida porque el control sobre porciones de esta base de datos depende del propietario de cada dominio en cada organización, entonces existen una serie de servidores raíz que se llaman route servers y que están distribuidos a lo largo de todo el planeta por cuestiones de performance y son los que tienen el comienzo de cada dominio, que es lo que llamamos "dominio." En donde tenemos por ejemplo los sufijos de cada país como .ar (Argentina), .uy (Uruguay), .br (Brasil), también tenemos los dominós raíz como .com, .edu, .org, etc. Entonces jerárquicamente en el primer nivel tenemos el punto, en el segundo nivel tenemos los sufijos de cada país, después para abajo tenemos los subdominios de cada país, por lo que tendríamos .com.ar, .edu.ar, .org.ar y de allí para abajo todos los dominios. De esta manera se va armando un camino a lo largo de esa jerarquía y conjuntos de esta son administrados por distintas personas.

_ Los dominios raíz son administrados por personas que gestionan las direcciones IP y nombres de dominio en todo el planeta, luego un ente regulador es NIC que gestiona todos los dominios terminados en .ar, también esta ARIU que gestiona los dominios para las escuelas y universidades, luego todo lo que sea anterior a estos lo gestionamos nosotros con nuestros propios servidores DNS con un nombre donde definimos todos los host que va a tener nuestro dominio. Como este es un servicio tan fundamental y tan necesario para el funcionamiento de internet y las redes en general, es mucha la información que tiene y que podemos tomar desde allí. Hay herramientas que hacen escaneos para ver cuáles son los hosts que tienen determinado dominio para recaudar información.

_ Dentro del servicio de DNS tenemos los principales "Resource records" o registros de recursos:

- A (Address): este registro relaciona un nombre de dominio o host correspondiente a una dirección IP especifica.
- MX (Mail eXchanger): este registro identifica los sistemas de correo electrónico válidos para un determinado dominio.
- NS (Name Server): este registro identifica los servidores DNS asociados a un dominio.

- PTR (PoinTeR): este registro proporciona resolución reversa, es decir, a partir de consultar con una IP nos dice cuál es el nombre de dominio. Los registro reversos suelen estar hosteados en los servidores DNS de los proveedores de internet.
- TXT (TEXT): este registro asocia un texto arbitrario con un nombre de dominio, y se puede usar con distintos fines.

_ Para hacer resolución de nombres, en Linux tenemos la herramienta *host* y en Windows la herramienta *nslookup*. Con -t preguntamos un tipo de recurso, en donde con *a* de alias nos devuelve la información, con *mx* nos devuelve los servidores de correo. También tenemos *txt*, entre otras

Bases de datos vulnerables

Conceptos

Vulnerabilidad: es una debilidad que está latente, influye negativamente en un activo y que posibilita la materialización de una amenaza y ataque exitoso. Se forma mediante la intersección de tres elementos, para que realmente se pueda dar la vulnerabilidad, que por un lado es una susceptibilidad o debilidad (flaw), luego el acceso del atacante a la debilidad y por último la capacidad del atacante para explotar dicha falla. Es decir, podemos tener un sistema muy viejo con varias vulnerabilidades, pero como sabemos que este sistema es muy vulnerable no lo exponemos a internet, o acotamos su acceso, con lo cual siempre está la vulnerabilidad pero al no ser accesible no puede explotarse, aunque también está el caso de que sea accesible el sistema, pero el atacante tiene que saber aprovechar esa vulnerabilidad mediante conocimientos y herramientas. Solamente cuando se la intersección de estas tres condiciones es que se puede explotar tal vulnerabilidad. Esta vulnerabilidad no se la piensa solamente como software sino que también puede abarcar distintos ámbitos como tecnología, procedimientos, controles y personas.

_ Cuando hablamos de seguridad en IT, tenemos que entender que paso, porque y sobre todo modificar los procedimientos, herramientas, o lo que haga falta para evitar que una falla vuelva a ocurrir. Uno permanentemente está analizando cual puede ser un punto débil y en cuanto a eso tomar medidas para evitar que algo ocurra.

Tipos de vulnerabilidades

_ En cuanto a lo que son vulnerabilidades de software nos vamos a centrar en las más comunes, sobre las cuales se basan la gran mayoría de los ataques o hackeos o incidentes que vamos a ver con distintos productos de software. Muchas están relacionadas y se combinan entre sí para lograr distintos objetivos.

Buffer Overflow: como su nombre lo indica es rebalsar o salirse de un buffer de memoria, haciendo que uno pueda escribir en la memoria fuera del espacio que tiene asignado para utilizar, entonces si logramos esto podríamos estar escribiendo información en un espacio de memoria que le corresponde a otro usuario que podría tener mayores privilegios que yo, por ejemplo de administrador, y luego vamos a poder ejecutar ese código con los permisos y el alcance que tiene un administrador. Este es una anomalía en la que un proceso guarda datos en un buffer fuera del espacio de memoria que el programador tiene para utilizar. Los datos extra sobrescriben la memoria adyacente, lo cual puede contener datos o instrucciones de otros programas.

_ En este ejemplo creamos una función con una variable de carácter de 12 bytes de longitud, y la función copia lo que recibe como parámetro a esa variable. Como vemos no se está checkeando si lo que recibimos tiene la longitud de la variable de destino y que quepa realmente, entonces el programa simplemente recibe el argumento, lo pasa a la función y esta misma escribe lo que recibe como argumento en la variable c. Como la variable c tiene 12 bytes de longitud asignado, si escribimos más allá de eso, lo que escriba de más va a ser en otra porción de la memoria que no era la que debía. El lenguaje C es muy propenso a esto. El lenguaje C es muy propenso a esto.

```
#include <string.h>
void foo (char *bar) {
   char  c[12];
   strcpy(c, bar);  // no bounds checking...
}
int main (int argc, char **argv) {
   foo(argv[1]);
}
```

Format string: surge de utilizar los datos de entrada de un usuario sin que sean filtrados o validados, pasándoselos luego a una función del programa que pueda interpretarlos textualmente. Esto está muy relacionado con el ejemplo anterior ya que no validamos los datos que entran y esto en general puede posibilitar algún tipo de ataque a futuro. Entonces básicamente format string es procesar y validar la información de entrada, si por ejemplo tenemos un campo de un usuario que es el DNI, no vamos a permitir en este caracteres especiales ni letras ya que lo que vamos a esperar son números enteros de ocho caracteres, entonces debemos hacer esas validaciones para evitar errores.

Code injection: surge de procesar datos inválidos ingresados por el usuario, y puede utilizarse para introducir (inyectar) código en un programa para cambiar el curso de ejecución. El ejemplo típico es lo que llamamos SQL inyection, que es básicamente en el campo de un formulario escribir código SQL, que si no está adecuadamente validado pasa directamente a la ejecución del programa y realiza distintas funciones o efectos indeseados.

Directory traversal: se da cuando no se validan correctamente las entradas de nombres de archivos por parte de los usuarios. Permite cambiar la ruta (path) de los archivos y obtener así información que no debe ser accesible. Es decir, es una falla en la cual se permite salir más allá del directorio que tengo asignado para funcionar el sistema, haciendo referencia a poder atravesar el directorio donde estamos confinados, por ejemplo que un usuario cualquiera pueda escalar directorios con cada vez más privilegios en un servidor web.

Race conditions: estos son bastante comunes y se producen cuando procesos separados o threads de ejecución dependen de algún estado compartido. Las operaciones con estados compartidos deben incluir mecanismos de sincronización para evitar colisiones entre procesos o threads. Es decir, esto se produce cuando tenemos distintos procesos o threads corriendo en forma concurrente, por ejemplo que modifican un cierta información, y no tenemos los mecanismos de sincronización para asegurar que esa concurrencia de la ejecución no altere la información. Puede haber situaciones donde un proceso modifica una variable, venga otro y la vuelva a modificar sin que el primer proceso se entere, y finalmente el proceso guarde lo que había en la variable sin saber que en el medio fue modificada por otro proceso.

Privilege escalation: permite obtener acceso a recursos que normalmente han sido restringidos para el usuario en cuestión. Generalmente ocurre cuando una aplicación con privilegios elevados tiene una falla que permite asumir dichos permisos. Esto es básicamente todo lo que cualquier atacante busca inicialmente y es ir escalando privilegios, o sea, lograr que un usuario de bajos privilegios pueda llegar a ejecutar o tener los privilegios de un administrador o un usuario que tenga los privilegios que necesito para realizar el ataque.

Revelación responsable (responsible disclosure): se refiere a los procedimientos y las formas por los cuales uno puede reportar o revelar una vulnerabilidad para que sea parchada o solucionada. No podemos exponer libremente vulnerabilidades sobre tal cosa en nuestro blog por ejemplo porque estaríamos revelando la falla a gente tanto buena como mala, por eso hay distintas políticas de procedimientos recomendados para reportar vulnerabilidades, pero en general se cumplen los siguientes pasos:

- El investigador que descubre una vulnerabilidad informa en forma confidencial al proveedor (vendedor) del software. Acá se usa el cifrado.
- Si el proveedor es receptivo y coopera, el investigador espera a que se publique el arreglo para revelar toda la información, salvo los exploits en el caso de que existieran.
- Si el proveedor no actúa de forma correcta, el investigador procede con una revelación completa (full disclosure), salvo el exploit.
- En cualquier momento, si se tiene constancia de que circula un exploit, se procede con la revelación de la información sobre la vulnerabilidad para proteger a los usuarios.

Vulnerabilidades famosas

_ Algunas vulnerabilidades famosas con la entidad donde se publicaron la mismas. Muchas veces se le pone un apodo o nombre a la vulnerabilidad para hacerlo más amigable:

- MS17-010 (Eternal Blue): fue uno de los ataques más costosos de la historia. Afecta a SMB (servicio de compartición de archivos de Windows) y fue utilizado por los ramsons WannaCry y Petya que se basaron en ella. Los reportes de seguridad de Microsoft empiezan con el identificador MS.
- CVE-2019-0708 (BlueKeep): afecta al servicio RDP (escritorio remoto) y permite ejecutar codigo en forma remota desde afuera y sin autenticación. CVE viene de Common Vulnerabilities and Exposures.
- Spectre/Meltdown: explotan vulnerabilidades críticas en procesadores modernos (mayormente Intel), y permite que un programa robe datos de otros ejecutándose en ese momento. Permiten el acceso al hardware por ejemplo el cache del procesador.
- CVE-2014-0160 (Heartbleed): afecta la implementación OpenSSL de la extensión TLS Heartbeat, y se basa en una inadecuada validación de la entrada (input validation) permitiendo acceder a porciones de memoria.
- CVE-2014-6271 (Shellshock): afecta el shell bash y permite tomar el control completo de sistemas Linux, Unix, Mac OS X, permitiendo ejecutar comandos con permisos de root.

Bases de datos de vulnerabilidades

_ Toda la información que se va reportando de forma responsable o no, que son vulnerabilidades que van apareciendo, se registran en distintas bases de datos llamadas base de datos de vulnerabilidades, hay algunas que son de entidades como ONGs o entes gubernamentales, y dependiendo de que se traten uno puede encontrar información más adecuada en una u otra. En estas bases de datos cada vulnerabilidad suele tener un ID asignado para diferenciarse.

- Common Vulnerabilities and Exposures (CVE): https://cve.mitre.org/
- National Vulnerability Database (NVD): https://nvd.nist.gov/
- Security Focus BUGTRAQ: https://www.securityfocus.com/bid
- Exploits Data Base (EDB): https://www.exploit-db.com/
- VulDB: https://vuldb.com/
- 0DAY Today: https://0day.today/
- Computer Incident Response Center Luxembourg (CIRCL): https://cve.circl.lu/

Port Scanning

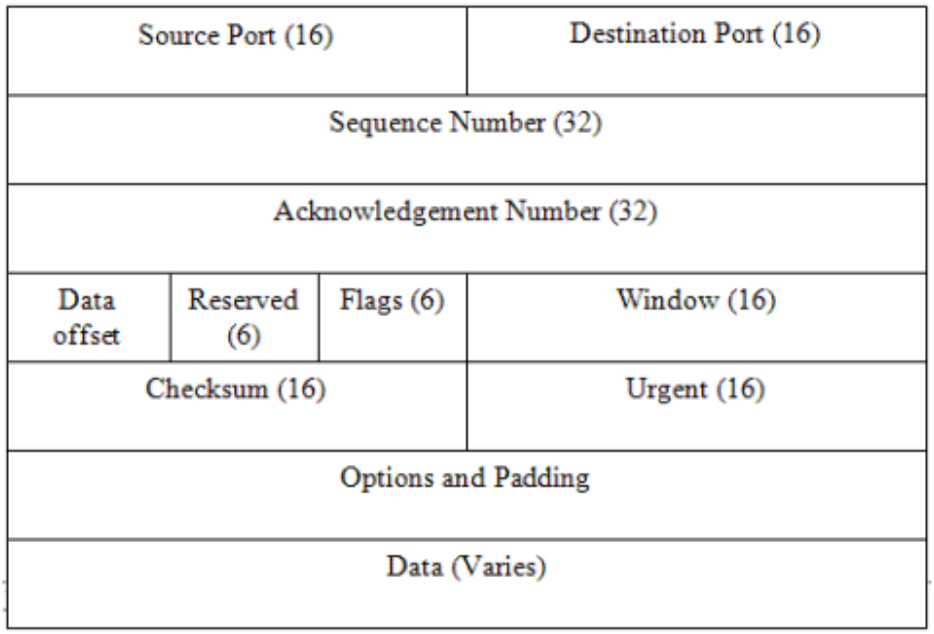
_ Ahora nos centramos más en lo que es el stack de red con servicios y protocolos, anteriormente vimos vulnerabilidades de aplicaciones y demás.

Port Scanning

_ Port Scanning es una técnica mediante la cual se realiza un escaneo o barrido de la red apuntado a un target, equipo o rango de red y nos indica cuales son los puertos TCP y/o UDP de ese determinado host o target, en busca de puertos abiertos que permitan conectarse a algún servicio. Esto nos permite tener un paneo general de que puede estar corriendo en un determinado equipo o que está expuesto a nuestro alcance. Cada puerto tiene como contraparte un servicio por detrás que lo está utilizando, lo está abriendo y que está escuchando en ese puerto, por ende un servicio escuchando es una potencial puerta de entrada al equipo atacado. Si los puertos son estándar podemos saber de algún modo cuales son los servicios que están por detrás y a lo cual nos podemos conectar.

_ Existen una serie de servicios “mayores” que utilizan una serie de puertos conocidos (well-known ports) y por estándares están reservados para determinados usos como por ejemplo el puerto 80 para HTTP, el puerto 433 para HTTPS, el puerto 110 para POP3, etc. En este enlace http://www.iana.org/assignments/port-numbers se hace referencia a las RTC que definen cuales son los usos para los puertos de servicios donde la mayoría están utilizados y podemos ver cosas como el nombre del servicio, el puerto que utiliza y si utiliza TCP, UDP o ambos. Mediante herramientas de port scanning se puede buscar una lista de puertos, un rango, o todos los posibles puertos TCP y UDP.

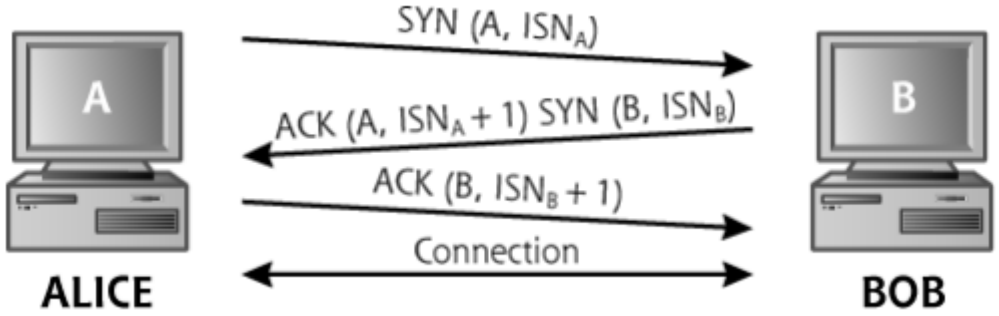
Encabezado TCP: por debajo el funcionamiento interno de este escaneo, si tomamos el protocolo TCP, este utiliza alguno de los campos que vimos clases anteriores para poder identificar información. Básicamente lo que buscamos son puertos de destino abiertos (destination port) pero vamos a usar algunas flags para poder realizar escaneos de distintos tipos según las flags que vamos a setear.



_ En este básicamente lo que más nos interesa es el puerto de destino, el puerto de origen, el número de secuencia y de acknowledge.

Tipos de scanning

TCP Connect Scan: intenta completar la negociación de tres vías (3-way handshake) con cada puerto buscado en el equipo "target". Es decir, intenta realizar una conexión completa a cada uno de los puertos que va a escanear, donde a partir de la negociación de tres vías se establece la conexión a través de la cual van a traficarse los datos y la información del servicio o protocolo que estemos utilizando en ese caso. Entonces el escaneo TCP connect lo que hace es, por cada uno de los puertos que quiere escanear realiza este proceso.



- Puerto abierto: si se completa correctamente la negociación, significa que el puerto está abierto, es decir, recibimos las respuestas desde el target y se completó la negociación para un puerto determinado, donde luego ese puerto se considera abierto.
- Puerto cerrado: si el puerto está cerrado el target devuelve o un RESET (paquete TCP que tiene una flag seteada en 1), o ICMP Port Unreachable o nada. Dependiendo de las implementaciones del protocolo TCP, puede ser el comportamiento que se tenga, y hay estándares que establecen que debe devolver un RESET o un ICMP Port Unreachable o nada.

_ A continuación tenemos las ventajas y desventajas de este escaneo:

- Ventajas: se realiza un comportamiento estándar, es decir, es la forma en la que fue definido y se dice como hay que utilizar el estándar, entonces esto hace que no tengamos fallas o comportamientos extraños en el target.
- Desventajas: es fácil de detectar, porque como se establece la conexión, aquello a lo que nos conectamos normalmente registra en un log o en un archivo donde indica todas las acciones que realiza.

TCP SYN Scan: básicamente completa solo los dos primeros pasos de la negociación, es decir, envía el paquete de sincronización SYN y espera el SYN-ACK correspondiente. No termina enviando el tercer paquete que completaría la conexión sino que solo los dos primeros. Entonces:

- Puerto abierto: si se recibe una respuesta, o sea el SYN-ACK correspondiente, significa que el puerto está abierto.
- Puerto cerrado: si el target devuelve un RESET, ICMP Port Unreachable o nada, significa que el puerto está cerrado.

_ A continuación tenemos las ventajas y desventajas de este escaneo:

- Ventajas: es menos detectable porque al servicio al que intentamos conectarnos no se termina de conectar por ende no se registra la conexión, y es más rápido porque no tenemos que completar toda la conexión.
- Desventajas: en algunos equipos viejos o desactualizados pueden fallar ante este tipo de scanning, produciéndose un DoS (denegación de servicio).

TCP FIN Scan: envía paquetes de finalización de la conexión (FIN) sin que la conexión exista. Es decir, en vez de iniciar como sería una comunicación normal o a medias, envía un paquete que está finalizando la conexión que ya existía de antes.

- Puerto abierto: si el puerto está abierto no se recibe nada, es decir, ninguna respuesta.
- Puerto Cerrado: si el puerto está cerrado, según el estándar TCP cuando uno recibe un paquete de finalización sobre un puerto cerrado, el estándar dice que el sistema target debe responder con un RESET (especificación TCP).

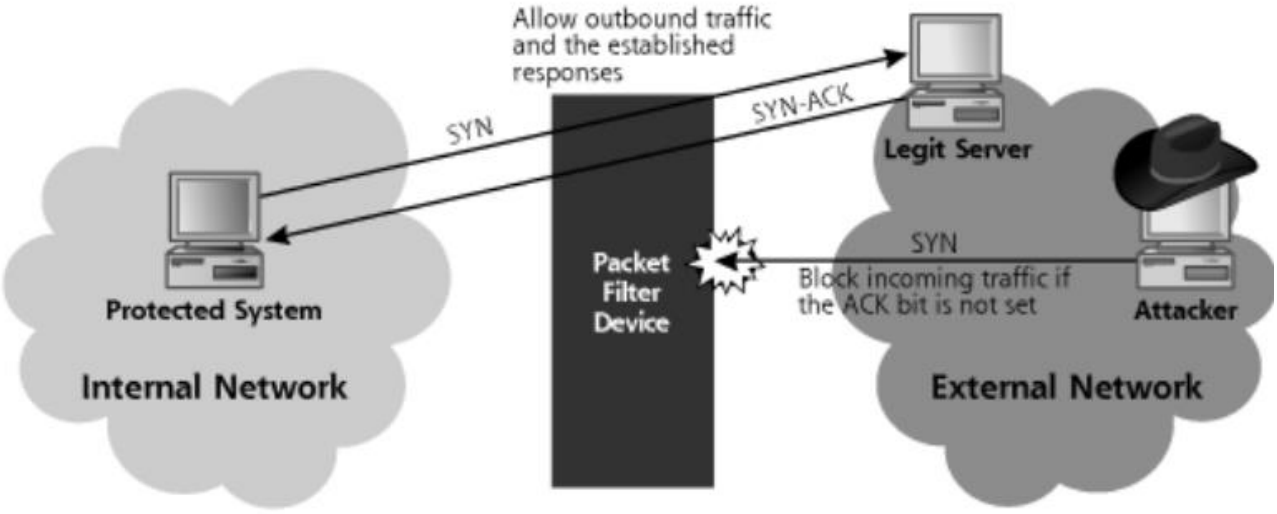
TCP Xmas Tree Scan: envía paquetes con todos los bits de control activados (URG, ACK, PSH, RST, SYN y FIN). Este básicamente setea todos los bits de flags en 1 apuntando por ejemplo a algunos dispositivos viejos que por ahí chequeen si esta seteada o no determinada flag para dejar pasar el paquete, entonces seteando todos en 1 de algún modo puede confundir y lograr pasar el paquete.

- Ventajas: puede pasar a través de algunos routers o firewalls viejos que busquen que bits de control específicos estén activados.
- Desventajas: puede ser detectado por IDSs modernos, entonces si vemos un paquete TCP que tiene todos los bits seteados, inmediatamente se sabe que es un ataque o escaneo de este tipo porque no hay ningún comportamiento estándar que utilice todos los bits seteados a la vez.

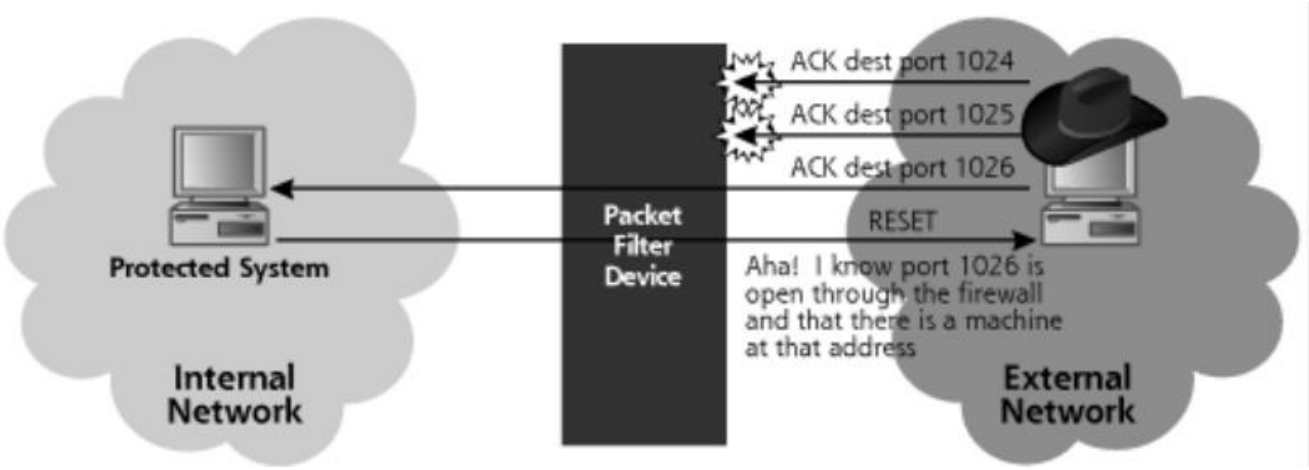
IDS (Intrusion Detection System): son como si fueran un antivirus pero de red, es decir, un servicio que en todo momento está monitoreando el tráfico de red en búsqueda de patrones extraños o de ataques conocidos.

TCP Null Scan: es el opuesto del Xmas Tree, donde este envía paquetes con todos los bits de control desactivados y todas las flags seteadas en cero. Tiene los mismos objetivos, ventajas y desventajas que el anterior, siempre pensando en dispositivos viejos porque los modernos reconocen este tipo de comportamiento extraño.

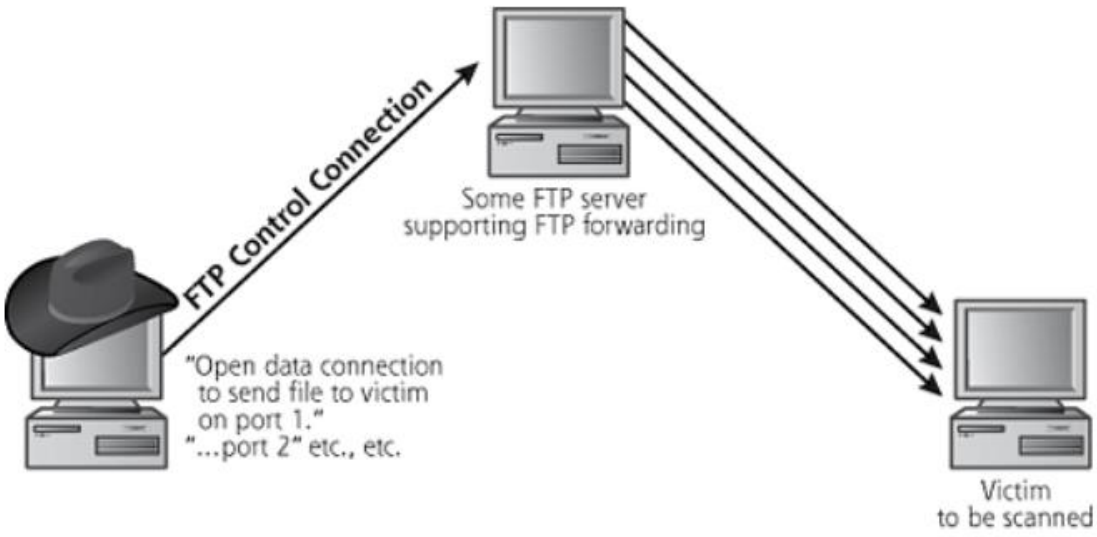
TCP ACK Scan: básicamente envía los paquetes con el bit ACK activado para que los firewalls crean que son paquetes de respuesta. Analizando el esquema tenemos la red externa como por ejemplo internet, también tenemos nuestro dispositivo de filtrado de paquetes o firewall y por último la red interna. Normalmente los firewalls están configurados para impedir las conexiones externas salvo que explícitamente se permita algún puerto que se esté necesitando publicar, pero todo lo que no esté explícitamente permitido esta denegado, entonces si realizamos por ejemplo un FIN Scan, Connect Scan o lo que sea, el firewall va a filtrar y bloquear todas esas conexiones, por lo que no nos va a servir ese tipo de escaneo. Por otro lado el firewall en las configuraciones típicas permite que el tráfico generado en la red interna pueda volver a salir a internet, donde se inicia la conexión con el SYN correspondiente al servidor, y el servidor me responde con un SYN ACK donde esta respuesta que se origina como respuesta como un requerimiento externo el firewall por default la deja pasar porque es parte de la misma conexión.



_ El ACK Scan consiste en que el atacante envía pruebas a distintos puertos con el bit de ACK seteado, simulando ser una respuesta a un tráfico generado desde adentro. Hay algunos dispositivos de firewall sobre todos los más viejos que podían ser engañados bajo esta condición y entonces dejaban pasar el ACK pensando que era respuesta de algo que inicio un host interno. Como vemos en la imagen se envían los ACK a distintos puertos y cuando pasa de largo, es decir, cuando se recibe un paquete de ACK en un puerto en el que no lo estamos esperando porque no lo pedimos, el sistema interno responde con un RESET pudiendo ser detectado por el atacante y de este modo sabe que ese puerto está abierto.



FTP Bounce Scan: el atacante utiliza una vieja funcionalidad habilitada en algunos servicios de FTP para ocultarse. Es un escaneo por rebote, donde básicamente lo que hace es utilizar un dispositivo viejo o básico que tenga la funcionalidad de FTP activada y usarla como trampolín para allí largar el escaneo hacia el target que se esté buscando. Hay una serie de dispositivos que tienen este servicio activado, en donde el atacante le pide al dispositivo que inicie una conexión en un El atacante le pide al dispositivo que inicie la conexión en un tercero en un puerto determinado y el dispositivo replica eso hacia la víctima. La gran ventaja de esto es que se pasa desapercibido y lo más furtivo posible para que no se lo detecte, por lo que la víctima va a ver conexiones desde este dispositivo intermedio. Desde el equipo de la víctima no se le puede hacer una auditoria a este dispositivo intermedio al ser tan básico, como puede ser una impresora o cámara de seguridad, donde seguro no tiene logs ni registros de nada y de esta forma perdemos el vínculo con el atacante y no podemos rastrearlo.



UDP Scan: se envía un paquete UDP a cada puerto. Como vimos UDP es muchísimo más simple y básico que lo que es TCP, y por ende los escaneos son mucho más simples. Básicamente se envía un paquete UDP a cada puerto y entonces:

- Puerto abierto: si el puerto está abierto se recibe otro paquete UDP de respuesta.
- Puerto cerrado: si el puerto está cerrado el sistema target responde con un ICMP Port Unreachable.

Version Scan: luego de identificar los puertos abiertos podemos ir más allá e intentar determinar qué servicio, producto software o versión en particular está escuchando detrás de ese puerto. Entonces con este podemos lograr que nos indique la información de un puerto 80 que este abierto, que seguro es un servidor web, y que nos diga que es un Apache versión tal que esté funcionando detrás del puerto.

Ping Sweep: significa barrido de ping y lo que hace es enviar un ping a todo un rango de direcciones que indiquemos y de ese modo saber que dispositivos están encendidos en nuestra red. Es decir, se envían paquetes ICMP Echo Request a una lista o rango de direcciones IP para determinar cuales tienen hosts activos en esa red.

OS Fingerprinting: se envía una serie de paquetes a varios puertos en el target para, en base a la respuesta recibida, determinar el sistema operativo remoto. Utiliza las respuestas que nos da el stack TCP de aquello a lo cual nos estamos conectando para identificar el sistema operativo. Como sabemos cada sistema operativo tienen sus distintas implementaciones de TCP/IP y difieren en cosas muy sutiles pero que nos permiten de algún modo reconocer patrones que nos indiquen que sistema operativo es.

_ Todas estas técnicas de escaneo forman parte de las técnicas de reconocimiento, y con estos podemos obtener información más concreta del target que buscamos.

Herramientas

Nmap: es la más completa y la que más se suele utilizar para scanning, es open source y está disponible libremente, corre en los sistemas operativos más utilizados como Unix, Linux, Windows, y puede utilizarse desde línea de comandos o se integra con otras aplicaciones que lo usan como librería y que aprovechan todas las funciones que esta provee. A continuación mostramos los parámetros de la herramienta:

Tipo Scan	Opción línea comandos
TCP Connect	-sT
TCP SYN	-sS
TCP FIN	-sF
TCP Xmas Tree	-sX
Null	-sN

TCP ACK	-sA
FTP Bounce	-b
UDP	-sU
Version	-sV
Ping Sweep	-sP
OS Fingerprint	-O

Packet Sniffing

Packet Sniffing

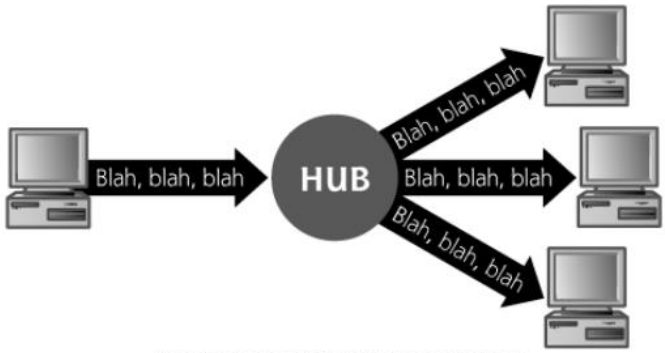
_ Packet sniffing significa básicamente husmear o ver el tráfico de la red, y se utiliza una herramienta llamada packet sniffer que nos permite analizar todo el tráfico.

Packet Sniffer: también conocido como “packet analyzer”, “protocol analyzer” o “network analyzer”. Es un software o hardware que puede interceptar y registrar el tráfico que pasa a través de una red digital, o parte de ella. Esta nos solo es con usos malignos o mal intencionados sino que en el trabajo de un ingeniero también puede ser utilizado para:

- Analizar problemas de red ante un comportamiento extraño o algo que no funciona y poder así entender lo que está pasando.
- Detectar intentos de intrusiones en la red. IDS básicamente es un packet sniffer que está permanentemente capturando el tráfico de red y buscando si hay un ataque o comportamiento extraño.
- Obtener información para atacar una red. Como veíamos recién con el scanning uno puede relevar y hacer un reconocimiento con esas herramientas.
- Monitorear el uso de red, en donde podemos hacer estadísticas con productos que por ejemplo hacen gráficos de torta de que porcentaje del tráfico de la red es TCP, o es tráfico web o tráfico de correo o el protocolo que se nos ocurra.
- Espiar usuarios u obtener información sensible. Si estamos capturando un tráfico que no es cifrado y es texto plano, uno puede ver directamente en ese tráfico información como usuarios y contraseñas directamente visibles.
- Hacer ingeniería reversa de protocolos propietarios. Recordamos el concepto de propietario vs lo abierto, en donde no por ser propietario y oculto va a ser más seguro que algo abierto cuyas especificaciones estén libremente disponibles. Uno puede utilizar estas herramientas para agarrar un protocolo propietario del cual no tenemos especificaciones y podemos ir estudiando cómo funciona y cómo se comporta y hacer una ingeniería reversa, es decir, entender cómo funciona y de algún modo documentarlo.

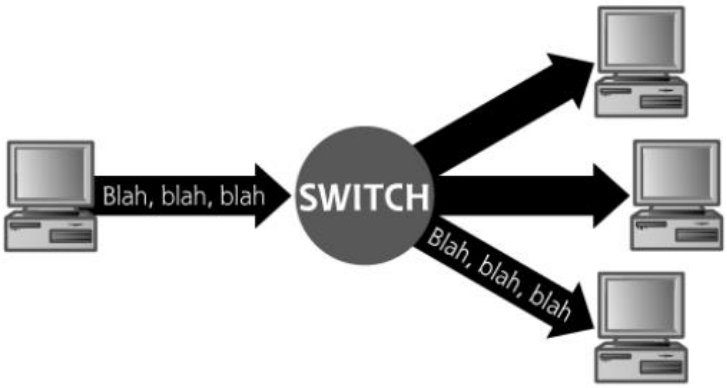
Tipos de sniffing

Pasivo: consiste en escuchar y capturar el tráfico que pasa por la red, y no tenemos que realizar ninguna acción para que ese tráfico me llegue y lo vea. Para ver el tráfico de toda la red, se requiere un dispositivo del tipo hub que es un predecesor de los switch. Un hub es un dispositivo en el que se conectaban todos los equipos de la red pero tenía la característica de que simplemente era un retransmisor, es decir, lo que envía quien que genera la comunicación llega al hub, y luego este lo retransmite a todos los hosts. Entonces si el equipo individual del grafico quería hablar con el último de los tres, toda la información llegaba igualmente a los otros, en donde si el primer equipo utiliza alguna herramienta de sniffing podría ver el tráfico entre los otros equipos o hosts. Esto No actualmente no se usa al menos que sea una red muy vieja. Por otro lado no es detectable.



BROADCAST ETHERNET

Activo: ahora usamos un switch que ya es más inteligente porque mantiene dentro de su memoria y funcionamiento tablas ARP, entonces sabe cuál es el host que está hablando con cual y solamente retransmite la información a través del puerto correspondiente y nada a los otros. No podremos calcular el tráfico desde otro host que no esté participando de la comunicación. Entonces por eso surge la necesidad de otro tipo de sniffing que es el activo en donde se lanza un ataque de ARP Spoofing para capturar tráfico en redes con dispositivos como un switch para engañar a los dispositivos o a alguno de los equipos para hacerle creer que la MAC con la cual tienen que hablar es la del atacante y no la del usuario real. Como esto implica una acción de nuestra parte es detectable ya que tenemos que enviar peticiones raras en la red y detectables por un IDS por ejemplo para generar esta confusión en los switch o en los equipos y de ese modo podemos ser detectados.

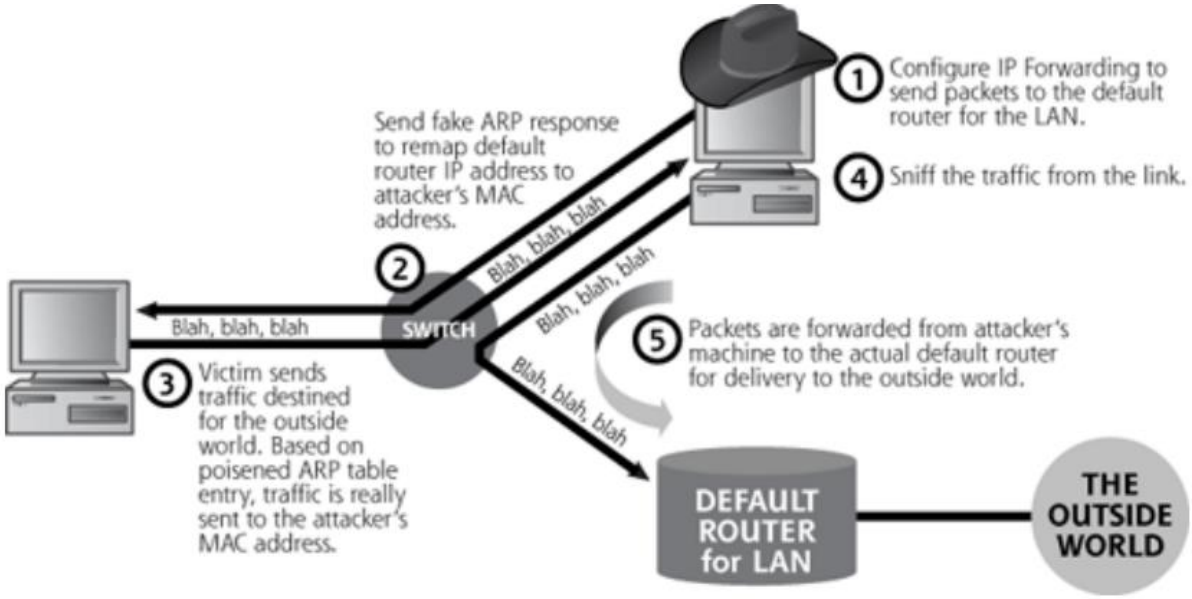


SWITCHED ETHERNET

ARP Spoofing

_ ARP Spoofing o también conocido como ARP Poisoning, es una técnica utilizada para atacar a una red Ethernet y que permite al atacante capturar tráfico en una red con switchs. Lo que hace es enviar paquetes o mensajes ARP falsos, confundiendo así a los hosts y dispositivos de la red.

_ Analizando el grafico, tenemos el atacante que configura su equipo para que pueda hacer reenvío de conexiones de otros equipos, se lo configura como si fuera un router o un ip forwarding, y lo que hace es enviar peticiones falsas de ARP respuestas, mapeando la IP en la puerta de enlace del router o la red a su dirección MAC. Esto hace que la víctima en vez de enviar el tráfico que está destinado a internet al router, se lo envía sin saberlo al atacante el cual puede ver el tráfico, puede hacer un sniffing, guardarlo o modificarlo, y luego lo reenvía transparentemente a través del router real y va a internet. Esto es un típico ataque de man in the middle, donde nos colocamos en el medio de la comunicación engañando a ambos extremos haciéndoles creer que están hablando entre si cuando en realidad están pasando por nosotros y de ese modo podemos ver el tráfico, modificarlo, etc.



- Prevención: una forma de prevenir esto es agregar o definir una entrada estática en la tabla ARP del equipo en cuestión con la IP y MAC del gateway por defecto. Esto lo haríamos con el siguiente comando, donde agregamos -s de static, es decir, agregamos una entrada estática, seguido de la dirección IP y luego la MAC:
  arp -s direccion_IP direccion_MAC

Herramientas

tcpdump: dentro de las herramientas de sniffing este es la más conocida y la más utilizada. Es una herramienta de línea de comandos para analizar el tráfico que circula por la red. Es libre y está disponible en todos los sistemas operativos, y este básicamente captura el tráfico permitiéndolo guardar en un archivo por convención ".cap" aunque en Linux no influye en nada. Podemos aplicarle distintos filtros de búsqueda a ese tráfico para ver lo que nos interese, entonces podemos filtrar de acuerdo a:

- type: haciendo referencia al tipo e indica a que se refiere el nombre o número pasado como parámetro. Tipos posibles son host, net, port y portrange. Ejemplo: 'host foo', 'net 128.3', 'port 20', 'portrange 6000-6008'.
- dir: según la dirección de la conexión, en donde se especifica la dirección desde/hacia el nombre o número indicado. Las direcciones posibles son src, dst, src or dst, src and dst, etc. Ejemplo: 'src foo', 'dst net 128.3', 'src or dst port ftp'.
- proto: filtramos por protocolo y se restringe la búsqueda a un protocolo en particular. Los protocolos validos son: ether, fddi, tr, wlan, ip, ip6, arp, rarp, tcp, udp, etc. Ejemplo: 'ether src foo', 'arp net 128.3', 'tcp port 21', 'udp portrange 7000-7009'.

_ Realizamos algunos ejemplos de aplicación de esta herramienta:

- Capturar el tráfico de la interfaz eth0 y enviarlo a un archivo. El *-n* significa que no trate de resolver o traducir de la IP al nombre y viceversa, luego *-i* de interface y por último el nombre de la interfaz en la cual vamos a realizar la captura:
  tcpdump -w archivo.cap *-ni* eth0

- Capturar el tráfico ICMP de la interfaz wlan0, a excepción del dirigido a la IP 192.168.10.200:
  tcpdump -ni wlan0 icmp *and not host* 192.168.10.200

- Capturar en un archivo el tráfico HTTP o HTTPS de la interfaz wlan0:
  tcpdump -w captura.cap *-ni* wlan0 *port* 80 *or port* 443

Wireshark: esta herramienta es más sofisticada y más amigable, y es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, y como herramienta didáctica para educación. Se caracteriza por tener múltiples herramientas de análisis de tráfico y una completa interfaz gráfica. Utiliza como backend todas las librerías de tcpdump y permite mostrarlo y analizarlo de forma gráfica para poder entender cómo funcionan cada una de las capas del protocolo que estemos analizando. Tiene un montón de protocolos y herramientas cargadas que nos permiten hacer estadística y seguimiento de tráfico. Esta herramienta también tiene, similar a tcpdump, una sintaxis en la que podemos realizar distintos filtros:

- Filtros de captura: podemos filtra paquetes durante la captura, por ejemplo podemos filtrar por una red de origen o por host:
  src net 192.168.1.0/24
  host 10.20.30.1 and not port 80 and not port 22

- Filtros de pantalla: filtramos como se muestra u oculta información en la pantalla sobre los paquetes de la captura analizada. La sintaxis es el nombre del protocolo seguido de un punto y el campo del protocolo que nos interesa analizar:
  tcp.port eq 25
  ip.addr == 10.43.54.65
  http.host matches “acme\.(org | com | net)”

Seguridad en sistemas operativos

Sistema operativo Unix

Cuentas de usuario

_ En cada sistema operativo las cuentas de usuario se representan mediante una entidad que es el “User Name” o el “User ID”, que es un identificador que asocia la persona física con la cuenta virtual de este en el sistema operativo. Cada usuario tiene una cuenta en el sistema operativo. Existen distintas cuentas de usuarios del sistema y de usuarios reales y también cuentas de servicios que pueden estar corriendo en el sistema operativo. Cada proceso que se ejecuta en el sistema operativo con los permisos de una determinada cuenta, siempre son disparados por un usuario y corren con los privilegios de ese usuario, entonces dependiendo de cuál sea el usuario que levanto el proceso, serán los permisos de lectura, escritura, ejecución, etc, que pueda tener el proceso. La base de usuarios se almacena en un archivo de texto que tiene un formato especifico y normalmente se utiliza una línea por cada cuenta y los campos delimitados por algún carácter que en este caso es el carácter “:”. Entonces en el archivo /etc/passwd es el archivo en Linux donde se almacenan las cuentas de usuario y la información principal de esas cuentas. En este archivo vamos a tener una línea por cada cuenta y cada campo separado por el carácter “:”.

```
root:x:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:daemon:*:2:2:daemon:/sbin:
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
juan:$1$hwqqWPmr$TNL0UManaI/v0coS6yvM21:501:501:
  Juan Perez:/home/juan:/bin/bash
```

_ Entonces viendo esto vemos que hay un usuario especial que es el usuario root, que es el usuario administrador que tiene todos los privilegios y puede hacer cualquier cosa en el sistema operativo, y que se caracteriza por tener el User ID o identificador numérico del

usuario en cero. Entonces lo que determina que un usuario sea root es el User ID cero. Después vemos distintas cuentas para determinados usos y menores privilegios.

Formato /etc/passwd: los campos siguen el siguiente orden:

- username: es el nombre que va a tener la cuenta de usuario en el sistema. En Unix, los username no deben contener letras mayúsculas, por la condición de que estos sistemas son key sensitive.
- password: contraseña cifrada del usuario. Este campo puede también contener un asterisco, lo cual significa que el usuario está bloqueado, y por otro lado también puede tener la letra "x", que indica que la contraseña esta almacenada en el archivo /etc/shadow.
- UID: (user ID) una identificación numérica (entero) única para el usuario o la cuenta de usuario. En definitiva esto es lo que el sistema operativo maneja, pero nosotros le ponemos un nombre para que sea más fácil manejarlo.
- GID: (group ID) una identificación numérica única para el grupo primario del usuario. Es la misma idea que los user ID.
- full name: en este campo se coloca el nombre completo del usuario real o algún comentario que se quiera agregar.
- home directory: indica el directorio en el cual cae el usuario en cuestión al momento de loguearse. En este directorio el usuario tiene todos los privilegios y está destinado a que el usuario guarde toda su información personal y hacer lo que necesite. Normalmente la partición es /home y el nombre del usuario.
- shell: aquí se indica el comando que se va a ejecutar en el momento que el usuario inicie una sesión. No necesariamente requiere ser un Shell sino que es el programa que se va a ejecutar cuando el usuario inicie sesión, pero usualmente es el shell. En el caso del ejemplo es /bin/bash.

_ El archivo /etc/passwd tiene la característica de que es legible por todo el mundo. Es la base del usuario de un sistema que es necesario poder leerla por parte de cualquier usuario porque de allí se toma cuál es su ID, group ID, home, etc. Pero el problema es que al tener este archivo legible por todo el mundo y con contraseñas adentro es un riesgo para la seguridad, entonces lo que se hizo fue partir la información y crear un nuevo archivo etc/shadow.

_ Si listamos el archivo passwd vemos que es propiedad del usuario root y el grupo root, pero solamente root tiene lectura y escritura y todo el resto tiene solo lectura. Y si mostramos el contenido veremos los distintos usuarios con el home y el shell, también vemos que todos tienen el campo de la contraseña con una "x" que indica que esa información esta almacenada en el archivo /etc/shadow:

ls -l /etc/passwd

cat /etc/passwd

Archivo /etc/shadow: los sistemas UNIX modernos utilizan /etc/shadow para guardar las contraseñas de los usuarios y este archivo solo es accesible por el usuario root. Tiene un formato legible y similar como passwd, pero tiene en el primer campo el nombre del usuario, y en el segundo campo una representación para la contraseña:

```
root:$1$bed128365216c019988915ed3add75fb:
   14729:0:99999:7:::
daemon:*:14728:0:99999:7:::
bin:*:14728:0:99999:7:::
```

_ Si listamos el archivo /etc/shadow, vamos a ver que solo root tiene los permisos de lectura y escritura, lo mismo con el grupo, pero el resto de los usuarios no tienen ningún permiso. Si queremos ver el contenido, nos va a indicar que no podemos hacerlo a menos que lo hagamos como administrador y veremos los campos con las contraseñas.

ls -l /etc/shadow

cat /etc/shadow

sudo cat /etc/shadow

Cuentas de grupos

_ Además de tener las cuentas de usuarios tenemos los grupos, que son una forma de juntar determinados usuarios para asignarles en conjunto una determinada política y de ese modo simplificar la administración de permisos de usuarios. Linux y UNIX soportan la creación de grupos de usuarios. Los grupos se almacenan en el archivo /etc/group y como vemos tiene un formato similar a los anteriores, donde tenemos el nombre del grupo, el identificador del grupo group ID y después una lista separada por coma de los usuarios que pertenecen a ese grupo, entonces teniendo esta forma de agruparlos uno puede definir los permisos a varios usuarios de una sola vez.

```
root:x:0:
daemon:x:2:root,bin
desarrollo:x:25:juan
```

Representación de contraseñas

_ La representación de la contraseña en los archivos shadow fue evolucionando en el tiempo, como decimos, normalmente lo que es seguro hoy probablemente ya no lo sea de acá a unos meses o años ya que va evolucionando por un lado la capacidad de cómputo y por otro lado se pueden ir encontrando fallas o vulnerabilidades en los propios algoritmos que utilizamos. Inicialmente los primeros UNIX utilizaban la función crypt(3) con un algoritmo criptográfico sumamente débil. Luego se pasó al algoritmo de cifrado DES y

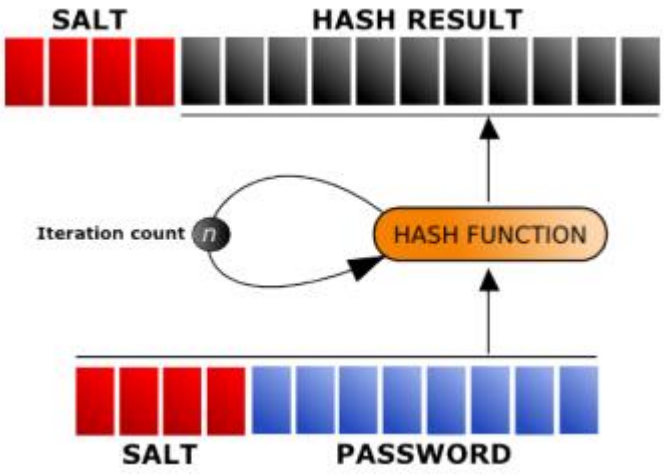
posteriormente a la función de hash MD5, donde este último fue durante mucho tiempo el estándar y lo más utilizado pero actualmente ya está en desuso porque se han encontrado huecos o se ha procesado gran parte del Hash con algunas tecnologías lo cual hace que sea posible encontrar la información que oculta. Actualmente existen implementaciones más seguras basadas en Blowfish, MD5 y SHA (SHA-256 y SHA-512).

_ El formato de la contraseña en el campo correspondiente de /etc/shadow está formado por tres campos separados por el carácter "$" donde cada campo tiene un sentido, el primero es el identificador del algoritmo que se va a utilizar, el segundo es una SALT que es una cadena aleatoria, y el ultimo campo es la contraseña representada con el algoritmo seleccionado. A continuación vemos una tabla con los distintos identificadores que podemos llegar a ver:

$<ID>$<SALT>$<PWD>

ID	Método
1	MD5
2b	Blowfish
5	SHA-256
6	SHA-512

Password salting: los algoritmos de cifrado y hashing son públicamente conocidos, y en base a la entrada que ingresa el usuario se calcula el hash correspondiente. Si se consigue la base de usuarios, pueden probarse todas las combinaciones hasta encontrar las contraseñas. Si alguien logra hacerse de un etc/shadow mediante el password salting hacemos que este proceso sea mucho más dificultoso. Lo que se hace es, al hash que se le calcula a la contraseña ingresada por el usuario, se le antepone una cadena aleatoria que se llama SALT, luego se le calcula función de hash con la concatenación de la SALT con la password que ingresa el usuario, y se almacena el resultado. La función hash se suele iterar o calcular muchas veces para generar el resultado porque lo que se busca con esto es dificultarle la tarea al atacante en el hecho de encontrar la contraseña sobre todo con el tiempo que le llevaría. Este proceso no lo utiliza Windows.



Sistema operativo Windows

Cuentas de usuario

_ En Windows existen dos tipos de cuentas y estas son las cuentas por defecto, o sea cuentas ya predeterminadas por el sistema operativo, y por otro lado el resto cuentas creadas por un administrador. La cuentas por defecto son “Administrator” y “Guest”, donde la cuenta de administrador tiene el mayor nivel de privilegios y no puede borrarse ni bloquearse porque se inutilizaría el sistema. En el caso de la cuenta Guest por defecto y por cuestiones de seguridad viene deshabilitada. El resto de cuentas de usuarios, sistemas o procesos son creadas por el administrador.

Cuentas de grupo

_ Los grupos se utilizan para controlar accesos y privilegios. A diferencia de Unix, en Windows existen dos tipos de grupos:

- Globales: estos grupos están en un dominio de Windows y no proporcionan directamente ningún acceso a ningún recurso, sino que estos grupos engloban otros grupos locales o usuarios locales que son los que si dan el nivel de acceso.
- Locales: brindan acceso a los recursos del equipo local. Uno puede meter dentro de un grupo local un grupo global y de ese modo darle permiso.

_ Por ejemplo, tenemos nuestra estación de trabajo Windows que pertenece a un dominio. En esa estación de trabajo hay un administrador local y un grupo de administradores locales, y dentro de ese grupo podemos incluir el grupo de administradores del dominio general de la red y de ese modo le estamos dando permiso a cualquier administrador general de la red a manejar mi equipo y utilizar los permisos de privilegio que le asignemos.

_ Dentro de lo que son los grupos tanto locales como los globales tenemos una serie de grupos pre definidos, pero luego uno puede crear todos los grupos que necesite:

Grupos Locales	Grupos Globales
Administrators	Domain Administrators
Account Operators	Domain Users
Server Operators	
Backup Operators	
Print Operators	
Replicators	
Users	
Guests	

Representación de contraseñas

_ En cuanto a la representación de contraseñas, en un sistema Windows no utilizamos como en Unix un archivo de texto sino que la información de las cuentas se almacena en una base binaria llamada SAM (Security Accounts Manager). Este SAM está ubicado dentro de la estructura de directorios de Windows, en lo que es \%systemroot%\system32\config\SAM. El archivo SAM se encuentra cifrado a nivel filesystem con una llave de 128 bits llamada SYSKEY y contiene las contraseñas representadas en dos formatos LM Hash y NT Hash. Lo que vemos a continuación es un dump de la SAM, hay utilizades que acceden a la base de usuarios de la SAM de Windows y hacen un dump a texto con el formato que vemos que es similar a lo que tenemos en Linux con una línea por cada cuenta donde tenemos el nombre de la cuenta, el ID y la representación de las contraseñas en los dos formatos:

```
Administrador:500:855c3697d9979e78ac404c4ba2c66533:
  7f8fe03093cc84b267b109625f6bbf4b:::
Invitado:501:552902031bede9efaad3b435b51404ee:
  878d8014606cda29677a44efa1353fc7:::
paula:1005:4d98b75fd1dacd79aad3b435b51404ee:
  74ed32086b1317b742c3a92148df1019:::
```

_ Como en este archivo tenemos las contraseñas expresadas en los dos formatos, el problema es que tenemos un eslabón más débil, ya que de nada nos sirve que tengamos una representación segura si está al lado de una vulnerable y fácil de romper por lo que no tiene sentido. Lo que se suele hacer es los sistemas operativos más modernos es deshabilitar la compatibilidad con los sistemas viejos lo cual deshabilita la opción LM.

_ Para saber cómo funcionan cada una de estas dos representaciones de contraseñas, es decir, LM Hash y NT Hash, tenemos que saber que existen estas dos porque originalmente con los primeros Windows surgió Hash LM (LAN Manager) y luego fue evolucionando hasta quedarse en Hash NT, pero por cuestiones de compatibilidad Windows mantuvo las dos representaciones, por si hay un cliente viejo en la red que pueda seguir interactuando con el equipo.

Hash LM: este algoritmo es bastante básico y vulnerable. Para el procesamiento, en este caso, se ajusta la longitud de la contraseña a exactamente 14 caracteres, si el usuario pone caracteres de menos se completa el resto de los campos con caracteres NULL al final de ser necesario, y si pone de más descarta lo que supere los 14 caracteres. La cadena resultante de 14 caracteres que queda, es dividida en dos partes iguales de 7 caracteres, y a cada par se le agrega un carácter de paridad requerido por el algoritmo DES para generar los hashes. Luego, cada parte se utiliza como “llave” para cifrar con el algoritmo DES una cadena de caracteres constante "KGS!@#$%", entonces lo que estamos guardando y cifrando es esa cadena de símbolos, utilizando una parte de la contraseña

que ingreso el usuario como llave con el algoritmo DES. Esto tiene una serie de debilidades:

- Los caracteres de contraseña están limitados al conjunto de caracteres ANSI (grupo de codificación de caracteres antigua que tiene un grupo pequeño de caracteres en comparación de las nuevas representaciones que surgieron posteriormente, con lo cual acotamos el universo de probabilidades de contraseñas distintas que podemos hacer).
- Las contraseñas mayores a 7 caracteres son divididas en dos partes que pueden ser luego atacadas por separado, por lo que acotamos cada vez más las posibilidades de combinaciones que puede haber, pasando de 2^{92} posibles combinaciones a 2^{46}.
- Todas las letras minúsculas son convertidas a mayúsculas antes de ser utilizadas para cifrar, por lo que también se reducen las posibilidades a 2^{43}.
- LM no utiliza “password salting” lo cual lo hace más vulnerable.

Hash NT: esta evolución de Hash LM amplia el conjunto de caracteres validos a Unicode, es decir, ya no tenemos solamente ANSI sino que también tenemos caracteres de Unicode como posibles entradas. Utiliza el algoritmo de hashing MD4 para producir un resumen de la contraseña expresada en codificación UTF-16-LE, donde el proceso seria MD4(UTF-16-LE(password)). Esta tampoco utiliza “password salting”.

Ataques contra contraseñas

Ataques contra contraseñas

_ Se pone en foco las contraseñas porque son la herramienta más utilizada en el mundo de la seguridad de la información para validar de algún modo una identidad. Esto es así porque las contraseñas son fáciles de acordarse y terminan siendo el mecanismo de identificación más barato, no solo en costo sino también en accesibilidad. Por otro lado las contraseñas al ser creadas por humanos son normalmente el eslabón más débil en la seguridad de los sistemas porque lo que se busca es la facilidad para recordarla o escribirla y por eso es más fácil para una herramienta de crackeo romper eso que un esquema criptográfico. Y repitiendo, estas se siguen usando porque son el mecanismo de autenticación más “barato”.

_ De forma genérica, el ataque a las contraseñas consiste en crear una suposición de clave de password guess, es decir, suponer que es determinada clave, luego cifrar esa clave o representarla con lo que utiliza aquello que queremos romper, comparar los hashes de aquello que calculamos respecto a lo que tenemos y si coinciden significan que encontramos las contraseñas y si no iterar nuevamente hasta encontrarla.

Buenas prácticas: como decimos que las contraseñas son el eslabón más débil debemos aplicar una serie de buenas prácticas, de forma análoga, decimos que las contraseñas son como la ropa interior donde no deberíamos dejarla donde todo el mundo debería verla, deberíamos cambiarla regularmente, y no la deberíamos prestar a extraños.

Tipos de ataques

Password Guessing: este es el más simple y consiste en adivinar de algún modo la contraseña en función de conocer alguna información respecto del usuario, lugar o del servicio. Algunos ejemplos de contraseñas que podemos adivinar son en blanco, contraseñas por defecto(se puede usar una Default Password List como http://www.phenoelit.org/dpl/dpl.html de distintos dispositivos), usar o probar las palabras “password”, “passcode”, “admin”, también probar con una fila de teclas del teclado como qwerty, asdf, también el nombre del usuario o algún conocido suyo, el número de teléfono, dirección, DNI, etc. Entonces en base a lo que uno conoce de la persona por ahí se puede descubrir la contraseña.

Ataque de diccionario: aprovecha el hecho de que los usuarios normalmente utilizan contraseñas débiles (como palabras que pueden encontrarse en un diccionario), para probar sucesivamente con todas las palabras de una lista extensiva (llamada “diccionario”) hasta hallar la contraseña. Lo que hace es utilizar una lista de palabras (diccionario) con las cuales se prueba si son la contraseña, entonces este ataque se enfoca en probar de a una todas las palabras que están en esa lista que es el diccionario y que tienen mayor probabilidad de ocurrir, y esto se hace porque como tenemos que probar de a una todas las opciones, probamos con aquellas que son más probables y usamos todas las combinaciones de la lista, por lo que esto suele tener un gran porcentaje de efectividad, y es más rápido que otro tipo de ataques. Al diccionario lo podemos crear.

Ataque fuerza bruta: consiste en probar “todas” las posibles contraseñas hasta hallar la buscada. En teoría, si no hay un límite en el número de intentos, un ataque de fuerza bruta va a encontrar siempre la contraseña buscada. El problema es que a medida que la longitud de la clave aumenta, también la cantidad de intentos necesarios para hallarla y el tiempo que conlleva. En la práctica es difícil utilizar este método por el poder computacional y tiempo que demandan, ya que puede llevar de horas a siglos. Existen técnicas llamadas “smart brute force” o “hybrid password cracking” que comienzan probando con palabras de diccionario más probables, luego con variaciones de estas palabras y si esto falla finalmente probamos con el resto de combinaciones.

Ataque pre-computo: como una evolución de los ataques de fuerza bruta surgieron los de pre-computo que consisten básicamente en precalcular todas las posibles combinaciones de un algoritmo determinado y almacenarlas en una base de datos key-value que pueda ser luego consultada rápidamente, utilizando lo que llamamos Tablas Rainbow, donde nos permite calcular una sola vez todo, y usarlo N veces después, es decir, es como si

hiciéramos una ataque de fuerza bruta pero automático porque ya tenemos todo pregrabado. Hay herramientas que parten de un universo de combinaciones posibles entre cientos de equipos que se presten para esto, entonces al primer millón de combinaciones se la damos al equipo A, al segundo millón se lo damos al equipo B y de ese modo se hace un cálculo distribuido y reporta al sitio central los resultados del cálculo, luego baja al próximo y sigue procesando, y de ese modo se logra una terrible capacidad de cómputo y calculo por el hecho de ser distribuido entre miles de equipos, donde todo esto se va guardando y se arma la tabla correspondiente. Si uno tiene la tabal lista entonces uno encuentra la clave en segundos. Tiempo de éxito de varios ordenes de magnitud menos que la fuerza bruta. Existen sitios donde no solo se pueden bajar las tablas Rainbow en forma gratuita sino que también hay sitios que las tienen publicadas online sin la necesidad de bajarse la tabla:

http://ophcrack.sourceforge.net/tables.php

https://freerainbowtables.com/

http://project-rainbowcrack.com/table.htm

_ Sitios con tablas Rainbow on-line:

http://www.onlinehashcrack.com/

http://crackstation.net/

http://online.crackmyhash.com/

Herramientas

_ En el caso de la distribución Kali Linux, este tiene todo el set de herramientas de seguridad, que permiten distintas técnicas para descubrir contraseñas:

THC Hydra: cracker de mecanismos de autenticación de red, tales como telnet, ftp, http, smtp, etc. Por ejemplo, si queremos adivinar la contraseña de una cuenta de correo, entonces configuramos cual es el servidor de correo, cual es el usuario que ya conocemos y por ejemplo cargamos un diccionario para probar todas las combinaciones de la contraseña de esa cuenta, entonces cuando lanzamos el ataque la herramienta va a ir probando uno a uno las combinaciones de las palabras del diccionario, podemos poner un delay de tiempo de espera para que no sea bloqueada la cuenta, y con el tiempo y la suerte suficiente la herramienta va a encontrar la contraseña.

Medusa: un network login brute-forcer paralelo para distintos servicios de red (HTTP, FTP, IMAP, SMB, etc). Esta puede iniciar distintas sesiones y prueba en una sesión las 100 primeras combinaciones y en otra las 200 siguientes y de ese modo va optimizando el tiempo.

John the Ripper: es la herramienta por excelencia más potente para UNIX/Linux que realiza cracking con contraseñas simples, listas de palabras, fuerza bruta y/o smart brute force.

RainbowCrack: un cracker de contraseñas de fuerza bruta que utiliza tablas Rainbow. Para usarla vamos a los sitios de las tablas Rainbow, bajamos la tabla correspondiente de lo que queremos crakear por ejemplo Windows XP o Windows Vista y con esta herramienta leemos la tabla y probamos la contraseña.

Brutus: es para hacer password guessing de forma remota para Windows. Soporta todo tipo de servicios de red como HTTP, POP3, SMTP, IMAP, etc.

Defensas

_ De acuerdo a lo que vimos, respecto a las contraseñas, tenemos algunas defensas o buenas prácticas a seguir:

- NO usar contraseñas. Es decir, siempre que haya otro mecanismo más seguro o disponible complementario, utilizarlo. Por ejemplo, la mayoría de los servicios actuales como Gmail, Outlook, Facebook, Twitter, ofrecen un mecanismo de autenticación de dos factores, es decir, que requieren dos cosas para acceder como primero la contraseña y además un token generado de distintas formas como una notificación por mail o un mensaje sms, donde si nos descubren la contraseña igual no van a poder acceder porque les falta el otro elemento de validación.
- Si no queda otra que usar contraseñas entonces debemos definir políticas de contraseñas fuertes, como por ejemplo definir una longitud mínima, no utilizar una palabra de diccionario comunes, cambiarlas cada 30 o 60 días, etc.
- Utilizar “passphrases” en vez de “passwords”, o las primeras letras de una frase recordable, para generar una cadena que no es palabra de diccionario teniendo bastante dificultad.
- Utilizar herramientas que eviten la utilización de contraseñas débiles, ya sea mediante políticas o librerías que lo requieran.
- EDUCAR en todo momento a los usuarios, ya que un usuario puede ser nuestro principal aliado o enemigo, entonces educándolos les damos las herramientas que necesitan para poder manejarse de forma segura, y así lograr que el entorno o ecosistema en general sea más seguro y no prosperen ataques o sea más dificultoso.

Advanced Persistent Threats

Definición

_ Una amenaza persistente avanzada (Advanced Persistent Threat o APT) es un ataque cibernético prolongado y dirigido en el que un intruso obtiene acceso a una red y permanece sin ser detectado por un período de tiempo. La intención de un ataque APT generalmente es monitorear la actividad de la red y robar datos en lugar de causar daños a la red u organización. Estos ataques generalmente apuntan a organizaciones en sectores como la defensa nacional, la industria manufacturera y la industria financiera, ya que esas empresas manejan información de alto valor, incluida la propiedad intelectual, planes militares y otros datos de gobiernos y organizaciones empresariales. El objetivo de la mayoría de estos ataques es lograr y mantener el acceso continuo a la red objetivo en lugar de entrar y salir lo más rápido posible. Debido a que se requiere una gran cantidad de esfuerzo y recursos para llevar a cabo ataques APT, los piratas informáticos suelen apuntar a objetivos de alto valor. Para obtener acceso, los grupos de APT usan métodos avanzados de ataque, que incluyen exploits avanzados de vulnerabilidades de cero day, así como el spear phishing y otras técnicas de ingeniería social para así mantener el acceso a la red objetivo sin ser descubierto. Algunas APT son tan complejas que requieren administradores de tiempo completo para mantener los sistemas y el software comprometidos en la red objetivo. Aunque estos ataques pueden ser difíciles de identificar, el robo de datos nunca es completamente indetectable.

Características

Atacante: este selecciona objetivos con base en intereses políticos, comerciales o de seguridad y tiene una definición clara de la información que busca obtener de la víctima. Para evadir la detección, los hackers mantienen un perfil bajo dentro del ambiente de TI de las organizaciones que infiltran, incluso pueden llegar a esperar meses enteros para que se den las condiciones óptimas para un ataque.

Persistencia: si un objetivo se resiste a ser penetrado, el hacker no abandonará la misión, lo que hará es cambiar la estrategia y desarrollará un nuevo tipo de ataque.

Control y enfoque: una APT está enfocada en tomar control de elementos cruciales de la infraestructura, como redes de distribución eléctrica o sistemas de comunicaciones; también busca comprometer la propiedad intelectual de otros o información de seguridad nacional, mientras que los datos personales no suelen ser de interés para un atacante de este estilo.

Tiempo y dinero: los perpetradores de una APT no suelen preocuparse por el costo del ataque, incluso pueden no preocuparse de los ingresos a partir del mismo, ya que a menudo están financiados por estados nacionales o por el crimen organizado.

Automatización: los hackers hacen uso de software y sistemas automatizados para aumentar el poder de penetración contra un solo objetivo. Muchas veces utiliza múltiples vectores de ataque simultáneos, tanto automatizados como humanos.

Una sola capa: solo un grupo u organización posee y controla todos los roles y responsabilidades durante el ataque. Estos roles y responsabilidades no están distribuidos en grupos externos a la organización atacante.

Redes sociales: es muy común el uso de herramientas de redes sociales, como invitaciones falsas de Linkedln, para ganar la confianza de las víctimas y comprometer así los sistemas y las credenciales de acceso a los mismos.

Complejidad de una APT

_ Es difícil detectar una APT por las siguientes razones:

- Más que tomar control de las aplicaciones y de la infraestructura de la red, buscan aprovecharse de los recursos y privilegios de las personas que forman parte de la organización.
- Usan firmas de ataque únicas y de gran creatividad.
- El comportamiento y las "firmas" de un ataque de este tipo son difíciles de correlacionar con los de ataques conocidos, aun si la empresa utiliza un SIEM (Security Incident and Event Management).
- Normalmente una APT es distribuida a lo largo de periodos de tiempo prolongados, haciéndola difícil de correlacionar con base en los datos de fecha y hora.
- Los ataques parecieran venir de una gran variedad de fuentes, haciendo muy difícil la identificación de la red hostil.
- El tráfico de datos del ataque por lo general se encubre a través de cifrado, compresión o enmascarando las transmisiones dentro del comportamiento "normal" de programas comprometidos.
- Los ataques APT suelen diseñarse para evadir las soluciones antimalware y los IPS, además de que pueden ser compilados para una industria u organización específica.

Ciclo de vida de APT

_ El ciclo de vida de una APT es mucho más largo y complejo que otros tipos de ataques:

1)_ Se define el objetivo, en donde se determina a quién se dirige, qué se espera lograr y por qué.

2)_ Se seleccionan los miembros del equipo, se identifican las habilidades necesarias y se busca un acceso interno.

3)_ Se buscan las herramientas, ya sea, disponibles actualmente o se crean nuevas aplicaciones para obtener las herramientas adecuadas para el trabajo.

4)_ Se estudia el objetivo, se busca a quién tenga el acceso, también qué hardware y software utiliza el mismo y cómo diseñar mejor el ataque.

5)_ Prueba de detección, donde se implementa una pequeña versión de reconocimiento del software, probando las comunicaciones y alarmas, identificando los puntos débiles.

6)_ Despliegue, donde comienza la infiltración.

7)_ Una vez que esté dentro de la red, se averigua a dónde ir y encuentre su objetivo.

8)_ Cuando el objetivo es adquirido, se solicita la evacuación creando un túnel para comenzar a enviar datos desde el objetivo.

9)_ Se amplía el acceso y se obtienen credenciales, donde se crea una "red fantasma" bajo su control dentro de la red de destino, aprovechando su acceso para ganar más movimiento.

10)_ Se aprovechan otras vulnerabilidades para establecer más zombis o extender su acceso a otras ubicaciones valiosas.

11)_ Una vez que se encuentre lo que estaba buscando, se extraen los datos a la base.

12)_ Se deben cubrir pistas y hacer que no lo detecten, se debe permanecer oculto en la red, además permanecer indetectable a los controles de sigilo y luego limpiar todo rastro de intrusión.

Señales de un ataque APT

_ Ante un ataque de este tipo, se suelen presentar las siguientes señales:

- Actividad inusual en las cuentas de usuario, como un aumento en los inicios de sesión de alto nivel o grandes transferencias de datos fuera del horario normal de oficina o en ubicaciones inusuales.
- Incrementos repentinos en el tráfico de red, sobre todo en las transferencias salientes.
- Presencia generalizada de troyanos de puerta trasera (backdoor).
- Paquetes de datos inesperados o inusuales, que pueden indicar que los datos se han acumulado en preparación para la exfiltración.
- Anomalías en los datos salientes o un aumento repentino e inusual en las operaciones de la base de datos que involucran cantidades masivas de datos.
- Consultas repetidas a nombres DNS dinámicos.
- Búsquedas inusuales de directorios y archivos de interés para un atacante, por ejemplo, búsquedas en repositorios de código fuente.

- Archivos de salida grandes no reconocidos que se han comprimido, cifrados y protegidos con contraseña.
- Detección de comunicaciones hacia/desde direcciones IP falsas.
- Cambios inexplicables en las configuraciones de plataformas, enrutadores o firewalls.
- Mayor volumen de eventos/alertas de IDS.

Defensa y protección

_ Las defensas deben incluir múltiples herramientas y técnicas de seguridad, estas incluyen:

Filtrado de correo electrónico: como la mayoría de los ataques APT aprovechan el phishing para obtener acceso inicial, se deben filtrar correos electrónicos y bloquear enlaces o archivos adjuntos maliciosos dentro de los mismos.

Protección de endpoints: como todos los ataques APT implican la toma de control de dispositivos endpoints, la protección antimalware avanzada y la detección y respuesta de endpoints pueden ayudar a identificar y reaccionar ante un endpoint comprometido.

Control de acceso: las medidas de autenticación sólidas y la administración cercana de las cuentas de usuario, con enfoque en las cuentas con privilegio, pueden reducir los riesgos de APT.

Monitoreo del tráfico, comportamiento de usuarios y entidades: puede ayudar a identificar penetraciones, movimientos laterales y exfiltración en diferentes etapas de un ataque APT. Monitoreo del tráfico entrante y saliente de la red, sobre todo el saliente.

Adquirir o incorporar a la infraestructura de seguridad: existen soluciones anti-APTs de en el mercado como FireEye, Trend Micro, McAfee, Palo Alto, Kaspersky Labs, etc.

Promover la importancia de la seguridad dentro de nuestra organización: mediante formación interna, cursos online, capacitación constante, etc.

Mitigaciones

_ La Dirección de Seguridad de la Información de los EEUU (IAD) redactó una publicación con sus Estrategias de Mitigación para la Seguridad Informática con referencias a las APTs. Se destacan cuatro áreas clave:

- Integridad del dispositivo
- Contención de daños
- Protección de cuentas
- Transporte seguro y disponible.

Casos resonantes

- Stuxnet
- Operación Aurora
- GhostNet
- MACHETE
- APT38

Firewalls

Definición

_ Un firewall o “cortafuegos” es un sistema diseñado de seguridad para prevenir el acceso no autorizado hacia o desde una red privada. Se puede implementar en forma de hardware, de software o en una combinación de ambos, en donde poder determinar la diferencia entre estos, lleva a una mejor comprensión de los mecanismos necesarios para protegerse mejor en un ambiente privado, y en uno público. Es importante recordar que representan una primera línea de defensa porque pueden evitar que un programa malicioso o un atacante obtengan acceso a su red y a su información antes de que se produzca cualquier posible daño ya que examinan los mensajes que ingresan o egresan de una red local y bloquean aquellos que sean necesarios, impidiendo que los usuarios no autorizados accedan a redes privadas conectadas a Internet, especialmente a intranets. Además se debe comprender que para que el firewall pueda ser efectivo, todo el tráfico que haya entre dos redes debe pasar por él, ya que de este modo, podrá aplicar las políticas que se hayan definido.

Funciones

_ A continuación tenemos las funciones específicas de los firewalls:

- Crear una barrera que permita o bloquee intentos para acceder a la información en su equipo.
- Evitar que usuarios no autorizados accedan a los equipos y las redes de la organización que se conectan a Internet.
- Supervisar la comunicación entre equipos y otros equipos en Internet.
- Visualizar y bloquear aplicaciones que puedan generar riesgo.
- Advertir de intentos de conexión desde otros equipos.
- Advertir de intentos de conexión mediante las aplicaciones en su equipo que se conectan a otros equipos.
- Detectar aplicaciones y actualizar rutas para añadir futuras fuentes de información.
- Hacer frente a los cambios en las amenazas para la seguridad.

_ Los firewalls no se auto configuran y tampoco brindan una protección total frente a ataques, sino que son parte de una estrategia de seguridad, de manera que la mejor forma de asegurarnos contra softwares maliciosos, es con una combinación de estos. También es necesario resaltar que no protege amenazas o ataques de usuarios negligentes ni protege ataques que no pasen por medio del firewall ni la copia de datos importantes si se ha accedido a ellos, tampoco protege de ataques de ingeniería social, de modo que si bien la protección por firewall es una excelente primera línea de defensa, también es buena idea seguir estos consejos importantes para proteger sus datos y sus dispositivos:

- No hacer click en enlaces ni abra archivos adjuntos de personas que no conoce.
- Ser consciente de que cada nuevo dispositivo conectado a internet que trae a su hogar es una posible vía de ataque. Hay que asegurarse de restablecer las contraseñas predeterminadas y mantenga esos dispositivos actualizados con las últimas versiones del fabricante.

_ La motivación de los Firewalls es tratar de proteger los datos, los recursos y la reputación contra ataques de intrusos, negaciones de servicio y robos de información.

Ventajas y desventajas

_ La ventaja más importante es la protección contra amenazas externas. Además el respectivo administrador de red puede tener un mayor control sobre la seguridad de la misma, también puede determinar los puertos específicos que deben recibir o enviar datos relacionados con varias tareas. Estas conllevan los siguientes beneficios:

- Controlar el tráfico
- Mayor privacidad
- Control de acceso
- Proteger la red de troyanos

_ Si bien pueden bloquear un potencial acceso de humanos, no pueden defendernos de las amenazas que existen en forma de malware como virus. Además, lo más probable es que todo el sistema informático pueda afectar incluso si el Firewall está activo y en ejecución. Por esta misma razón, se podría decir que su Firewall no es la herramienta de seguridad más completa. Entonces, lo mejor es tomar su Firewall como una “medida de seguridad” pero no como un sistema informático sofisticado. Para agregar, también se encuentran los siguientes inconvenientes a la hora de utilizar un firewall:

- Costo: se considera una inversión inicial que varía según el tipo del firewall.
- Rendimiento: los firewalls basados en software pueden limitar el rendimiento general de su computadora y ralentizarlo.

- Acceso restringido para usuarios: este puede ser bastante molesto ya que si se trata de un usuario avanzado, tendrá que enfrentarse a diversas cargas para lograr tareas avanzadas.
- Indefenso ante ataques malware: los firewalls pueden bloquear troyanos, pero no son efectivos contra virus y otro malware ya que pueden ingresar al sistema camufladas, entonces, incluso si se tiene un firewall, el ataque puede ocurrir.
- Complejidad: en caso de ser una organización grande, se requiere personal dedicado para realizar la instalación y mantenimiento del mismo, por lo cual se pueden asumir costos extras.

Tipos de firewalls

_ Existen diversos tipos de cortafuegos pero en primera instancia se podrían clasificar según rasgos generales en tres categorías. La presencia de uno en la red no es excluyente para los otros, es decir, se puede contar con un firewall de hardware y uno de software, en donde esta fórmula es recomendada ya que potencia y refuerza el esquema de seguridad.

Firewall de hardware: vienen incluidos en algunos enrutadores y requieren poca o ninguna configuración, ya que están incorporados en su hardware. Monitorean el tráfico de todas las computadoras y dispositivos que están conectados a la red de dicho enrutador, lo que significa que podemos filtrar el acceso a todos ellos solo con una pieza de equipo. Los firewalls de hardware brindan seguridad esencial para el Internet de las cosas (IoT).

Firewall de software: ayudan a mantenerse protegido en lugares públicos. Se ejecutan como un programa en las computadora o dispositivo y observan de cerca el tráfico de la red para ayudar a interceptar programas maliciosos antes de que lleguen al equipo.

Firewall as a Service o Cloud Firewall: sigue el mismo principio de funcionamiento que el firewall de software, solo que vive en la nube como es Amazon, Azure, etc. La diferencia entre el firewall de software y cloud, es que con cloud, ya se tiene preconfigurado y listo para usar. Además cuenta con el beneficio de que pueden crecer a la misma magnitud que la organización y, de manera similar a los firewalls de hardware, funcionan bien con la seguridad perimetral. A su vez, también se pueden clasificar según su estructura y funcionalidad. Entre los cuales se pueden mencionar:

- Packet-filtering firewalls o firewalls de filtrado de paquetes: funcionan en el router analizando la cabecera de cada paquete y comparando cada paquete recibido con un conjunto de criterios establecidos antes de que cada uno sea enviado o suprimido. Las reglas en este tipo de dispositivos se basan en el protocolo de transporte (tcp, udp, icmp), en las direcciones origen y destino del paquete y en el puerto destino (Telnet, ftp, http).

- Circuit-level gateways: una puerta de enlace de nivel de circuito opera en la capa de transporte de los modelos de referencia de Internet o OSI y, como su nombre lo indica, implementa el filtrado a nivel de circuito en lugar del filtrado a nivel de paquete. Este comprueba la validez de las conexiones (es decir, circuitos) en la capa de transporte (generalmente conexiones TCP) contra una tabla de conexiones permitidas, antes de que se pueda abrir una sesión e intercambiar datos.
- Stateful inspection firewalls: estos combinan la tecnología de inspección de paquetes y la verificación de protocolo de enlace TCP para crear un nivel de protección mayor que cualquiera de las dos arquitecturas anteriores podría proporcionar por sí solo. Estos no sólo examinan cada paquete, sino que también hacen un seguimiento de si dicho paquete forma parte o no de una sesión TCP establecida.
- Application-level gateways: puerta de enlace de nivel de aplicación o Proxy. Este tipo de firewalls operan en la capa de aplicación del modelo OSI, filtrando el acceso según las definiciones de la aplicación. Se considera como uno de los firewalls más seguros disponibles, debido a su capacidad para inspeccionar paquetes y garantizar que se ajusten a las especificaciones de la aplicación.
- Los firewalls de traducción de direcciones de red (NAT): estos permiten que múltiples dispositivos con direcciones de red independientes se conecten a Internet utilizando una sola dirección IP, manteniendo ocultas las direcciones IP individuales.
- Multilayer inspection firewalls: los firewalls de inspección multicapa combinan el filtrado de paquetes con la monitorización de circuitos, a la vez que permiten conexiones directas entre los hosts locales y remotos, que son transparentes para la red.
- Firewall UTM (Unified Threat Management): se caracteriza por combinar diferentes elementos de los dos anteriores firewalls (proxy y stateful). Lamentablemente tiene el mismo problema que el firewall proxy, puede convertirse en un cuello de botella si no se tiene la capacidad y/o rendimiento adecuado.
- Próxima generación o NGFW: las características que definen a los firewall de próxima generación son la identificación y control de aplicaciones, autenticación basada en el usuario, protección contra malware, protección contra exploits, filtrado de contenido (incluido el filtrado de URL) y control de acceso basado en la ubicación.
- NGFW centrado en amenazas: este tipo de firewall ofrece las mismas características del anterior, con la diferencia que detecta y corrige amenazas.

Arquitectura de redes

_ Al hablar de arquitectura de firewall, nos referimos a aquellas representaciones físicas y lógicas en torno al posicionamiento de los activos computacionales. Por lo cual, se trata de una edificación que, a través de su diseño y planificación, sirve para implementar la estructura de la red en cuestión, con lo cual, aprobará o denegará el tráfico de los elementos apropiados, una vez canalizado.

Firewall de filtrado de paquetes o Arquitectura Screening Router: es el modelo más antiguo y más simple de implementar, y consiste en un dispositivo capaz de filtrar paquetes. Se basa en aprovechar la capacidad de algunos routers para efectuar un enrutamiento selectivo y así, restringir o admitir el tránsito de paquetes por medio de listas de control de acceso en función de algunas particularidades.

Arquitectura Screened Host: posee un firewall compuesto por un router para el filtrado de paquetes y un host bastión para el filtrado de conexiones a nivel de circuito y aplicación. La primera línea de protección corresponde al router con filtrado de paquetes, el host bastión se encuentra conectado a la red interna como un host más.

Arquitectura Dual-Homed Host: se trata de un host que cuenta con dos tarjetas de red y cada una de ellas, se conecta a una red diferente. Se compone de simples máquinas Unix, denominadas anfitriones de dos bases, equipadas con dos tarjetas de red, donde una se conecta a la red interna a proteger y la otra a la red externa. De este modo el tráfico entre la red interna y el exterior está completamente bloqueado.

Arquitectura Screened Subnet: es la arquitectura más popular. Agrega un nivel extra de seguridad que la arquitectura Screened host, agregando un perímetro a la red, que aísla fuertemente la red interna de Internet. Los hosts bastiones son las máquinas más vulnerables en la red. Aunque se esfuerce en protegerse, estas son las máquinas que pueden ser atacadas, porque ellas son las máquinas que son vistas por la red externa.

Funcionamiento de las reglas de filtrado

_ Lo que les da su utilidad a los firewalls como herramienta de protección es la capacidad de poder discriminar en el tráfico entrante a la red. Esto se realiza mediante las reglas de filtrado. Existen tres tipos de reglas:

- Permitir: el tráfico marcado como permitido podrá entrar atravesar el firewall.
- Denegar: el tráfico denegado no podrá pasar, pero se le devolverá un error de “Unreachable”.
- Descartar: el tráfico marcado como descartado no podrá acceder ni se le devolverá nada.

_ Luego, cada regla tiene además un conjunto de parámetros contra los cuales se evaluará cada paquete recibido.

Seguridad en Redes Wireless

Redes de comunicación inalámbrica

_ Una Wireless Network es aquélla que se lleva a cabo sin el uso de cables de interconexión entre los participantes de una comunicación, es decir, se utilizan ondas de radio para conectar los dispositivos y tanto la transmisión como la recepción de información se realiza a través de puertos; por ejemplo una comunicación entre teléfonos móviles es inalámbrica, mientras que una comunicación con teléfono fijo tradicional no lo es. Algunos de los dispositivos que comúnmente utilizan las redes inalámbricas son las computadoras de escritorio y portátiles, teléfonos móviles o celulares, televisores, tablets, impresoras, reproductores de multimedia, periféricos (auriculares, mouse, teclado), dispositivos localizadores, etc. Estos dispositivos son susceptibles de intercomunicarse entre sí y, aunque pueden hacerlo por los sistemas de cables tradicionales, su mayor potencial se alcanza a través de las comunicaciones inalámbricas.

_ Las redes inalámbricas funcionan de manera similar a las redes cableadas, sin embargo, estas deben convertir las señales de información en una forma adecuada para la transmisión a través del medio de aire. Estas tienen y sirven a muchos propósitos:

- En algunos casos se utilizan para sustituir las redes cableadas.
- Se utilizan para proporcionar acceso a datos desde ubicaciones remotas. Lo bueno, es que la infraestructura inalámbrica puede ser construida a bajo coste en comparación con las alternativas cableadas o alámbricas tradicionales.
- Permiten a los dispositivos remotos que se conecten sin dificultad, independientemente que estos dispositivos estén a unos metros o a varios kilómetros de distancia, y todo esto se puede hacer sin la necesidad de romper paredes para pasar cables o instalar conectores, donde justamente esto hizo que el uso de esta tecnología sea muy popular, y se expandiera rápidamente.

_ Por otro lado, tenemos algunas cuestiones relacionadas con la regulación legal del espectro electromagnético. Tenemos en cuenta que las ondas electromagnéticas que se transmiten a través de muchos dispositivos, son propensas a la interferencia, por ende, todos los países necesitan regulaciones que definan los rangos de frecuencia y potencia de transmisión permitidos para cada tecnología, en el caso de Argentina el ente encargado de estas regulaciones es el ENACOM. Además, las ondas electromagnéticas no se pueden confinar fácilmente a un área geográfica limitada, y es por esta razón, que un hacker por ejemplo puede escuchar fácilmente a una red si los datos transmitidos no están codificados, por lo tanto, se deben tomar todas las medidas necesarias para garantizar la privacidad de los datos transmitidos a través de redes inalámbricas.

Tipos de redes inalámbricas

_ Las redes inalámbricas se pueden clasificar en distintos grupos según el área de aplicación y el alcance de la señal:

Redes inalámbricas de área corporal o WBAN (Wireless Body Area Network): cubren distancias de 1 o 2 metros. Esta se realiza entre dispositivos de baja potencia utilizados en el cuerpo humano, consistiendo en un conjunto móvil y compacto de comunicación como por ejemplo micrófonos, auriculares, sensores que controlan los parámetros vitales del cuerpo y movimientos, y transmiten datos de forma inalámbrica desde el cuerpo a una estación base, en donde los datos pueden ser remitidos a un hospital, clínica o a otro lugar, en tiempo real, etc.

Redes inalámbricas de área personal o WPAN (Wireless Personal Area Network): son aquellas que cubren distancias inferiores a los 10 metros. A diferencia de otras redes inalámbricas, una conexión realizada a través de una está implica, por lo general, poca o ninguna infraestructura o conectividad directa fuera del enlace establecido. Este tipo de redes se caracterizan por su bajo consumo de energía y también una baja velocidad de transmisión. Estas soluciones están pensadas para interconectar los distintos dispositivos de un usuario (por ejemplo, el ordenador con la impresora).

Redes inalámbricas de área local o WLAN (Wireless Local Area Network): estas redes están diseñadas para proporcionar acceso inalámbrico en zonas con un rango típico de hasta 100 metros y se utilizan sobre todo en el hogar, la escuela, una sala de computadoras, o entornos de oficina. Permite reemplazar los cables que conectan a la red los PCs, portátiles u otro tipo de dispositivos proporcionando a los usuarios la capacidad de moverse dentro de un área de cobertura local y permanecer conectado a la red para poder transmitir y recibir voz, datos, vídeo dentro de edificios, entre edificios o campus universitarios e inclusive sobre áreas metropolitanas. Las WLAN se basan en el estándar 802.11 del IEEE y son comercializadas bajo la marca Wifi.

Redes inalámbricas de área metropolitana o WMAN (Wireless Metropolitan Area Network): estas pretenden cubrir el área de una ciudad o entorno metropolitano, pudiendo extenderse hasta 50 km. Podemos considerarla como una red LAN extensa o una red WAN de menor tamaño. Básicamente lo que hace es interconectar redes WLAN unas con otras y ampliar así el rango de acción. Incluso podría ser utilizado en zonas de difícil acceso, como pueden ser lugares remotos dentro de un municipio, zonas rurales, etc. Las WMAN se basan en el estándar IEEE 802.16, a menudo denominado WiMAX (Worldwide Interoperability for Microwave Access).

Redes inalámbricas de área extensa o WWAN (Wireless Wide Area Network): también conocidas como de área global o WGAN, pueden cubrir toda una región (país o grupo de países), abarcando miles de kilómetros, y permitiendo la interconexión de varios sistemas de comunicaciones ayudando a que ésta sea cada vez más globalizada. Estas redes se

basan en tecnología celular y han aparecido como evolución de las redes de comunicaciones de voz. Éste es el caso de las redes de telefonía móvil conocidas como la primera generación 1G, la segunda generación 2G, la generación 2.5G (2G + GPRS), la generación 3G, la generación 3.5G, la cuarta generación 4G y por último la generación 5G. Pero también existen opciones satelitales mucho más económicas para usuarios residenciales o para pequeñas oficinas, o gente que esté en zonas remotas como islas, debido a su gran altura, donde las transmisiones por satélite pueden cubrir una amplia área sobre la superficie de la tierra.

Otras:

- Redes inalámbricas de área regional (WRAN), están constituidas por una integración de las redes WLAN y WMAN, bajo el estándar IEEE 802.22. Son un proyecto de solicitud de autorización (PAR) aprobadas por el IEEE-SA, cuya función es desarrollar un estándar para la radio cognitiva basada en las capas PHY/MAC/(interfaz de aire), para el uso de una licencia exenta de los dispositivos, para no interferir en el espectro que se asigna a la emisión del servicio de TV.

Arquitectura de tecnologías inalámbricas

_ A continuación definimos diversos términos utilizados en una arquitectura de red inalámbrica. La arquitectura lógica del estándar 802.11 contiene varios componentes principales:

Estación (Station - STA): puede ser una PC, un ordenador portátil, una PDA, un teléfono o cualquier dispositivo que tenga la capacidad de interferir en el medio inalámbrico.

Punto de acceso (Access Point - AP): también llamado estación base (BS), es un dispositivo que permite a los dispositivos inalámbricos que se conecten a una red cableada mediante Wifi, o estándares relacionados.

Conjunto de servicios básicos (Basic Service Set - BSS): consiste en un punto de acceso, junto con todas las estaciones asociadas. El punto de acceso actúa como un maestro para controlar las estaciones dentro de ese BSS. El BSS más simple se compone de un AP y una STA.

Conjunto de servicios extendidos (Extended Service Set - ESS): conjunto de uno o más conjuntos interconectados de servicios básicos (BSS) que aparecen como un solo BSS a la capa de control de enlace lógico de cualquier estación asociada con una de esas BSS.

BSS independiente (Independent Basic Service Set - IBSS): cuando todas las estaciones en el conjunto de servicios básicos son estaciones móviles y no hay conexión a una red cableada. Es una red ad hoc que no contiene puntos de acceso, lo que significa que no pueden conectarse a cualquier otro conjunto de servicios básicos.

Sistema de distribución (Distribution System - DS): es el mecanismo por el cual diferentes puntos de acceso pueden intercambiar tramas entre sí o bien con las redes cableadas, si las hubiera. El DS no es necesariamente una red y el estándar IEEE 802.11 no especifica ninguna tecnología en particular para este. En casi todos los productos comerciales se utiliza Ethernet por cable como la tecnología de red troncal.

_ Por otro lado, existen dos modos de configurar una red inalámbrica:

Modo Ad hoc: todos los dispositivos de la red se comunican directamente entre sí, de igual a igual, en el modo de comunicación punto a punto. No se requiere ningún punto de acceso para la comunicación entre dispositivos. Es el más adecuado para un pequeño grupo de dispositivos que se encuentren presentes y físicamente muy cerca entre sí. El rendimiento de la red sufre si el número de dispositivos aumenta. El modo ad hoc funciona bien en un entorno pequeño siendo la forma más fácil y menos costosa de configurar una red inalámbrica.

Modo Infraestructura: todos los dispositivos de la red están conectados con la ayuda de un punto de acceso. Los puntos de acceso inalámbricos son generalmente routers o switches que pasan los datos de la red inalámbrica a datos en una Ethernet cableada, actuando como un puente entre la LAN cableada y los dispositivos inalámbricos. Los clientes inalámbricos pueden moverse libremente del dominio de un punto de acceso a otro y seguir manteniendo la conexión de red sin cortes. Este modo ofrece una mayor seguridad, facilidad de gestión, y más escalabilidad y estabilidad, pero incurre en un costo adicional debido al despliegue de puntos de acceso.

Estándares y tecnologías inalámbricas

_ A continuación realizamos una comparación entre todos los tipos de redes y algunas de sus tecnologías:

Tipo de red	Nombre	Estándar	Banda de frecuencia	Rango nominal	Máxima Velocidad. Transmis.
WPAN	Bluetooth	IEEE 802.15.1	2.4 GHz	10 m:	720 Kbps
	IrDA	IrDA	Ventana Infrarrojo 850-900 nm longitud de onda	1 m	16 Mbps
	ZigBee	IEEE 802.15.4	868 MHz, 900 MHz, 2.4 GHz	10 m	250 Kbps
	UWB	IEEE 802.15.3	3.1-10.6 GHz (USA) 3.4-4.8 GHz & 6-8.5 GHz (Europa)	10 m	480 Mbps
WLAN	Wi-Fi	IEEE 802.11	2.4 / 5 GHz	100 m	1 Mbps
		IEEE 802.11ª	5 GHz	100 m	48 Mbps
		IEEE 802.11b	2.4 GHz	100 m	11 Mbps
		IEEE 802.11g	2.4 GHz	100 m	54 Mbps
		IEEE 802.11n	2.4 / 5 GHz	250 m	600 Mbps
		IEEE 802.11ac	5 GHz	250 m	1.3 Gbps

WMAN	WiMAX	IEEE 802.16	2-11 GHz y10-66 GHz	50 km	70 Mbps
WWAN	Móvil	AMPS, GSM, GPRS, UMTS, HSDPA, LTE	700 MHz, 850 MHz, 900 MHz, 1800 MHz, 1900 MHz, 2100 MHz, 2600 MHz	> 50 km	1 Gbps
	Satélite	DVB-S2	3-30 GHz	> 50 km	60 Mbps

Ventajas y desventajas

_ Como ventajas tenemos:

- Fácil instalación y reducción de costes: como no es necesario tender cables, podemos adaptar la red a cualquier entorno, y como el medio de transmisión es el aire, está libre de agresiones físicas, y no se requieren obras de mantenimiento ni para ampliación o remodelación.
- Movilidad: se puede tener acceso a la información en cualquier punto dentro de la zona de cobertura del punto de acceso, conservando su acceso con todas las prestaciones.
- Escalabilidad: facilidad de expandir la red después de la instalación inicial. Se puede ajustar el tamaño de la red a nuestras necesidades.
- Uso del espectro libre: la mayor parte de las redes inalámbricas operan en un rango de frecuencias de uso libre, es decir, no están sujetas al pago de ningún tipo de licencias para su uso.
- Altas tasas de transmisión: se pueden alcanzar tasas de transmisión que llegan hasta los 54 Mbps.

_ Ahora como desventajas tenemos:

- Interferencias: debido al medio usado, es muy difícil darse cuenta que dispositivos son los que están produciendo interferencias. Además las interferencias pueden provenir de otras redes inalámbricas próximas.
- Cobertura: el radio de acción de una red inalámbrica está limitado por la potencia máxima que se puede radiar según la legislación vigente. Para extender la zona de acción de la red hay que colocar repetidores.
- Velocidad de transmisión: la transmisión inalámbrica puede ser más lenta y menos eficiente que las redes cableadas.
- Seguridad: se debe diseñar una red, con el nivel de seguridad más alto posible, ya que la transmisión inalámbrica es más vulnerable, y para así evitar que usuarios no autorizados tengan acceso a la red o realicen ataques.

Seguridad WI-FI

_ Las redes de telecomunicaciones sufren muchos y variados ataques, y van desde la intrusión de virus y troyanos hasta la alteración y robo de información confidencial. La seguridad es uno de los problemas más graves a los cuales se enfrenta actualmente la tecnología Wifi. La mayoría de las redes son instaladas por administradores de sistemas y redes, por su simplicidad de implementación sin tener en consideración la seguridad y, por ende, convirtiendo sus redes en redes abiertas, sin proteger la información que circula por ellas. Existen varias alternativas para garantizar la seguridad de estas redes.

Peligros y ataques

_ Las violaciones de seguridad en las redes wireless (WLAN) suelen venir de los puntos de acceso no autorizados (rogue AP), es decir, aquellos que son instalados sin el conocimiento del administrador del sistema, y son aprovechados por los intrusos que pueden llegar a asociarse al AP y así acceder a los recursos de la red. Algunos ataques son:

Warchalking y Wardriving: el primero hace referencia a la utilización de un lenguaje de símbolos para reflejar visualmente la infraestructura de una red inalámbrica y las características de alguno de sus elementos. Estas señales se suelen colocar en las paredes de edificios situados en las zonas en las que existen redes inalámbricas para indicar su condición y facilitar el acceso a las mismas. Y el Wardriving es la acción de ir recorriendo una zona en busca de la existencia de redes wireless y conseguir acceder a ellas.

Ruptura de la clave WEP: este mecanismo de seguridad especificado en el estándar 802.11 es el cifrado de la información, utilizando una clave simétrica denominada WEP, sin embargo, WEP tiene deficiencias, como la corta longitud de su clave o la propagación de la misma, que permiten acceder a redes protegidas solamente mediante WEP.

Suplantación: es un ataque en el que el intruso pretende tomar la identidad de un usuario autorizado. Una variante de este es la escucha o eavesdropping. Como las comunicaciones inalámbricas viajan libremente por el aire cualquiera que esté equipado con una antena que opere en el rango de frecuencias adecuado y dentro del área de cobertura de la red podrá recibirlas. Una técnica es el spoofing que consiste en que el intruso consigue suplantar la identidad de una fuente de datos autorizada para enviar información errónea a través de la red. Otra técnica es la captura de canales o hijacking, que sucede cuando un intruso se hace con un canal que, desde ese momento, ya no estará accesible para usuarios autorizados disminuyendo así las prestaciones de la red.

Denegación de servicio (DoS): ataques en los que el intruso consigue que los usuarios autorizados no puedan conectarse a la red. Hay algunos ataques de denegación de servicio como crear un nivel elevado de interferencias en una zona cercana al punto de acceso, ataques por sincronización (Smurf) donde el intruso envía un mensaje broadcast

con una dirección IP falsa que, al ser recibida, causa un aumento enorme de la carga de red.

Mecanismos de seguridad

_ La seguridad Wifi abarca dos niveles. En el nivel más bajo, se encuentran los mecanismos de cifrado de la información, y en el nivel superior los procesos de autenticación. Al igual que en el resto de redes, la seguridad para las redes wireless se concentra en el control y la privacidad de los accesos. Un control de accesos fuerte impide la comunicación entre los usuarios no autorizados a través de los AP. Por otro lado, la privacidad garantiza que solo los usuarios a los que van destinados los datos transmitidos los comprendan. Así, la privacidad de los datos transmitidos solo queda protegida cuando los datos son encriptados con una clave que solo puede ser utilizada por el receptor al que están destinados esos datos. La seguridad en las comunicaciones se describe a menudo en términos de tres elementos:

Autenticación: garantiza que los nodos son quién y lo que dicen ser. Se basa normalmente en demostrar el conocimiento de un secreto compartido, como la pareja nombre de usuario y contraseña.

Confidencialidad: (privacidad) asegura que los intrusos no pueden leer el tráfico de red. Típicamente, la confidencialidad se protege mediante el cifrado del contenido del mensaje, donde esté cifrado aplica un método reversible de transformación (llamado algoritmo de cifrado o encriptación) al contenido del mensaje original (llamado texto plano), codificándolo u ocultándolo para crear el texto cifrado. Y sólo los que saben cómo revertir el proceso (descifrar el mensaje) pueden recuperar el texto original.

Integridad: asegura que los mensajes son entregados sin alteración. Esta se refiere a la capacidad de asegurarse de que el mensaje recibido no ha sido alterado de manera alguna y que es idéntico al mensaje que se envió. Los bytes de la secuencia de verificación de trama (Frame Check Sequence - FCS) son un ejemplo de comprobación de integridad, pero no se consideran seguros.

_ La seguridad es siempre relativa, nunca absoluta. Para cada defensa, hay (o seguro habrá) un ataque exitoso, y para cada ataque, hay (o seguro habrá) una defensa exitosa. Cuanto mejor sea la defensa, más tiempo y esfuerzo se necesita para romperla. La defensa adecuada es aquella que está equilibrada y que coincide con el número esperado de ataques.

Criptografía aplicada a redes inalámbricas

_ El cifrado es opcional en las WLAN, pero sin él, cualquier dispositivo compatible con el estándar dentro del alcance de la red puede leer todo su tráfico. Principalmente ha habido tres métodos de encriptación para hacer seguras las redes WLAN. En orden cronológico de aparición, estos son:

WEP (Wired Equivalent Privacy): proporciona un cifrado a nivel 2, basado en el algoritmo de cifrado RC4 que utiliza claves de 64 bits (40 bits más 24 bits del vector de iniciación) o de 128 bits (104 bits más 24 bits del IV). Sigue siendo utilizado pero es muy vulnerable por lo que deberían ser actualizados aquellos sistemas basados en WEP.

WPA (Wifi Protected Access): la configuración más común es WPA - PSK (Pre-shared Key), donde PSK es una clave secreta compartida con anterioridad entre las dos partes usando algún canal seguro antes de que se utilice. Estos sistemas utilizan casi siempre algoritmos criptográficos de clave simétrica. Las claves son de 256 bits y de 8 o más caracteres de longitud y hasta un máximo de 63.

WPA2 (Wifi Protected Access, versión 2): uno de los cambios más significativo fue el uso obligatorio de los algoritmos AES y la introducción de CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol) como un reemplazo del TKIP. La principal vulnerabilidad requiere que el atacante ya tenga acceso a la red Wifi protegida con el fin de tener acceso a ciertas claves para luego perpetrar un ataque en contra de los dispositivos de la red.

WPA3 (Wifi Protected Access, versión 3): mayor protección ante ataques, incluso cuando no se cuenta con una contraseña fuerte. Cifrado de 192 bits para redes empresariales o personales, donde se traten datos confidenciales.

_ A continuación realizamos una lista de clasificación de métodos actuales de seguridad Wifi de mejor a peor:

1. WPA3
2. WPA2 + AES
3. WPA + AES
4. WPA + TKIP/AES (TKIP aparece como método alternativo)
5. WPA + TKIP
6. WEP
7. Red abierta (ningún tipo de seguridad)

VPNs - Virtual Private Networks

Definición

_ Una Red privada virtual (VPN, Virtual Private Network) es una red privada que utiliza la infraestructura de una red pública para poder transmitir información. Las VPN más comunes son las VPN físicas, como la red de área local (LAN). Sin embargo, cuando ya no es factible colocar un cable físico para configurar una red física privada, donde las oficinas están separadas por cientos o miles de millas, los costos y las dificultades involucradas hacen que una red física sea imposible. En tal caso, es mejor usar los recursos en Internet para crear una red privada. Por lo tanto, las empresas establecen una VPN (red privada

virtual), que replica las características de una red física privada. Mediante el uso de una VPN, es posible que los empleados de la empresa accedan a la red de área local o amplia incluso cuando trabajan de forma remota. La VPN encripta el tráfico desde la computadora remota a la red de la compañía, manteniendo la información lejos de posibles hackers.

_ Una red privada virtual es una implementación o sistema que habilita una comunicación segura a través de un medio inseguro, siendo transparente para el usuario o aplicación que realiza y recibe la comunicación. Cuando usa Internet, hay un proceso constante en el que su dispositivo intercambia datos con otras partes en la web. Una VPN crea un túnel seguro entre su dispositivo (por ejemplo, teléfono inteligente o computadora portátil) e Internet. Permite enviar sus datos a través de una conexión segura y encriptada a un servidor externo. Desde allí, sus datos se enviarán a su destino en internet.

Funcionamiento

Sin una VPN: cuando se accede a una página web sin una VPN, nos conectamos a esa página a través del proveedor de servicios de internet o ISP. El ISP asigna una dirección IP única que puede ser utilizada para identificarle en la página web. Debido a que nuestro ISP maneja y dirige todo el tráfico, este puede ver las páginas web que visita. Y nuestra actividad puede ser vinculada a usted mediante esa dirección IP única.

Con una VPN: el software VPN en nuestra computadora encripta el tráfico de datos y lo envía (a través de su proveedor de servicios de internet) al servidor VPN a través de una conexión segura. Luego el servidor VPN descifra los datos cifrados de nuestra PC, posteriormente enviará nuestros datos a internet y recibirá una respuesta, que es para el usuario. A continuación, el servidor VPN vuelve a cifrar el tráfico y nos lo envía de vuelta. El software VPN de nuestro dispositivo descifrará los datos para que pueda comprenderlos y utilizarlos. Entonces nuestro tráfico todavía pasa por nuestro ISP, pero el ISP ya no puede leerlo ni ver su destino final. Las páginas web que visitamos ya no pueden ver su dirección IP original, solo la dirección IP del servidor VPN, la cual es compartida por muchos otros usuarios y cambia regularmente.

Conceptos

Proxy: suele confundirse VPN con proxy y hasta algunas proxies se autodenominan VPNs gratuitas para captar más usuarios. La diferencia es sustancial: Si bien el servidor VPN actúa en cierto sentido como proxy, o suplente, para su actividad web: en lugar de su dirección IP y ubicación reales, las páginas web que usted visite solo verán la dirección IP y la ubicación del servidor VPN. Sin embargo, un proxy no le brinda ningún tipo de protección adicional, como la encriptación. De forma más específica el VPN tiene como fin velar por la seguridad de sus usuarios, el proxy es simplemente un intermediario.

Autenticación: una vez autenticados, el cliente VPN y el servidor VPN pueden estar seguros de que solo están hablando entre sí y nadie más.

Tunelización: las VPN también protegen la conexión entre el cliente y el servidor utilizando tunelización y encriptación. En algunas ocasiones se denomina a las redes privadas virtuales como túneles, ya que estas transportan la información por un canal público, pero aislando la información del resto y consecuentemente creando unas paredes virtuales que separan nuestra información de la del resto. La tunelización es un proceso mediante el cual cada paquete de datos es encapsulado dentro de otro paquete de datos. Esto dificulta a que terceros puedan leerlos durante su tránsito. Estos paquetes contienen formatos específicos para coincidir con el tipo de protocolo en uso. Es decir, un paquete que sale de una «red A» se encapsula en un formato que se fija al protocolo de transmisión, atraviesa el túnel entre redes y al final, cuando llega a su destino «red B» se desencapsula.

Encriptación: mediante la encriptación, el sistema será más seguro, cuanto mayor seguridad nos suministre el sistema criptográfico. Los datos dentro del túnel también son encriptados de tal manera que sólo el destinatario 3previsto puede desencriptarlos. Esto mantiene completamente oculto al contenido de su tráfico en internet, incluso de su proveedor de servicios de internet.

Kill switch: es una parada de emergencia. Esta función con el software de la mayoría de las VPN bloquea automáticamente todo el tráfico de Internet si la conexión de su VPN se cae. De esta manera, sus datos permanecerán seguros, protegidos y anónimos. Si su VPN fallara temporalmente, quedaría desprotegido. Sus datos se enviarán a Internet sin la protección adicional de un túnel VPN. Como resultado, su dirección IP será visible para el mundo exterior de todos modos. Aquí es donde entra el interruptor de apagado. Un kill switch de VPN le asegura que nunca se conectará a Internet sin protección. Si su VPN no funciona bien mientras navega, el interruptor de apagado apagará su conexión a Internet por completo.

Ventajas y desventajas

_ Si bien es un recurso maravilloso, Internet también está plagado de malware, cookies, piratas informáticos, censura en todo el estado y delitos cibernéticos de alto nivel. Por lo tanto, un poco más de seguridad en línea es una adición bienvenida. Usar una VPN tiene varias ventajas:

- Oculta la dirección IP real: al conectarse a uno de los servidores VPN, nuestra dirección IP se oculta y se reemplaza por la del servicio VPN, es decir, la VPN falsifica nuestra ubicación y, por lo tanto, lo anonimiza parcialmente en la web, entonces se puede navegar por internet como si estuviéramos en Reino Unido, Alemania, Canadá, Japón o prácticamente en cualquier país, si el servicio de VPN cuenta con servidores allí.

- Cifra el tráfico de datos: una VPN cifra su tráfico de datos, y esto evita que los piratas informáticos y otras partes malintencionadas se apoderen de nuestros datos importantes (o al menos, puedan descifrarlos). Permite la conexión segura a wifi públicos, ya que otro modo puede ser inseguro.
- Mayor seguridad: utilizar una VPN lo protege de las violaciones de seguridad en muchas formas, como los análisis de paquetes, las redes Wifi clandestinas y los ataques de intermediarios.
- Evita las restricciones geográficas: con una VPN es posible conectarse a un servidor en un país diferente y, por lo tanto, permitir que todo su tráfico de datos pase por este otro país. Esto a menudo permite que ciertos sitios web bloqueados, servicios de transmisión y redes sociales sean accesibles. Si estamos en alguna parte del mundo que restringe el acceso a Google, Wikipedia, YouTube u otras páginas y servicios, utilizar una VPN permitirá recuperar el acceso a la internet libre. También se puede utilizar una VPN para atravesar los firewalls de las redes escolares, oficina, etc.
- Descargas en forma segura y anónima: debido a que la dirección IP está oculta y la conexión está encriptada, los terceros ya no podrán averiguar qué se está descargando exactamente o quién lo está descargando. Debido al túnel VPN cifrado, no pueden registrar lo que se descarga a través de la conexión VPN segura. Nadie puede interceptar y ver esta información. Las buenas VPN también le ocultan la actividad a su proveedor de internet, al operador de telefonía móvil y a cualquier otra persona que pueda estar escuchando, gracias a una fuerte capa de encriptación.
- Evitar la censura del gobierno: en países donde el gobierno controla y regula Internet, no todos los sitios web están disponibles, porque dicho país los bloquea. Este tipo de censura se puede eludir, como todas las demás restricciones geográficas, utilizando una VPN.
- Comprar en línea por menos: los precios de las compras en línea a veces pueden diferir según el país desde el que se realiza la compra. Al visitar una tienda web en línea desde una conexión a Internet dentro de un país, pude tener precios caros, pero una conexión VPN, al permite al usuario conectarse a servidores VPN de todo el mundo, esto hace que los sitios web registren a los usuarios como visitantes del país donde se encuentra el servidor VPN, lo que permite al usuario beneficiarse de las mejores tarifas internacionales.
- Mejoran los juegos en línea: una VPN permite jugar juegos en línea que pueden estar restringidos en el país donde nos encontremos, o quizás el juego que queremos jugar está programado para una fecha de lanzamiento posterior, entonces con una VPN, no hace falta esperar más.
- Evitarán que su ISP limite nuestra conexión a Internet: la limitación del ancho de banda es la limitación deliberada de nuestro ancho de banda por parte del proveedor de servicios de Internet. Ocasionalmente, a los ISP no les gustará el

hecho de que estemos descargando grandes cantidades de datos. Sin embargo, cuando se encriptan nuestros datos con una VPN, los proveedores de servicios de Internet no podrán ver lo que está haciendo en línea, y tampoco podrán aislarnos.

_ Una VPN puede parecer la solución perfecta para muchos problemas de privacidad en línea. En resumen, brinda seguridad, anonimato y libertad. Sin embargo, todo tiene sus inconvenientes y diferencias entre lo gratuito y pago.

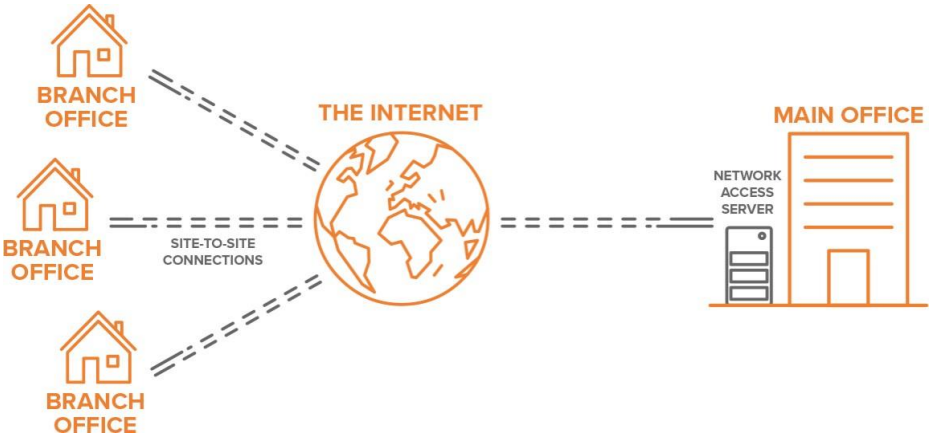
- Puede disminuir la velocidad: debido a que la conexión a Internet con una VPN se redirige y encripta a través del servidor VPN, es posible que la conexión a Internet se ralentice ligeramente.
- Correr el riesgo de ser bloqueado por ciertos servicios: las VPN están bloqueadas por servicios de transmisión como Netflix y Hulu, ya que estas empresas tienen contratos con distribuidores de películas que solo les permiten mostrar contenido en países específicos. Por lo que estas empresas bloquean las direcciones IP con las que acceden al servicio con grandes cantidades de personas al mismo tiempo.
- No son legales en todos los países: aunque pueda considerarse sospechoso, el uso de una VPN es legal en la mayoría de los países, la mayoría de las grandes empresas y corporaciones utilizan una VPN como parte de su seguridad. Sin embargo, existen algunas excepciones, ya que algunos países quieren tener un control total sobre las cosas que sus ciudadanos ven en Internet, por lo que solo se pueden usar VPNs aprobadas por el gobierno.
- Es difícil para los consumidores comprobar la calidad del cifrado: respecto a averiguar qué registros dice un proveedor que mantienen y leer más sobre la calidad y seguridad generales de la VPN. Esto incluye una breve explicación de qué protocolos y tipos de cifrado emplea el proveedor de VPN.
- La conexión se rompe: cuando la conexión a su servidor VPN se desconecta, de repente quedamos sin protección y volvemos a usar la dirección IP real.
- Sensación injustificada de impunidad en línea: hay algunas personas que creen que su conexión VPN las hace completamente anónimas y no se ven afectadas por el malware. Incluso con una conexión VPN estable y fuertemente encriptada, aún podemos:
  - Ser seguidos en la web por anunciantes, rastreadores, piratas informáticos, agencias de inteligencia, etc.
  - Ser objetivo y víctima de ataques de phishing.
  - Infectarnos con algún tipo de malware.
  - Quedar bloqueado de ciertas redes, bases de datos, páginas web, etc.
- VPN gratuitas, a veces peor que ninguna: algunas personas optan por probar un servicio VPN gratuito, pero muchos proveedores de VPN gratuitos no fueron diseñados para brindar al usuario más privacidad y anonimato en Internet, sino únicamente para ganar dinero. Además, muchas aplicaciones VPN gratuitas no son seguras y contienen software espía o malware en la descarga.

Tipos de VPN

_ Hay dos tipos fundamentales de VPN:

VPN de acceso remoto: (Remote Access VPN) permite a los usuarios conectarse a una red privada para acceder a servicios y recursos de forma remota. Esta conexión es segura y se realiza a través de internet mediante un remote access server.

VPN site-to-site: se utiliza principalmente en las empresas, en donde las organizaciones con instalaciones en diferentes ubicaciones geográficas utilizan esta VPN para conectar la red de una instalación a la red en otra, y en otra ubicación geográfica. Nos permite conectar 2 o más hogares entre sí, y tener acceso a todos los recursos compartidos, como si estuviéramos físicamente en todas las casas. El servidor VPN es el que posee un vínculo permanente a internet, acepta todas las conexiones que provienen de los sitios, y establece el túnel VPN. La VPN de sitio a sitio crea un puente virtual entre las redes, en oficinas geográficamente distantes, y las conecta a través de internet para mantener una comunicación segura y privada entre las redes.



Protocolos VPN

_ Los protocolos VPN son los métodos mediante los cuales se conecta su dispositivo al servidor VPN. Algunos protocolos favorecen la velocidad, otros favorecen la seguridad y algunos simplemente funcionan mejor en determinadas condiciones de red. Los protocolos determinan qué algoritmo de encriptación utilizar, cómo establecer y verificar

las claves de encriptación y cómo manejar los potenciales errores. Algunos de los más utilizados son:

Point to Point Tunneling Protocol (PPTP): este opera en el puerto TCP 1723, y es uno de los protocolos VPN más antiguos en uso, siendo contemporáneo con Windows 95, y estándar en todas las versiones de Windows desde entonces. Es uno de los más comunes, más fáciles de configurar, y computacionalmente rápidos. Por esa razón, es particularmente útil para aplicaciones en las cuales la velocidad es fundamental, como streaming de audio o video, o en dispositivos más antiguos y lentos, con procesadores más limitados. Sin embargo, el PPTP también está expuesto a serias vulnerabilidades de seguridad.

Internet Protocol Security (IPSec): este protocolo proporciona servicios de seguridad a la capa IP y a todos los protocolos superiores, como TCP y UDP (capa de transporte en internet). Proporciona todos los servicios necesarios para que la comunicación sea segura (autenticación, confidencialidad, integridad y no repudio). Por supuesto, también tenemos control de acceso, calidad de servicio y registro de actividad. Pero hay que descargar que es más lento que Open VPN, a veces es bloqueado por los firewalls, y a pesar de todo solo es moderadamente seguro.

Open VPN: es un protocolo altamente configurable y relativamente nuevo. Lo mejor de este es que es de código abierto. Aunque la palabra “abierto” tal vez no suene muy atractiva para una herramienta de privacidad, verdaderamente representa una enorme ventaja. Si hubiese fallos de seguridad en el código, la comunidad del código abierto las identificaría rápidamente. En combinación con un robusto algoritmo de encriptación, Open VPN es uno de los protocolos de VPN más seguros que existen. Es compatible con sistemas operativos Microsoft Windows, GNU/Linux, macOS e incluso tiene aplicaciones gratuitas para Android y iOS. Otro punto fuerte de Open VPN es que algunos fabricantes de routers lo están incorporando en sus equipos, por lo que se puede tener la posibilidad de configurar un servidor Open VPN en el router. Otro aspecto destacable es que, por ejemplo, sistemas operativos orientados a firewalls también lo incorporan. Open VPN utiliza la biblioteca OpenSSL para proporcionar cifrado.

IKEv2: desarrollado por Microsoft y Cisco, es la siguiente versión del protocolo Internet Key Exchange. Es un protocolo de tunelización basado en IPSec que proporciona un canal de comunicación VPN seguro y define los medios automáticos de negociación y autenticación para las asociaciones de seguridad IPSec de forma segura. Se le considera más liviano y estable que Open VPN, al tiempo que conserva cierto nivel de personalización. Sin embargo, solo está disponible a través de UDP, que a su vez es bloqueado por algunos firewalls. IKEv2 es uno de los protocolos más nuevos y tiene fortalezas significativas, particularmente su velocidad, además implementa una gran cantidad de algoritmos criptográficos,

Secure Sockets Layer (SSL/TLS): es la tecnología estándar para mantener segura una conexión a Internet, así como para proteger cualquier información confidencial que se envía entre dos sistemas e impedir que los delincuentes lean y modifiquen cualquier dato que se transfiera, incluida información que pudiera considerarse personal. Los dos sistemas pueden ser un servidor y un cliente (por ejemplo, un sitio web de compras y un navegador) o de servidor a servidor (por ejemplo, una aplicación con información que puede identificarse como personal o con datos de nóminas). TLS es una versión mejorada de SSL, que funciona de un modo muy parecido a SSL, utilizando cifrado que protege la transferencia de datos e información.

Secure Shell (SSH): es un protocolo de administración remota que le permite a los usuarios controlar y modificar sus servidores remotos a través de Internet a través de un mecanismo de autenticación. Es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. La principal diferencia entre VPN y SSH es que SSH se conecta a una computadora en particular mientras que una VPN se conecta a una red. Cada uno de ellos proporciona una capa adicional de seguridad al navegar en línea. SSH cifra las aplicaciones en lugar de todo el tráfico procedente de su dispositivo.

Utilización de criptografía en VPN

_ Hackear una conexión VPN implica vulnerar la criptografía de esta, ya sea descodificando el cifrado mediante algún tipo de vulnerabilidad, o bien robando la clave mediante algún método fraudulento; de allí la importancia de la criptografía utilizada. Los hackers y criptoanalistas utilizan ataques criptográficos para recuperar la información en forma de texto de la versión cifrada de ésta, sin la clave. No obstante, descodificar el cifrado es una tarea que demanda mucho en cuanto a computación y tiempo, y puede llevar años. En general, estas son las principales cosas hacer para obtener la experiencia en línea más segura:

- Una clave de cifrado larga, de al menos 128 bits de tamaño.
- Protocolos confiables de intercambio de claves, como ECDH o RSA-2048.
- Cifrados VPN fuertes como AES, Twofish o Camellia.
- Potentes protocolos de VPN como Open VPN, SoftEther e IKEv2.
- Un hash SHA-2 para la autenticación HMAC: idealmente 256 bits, 384 bits o 512 bits. La autenticación HMAC significa Código de Autenticación de Mensajes Basado en Hash, y es un Código de Autenticación de Mensajes (MAC) que se utiliza para verificar la integridad de los datos y la autenticación de un mensaje al mismo tiempo para asegurarse de que no haya sido modificado por terceros.
- Perfect Forward Secrecy

Intrusion Detection Systems

Conceptos básicos

Intrusión: conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso. Podemos decir que se trata de una violación de las políticas de seguridad del sistema. Tipos de intrusos:

- Usuario suplantador: es un individuo externo que no está autorizado a usar el sistema y penetra los controles de acceso para obtener provecho de la cuenta de un usuario legítimo.
- Usuario fraudulento: es un usuario interno legítimo que accede a recursos para los que el acceso no está autorizado o hace mal uso de sus privilegios.
- Usuario clandestino: es un individuo interno o externo que toma el control de supervisión del sistema y lo usa para evadir los controles de auditoría y de acceso o para suprimir información de auditoría.

Detección de intrusos: análisis automático de parámetros que modelan la actividad de un entorno con el propósito de detectar e identificar intrusiones.

Falso positivo: consiste en la detección de datos o paquetes como una amenaza o intrusión cuando en realidad no se trata de un intento de ataque sobre la red.

Falso negativo: consiste en los paquetes o datos que son amenazas para una red pero el sistema de seguridad no detecta dichas amenazas.

Firmas / Bases de datos de firmas: las bases de datos de antivirus se llaman históricamente firmas, tanto en el uso común como en el escrito. Las firmas de virus son una secuencia continua de bytes comunes en cierta muestra de malware, lo que significa que se contiene dentro de este o del archivo infectado y no en archivos no afectados.

Log: es un registro que deja un sistema informático. Por ejemplo: accesos de usuarios, actividades de borrado y cambios realizados en el sistema. Con esta información podemos ver claramente las actividades que se han realizado en nuestros sistemas y, por lo tanto, con un pequeño análisis podríamos detectar situaciones extrañas y anómalas.

Definición

_ El sistema de detección de intrusiones es una aplicación usada para detectar accesos no autorizados a un ordenador o a una red, es decir, son sistemas que monitorizan el tráfico entrante y lo cotejan con una base de datos actualizada de firmas de ataque conocidas. Ante cualquier actividad sospechosa, emiten una alerta a los administradores del sistema quienes han de tomar las medidas oportunas. Los IDS no sólo analizan el tráfico de la red, sino que también analizan su comportamiento y su contenido. Estos accesos pueden ser

ataques esporádicos realizados por usuarios malintencionados o repetidos cada cierto tiempo, lanzados con herramientas automáticas. Estos sistemas sólo detectan los accesos sospechosos emitiendo alertas anticipatorias de posibles intrusiones, pero no tratan de mitigar la intrusión. Su actuación es reactiva.

Ventajas y desventajas

_ Como ventajas tenemos las siguientes:

- Ver lo que está sucediendo en la red en tiempo real en base a la información recopilada.
- Reconocer modificación en los documentos
- Automatizar los patrones de búsqueda en los paquetes de datos enviados a través de la red.

_ Como desventajas tenemos las siguientes:

- Estas herramientas no están diseñadas para prevenir o detener los ataques que detectan.
- Son vulnerables a ataques DDos que pueden provocar inoperatividad.
- Pueden ocurrir falsos positivos (cuando detecta datos o paquetes como una amenaza o intrusión y se lanza una alarma pero no se trata de algún intento de ataque) o falsos negativos (paquetes o datos que son amenazas para una red pero el sistema de seguridad no detecta dichas amenazas).

Funcionamiento

_ El funcionamiento de estas herramientas se caracterizan por un sistema de gestión central que recibe las informaciones necesarias tanto desde el software basado en la red como desde el software basado en el host. Hay tres componentes básicos involucrados en el proceso de reconocimiento:

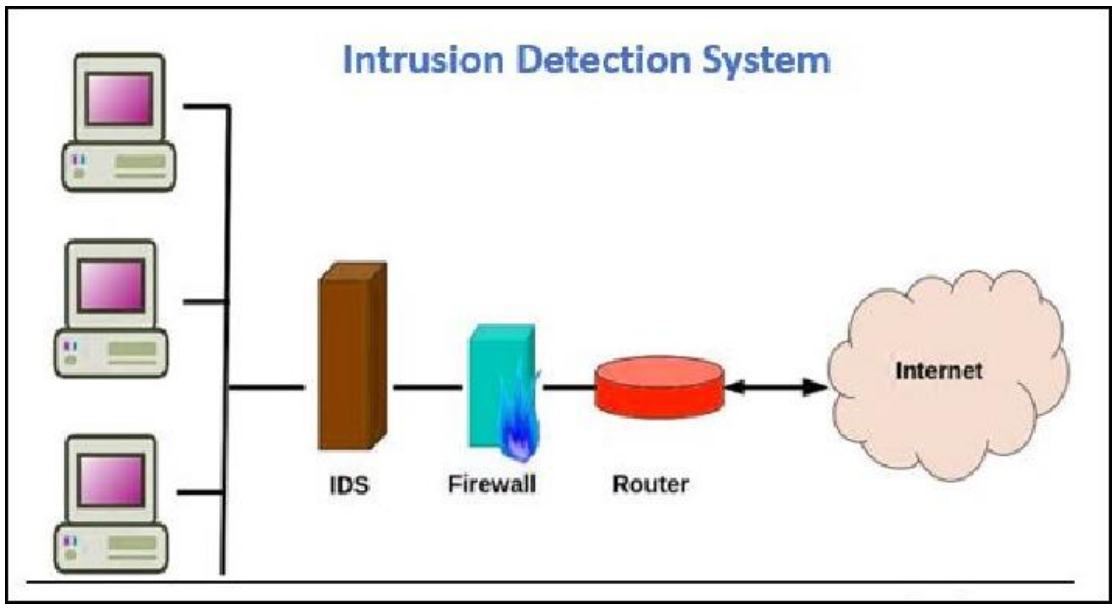
Monitoreo de datos: tiene la tarea de recoger y hacer un primer filtro a los datos necesarios para filtrar intrusos. Se trata de la auditoría de datos, que incluye archivos log de sistemas informáticos y aplicaciones de seguridad como, por ejemplo, la capacidad de la CPU, el número de conexiones de red activas o la cantidad de intentos de inicio de sesión.

Análisis: el monitor de datos envía el flujo de datos recogidos y previamente filtrados al llamado analyzer (analizador). Este debe editar y evaluar la información obtenida en tiempo real, de lo contrario no sería posible evitar los ataques a tiempo.

Informe de resultados: en la etapa final, el IDS informa al administrador de la red si encontró un ataque o un comportamiento sospechoso del sistema. Dependiendo del potencial de riesgo, existen diferentes posibilidades de notificarlo. Así, por ejemplo, un sistema que necesita defenderse enviaría:

- Un correo electrónico que explique la naturaleza del ataque.
- Una alarma local como una ventana emergente que active la consola de seguridad.
- Un mensaje de alerta a un dispositivo móvil.

_ Normalmente esta herramienta se integra con un firewall. El IDS es incapaz de detener los ataques por sí solo, excepto los que trabajan conjuntamente en un dispositivo de puerta de enlace con funcionalidad de firewall, convirtiéndose en una herramienta muy poderosa ya que se une la inteligencia del IDS y el poder de bloqueo del firewall, al ser el punto donde forzosamente deben pasar los paquetes y pueden ser bloqueados antes de penetrar en la red. Los IDS suelen disponer de una base de datos de "firmas" de ataques conocidos. Dichas firmas permiten al IDS distinguir entre el uso normal del PC y el uso fraudulento, y/o entre el tráfico normal de la red y el tráfico que puede ser resultado de un ataque o intento del mismo.



_ Algunas de las técnicas específicas que usan los IDS en sistemas informáticos son:

- Almacenamiento de paquetes bajo sospecha de ataque: cuando se sospecha de alguna intrusión, el IDS va a tomar la medida de guardar un registro detallado con todos los paquetes de información que ocasionaron una señal de alerta y que fueron capturados por el protocolo de detección.
- Verificación de la configuración de dispositivos externos: en donde al momento de detectar una intrusión o sospechar de su existencia, el IDS procede a solicitar una reconfiguración de los dispositivos externos que tiene la misión de bloquearla,

como por ejemplo, el firewall. Esto se realiza mediante el envío de una señal de alerta.

- Envío de una señal de alerta: los IDS cuentan con la función de notificar visualmente al usuario, así como también a los administradores del sistema sobre la presencia de una posible intrusión.
- Alerta mediante correo electrónico: algunos IDS cuentan con la capacidad de remitir un correo electrónico de alerta a uno o más usuarios en donde se informa la posible intrusión.
- Registro de la intrusión en una base de datos: toda detección de una posible intrusión debe ser registrada, para así llevar un control detallado del incidente.

Características de un IDS

_ Un IDS debe poseer las siguientes características:

Ligero: debe imponer mínima sobrecarga sobre el sistema.

Adaptable: debe ser fácilmente adaptable al sistema operativo ya instalado, pues cada sistema operativo tiene un patrón de funcionamiento diferente y el mecanismo de defensa debe adaptarse de manera sencilla a esos patrones. También debe hacer frente a los cambios de comportamiento del sistema según se añaden nuevas aplicaciones al mismo.

Confiable: debe funcionar continuamente sin supervisión humana y por lo tanto lo suficientemente fiable para poder ser ejecutado en segundo plano como parte del dispositivo o red que está siendo observada. También deben ser difíciles de vulnerar y suministrar “tranquilidad” a los especialistas de seguridad.

Robusto: debe ser tolerante a fallos en el sentido de que debe ser capaz de sobrevivir una caída del sistema y además resistir perturbaciones, en donde deberá monitorizarse a sí mismo para asegurarse que no ha sido perturbado.

Identificación: distinguir lo que es un ataque de lo que es compartir un recurso del sistema.

Colaborativo: debe ayudar a identificar de dónde provienen los ataques que se sufren, y recoger evidencias que pueden ser usadas para identificar intrusos.

Clasificación de IDS

_ Según tipo de respuesta:

Pasivos: realiza el sencillo trabajo de detección y alerta. Simplemente alerta al administrador de cualquier tipo de amenaza y bloquea la actividad en cuestión como medida preventiva. Solo notifican mediante algún mecanismo (alerta, log, etc) pero no actúa sobre el ataque o el atacante.

Activos: detecta actividad malintencionada, alerta al administrador de las amenazas y también responde a esas amenazas. Genera algún tipo de respuesta sobre el sistema atacante o fuente de ataque como cerrar la conexión, reprogramar el firewall o enviar algún tipo de respuesta predefinida.

_ Según estrategia de análisis:

Uso indebido: cuentan con el conocimiento a priori de las secuencias y actividades que conforman el ataque. Por lo tanto, para detectar intrusiones dentro de la información recopilada de la fuente, se realiza una comparación de la misma con los patrones de ataques previamente almacenados y, en caso de encontrar similitud, se genera una alarma. Con este método se logra detectar intentos de explotación de vulnerabilidades típicos, de los cuales ya existe información.

Detección de anomalías: tienen un conocimiento complementario a los de uso indebido, es decir, que parten del conocimiento de lo normal y toda actividad que se aleje de este comportamiento es considerada una intrusión. Este tipo de detección evita el proceso de actualización de una base de datos de patrones de intrusión y brinda la posibilidad de detectar ataques nuevos de los cuales no se tenga información alguna. Sin embargo, este tipo tiene algunas desventajas como generar falsos positivos ya que el comportamiento normal de los usuarios es extremadamente difícil de modelar por lo variable que puede llegar a ser y por lo tanto un comportamiento inusual no tiene necesariamente que ser ilícito. Otra debilidad es que necesitan un largo periodo de “entrenamiento”, previo a su uso, para poder identificar los comportamientos normales de los usuarios y sistemas dentro de la red.

_ Según el origen de datos:

HIDS (IDS basados en host): son diseñados para monitorear, detectar y responder a los datos generados por un usuario o un sistema en un host. Estos sistemas ayudan a las organizaciones a monitorear los procesos y aplicaciones que se ejecutan en dispositivos como servidores y estaciones de trabajo. HIDS rastrea los cambios realizados en la configuración del registro y la configuración crítica del sistema, archivos de registro y contenido, alertando sobre cualquier actividad no autorizada o anómala. Las tecnologías HIDS son de naturaleza “pasiva”, lo que significa que su propósito es identificar la actividad sospechosa, no prevenirla.

NIDS (IDS basados en red): son diseñados para analizar el tráfico de la red completa examinando los paquetes individualmente, detectando paquetes armados maliciosamente y diseñados para no ser detectados por los firewalls, encontrar cual es el programa al que se está accediendo y producir alertas cuando el atacante intenta explotar algún fallo de este programa. Es un dispositivo de red configurado en modo promiscuo. Analizan el tráfico de red, en tiempo real y no solo trabajan a nivel TCP/IP sino también a nivel de aplicación.

Acciones a tomar ante una intrusión

_ Una vez que nuestro IDS detecta una intrusión, hay dos tipos de respuestas:

Respuestas activas: estas respuestas se pueden dividir en dos clases:

- Aquellos que ejercen control sobre el sistema atacado y modifican el sistema para mitigar los efectos del ataque.
- Aquellos que ejercen control sobre el sistema atacante y se convierten en atacantes intentando remover la plataforma de operación del atacante. Esto no es legal.

Respuestas pasivas: estas responden con una notificación a la autoridad necesaria, y no intentan mitigar el daño hecho o buscar dañar al atacante. Para poder responder con eficacia el punto inicial es la planificación y organización, por lo que debemos:

- Planear: crear un plan simple, claro y preciso que determine de forma clara quién hace qué, cómo y cuándo para actuar de forma rápida.
- Crear un equipo de respuesta a incidentes: se debe crear un equipo con roles detallados y responsabilidades claras.
- Clasificar incidentes: es muy importante clasificar incidentes, establecer criticidades, vectores de ataque e impactos. Esto permite tener un histórico de incidentes que nos permite aprender para futuras ocasiones.
- Priorizar el negocio: se debe entender las prioridades del negocio y alinear el plan a sus necesidades.

Ejemplos de IDS

- Snort
- Suricata
- OSSEC
- Security Onion

Seguridad, usuarios y redes sociales

Ingeniería Social

_ Se llama ingeniería social a las diferentes técnicas de manipulación que usan los ciberdelincuentes para obtener información confidencial de los usuarios. Los ciberdelincuentes engañan a sus víctimas haciéndose pasar por otra persona, en donde el objetivo de este engaño es apropiarse de datos personales, contraseñas o suplantar la identidad de la persona engañada.

Tipos de ingeniería social

_ Los ingenieros sociales tienen más de unos pocos trucos bajo la manga para engañar a objetivos desprevenidos:

Phishing: los piratas informáticos aficionados envían correspondencia masiva, lanzando a una amplia red y con la esperanza de engañar a un gran número de destinatarios. Pero la mayoría de las veces, estos mensajes genéricos suelen ser demasiado impersonales para engañar a nadie. La mayoría de los ciberdelincuentes experimentados provocan un phishing a la vez. Tenemos algunos tipos:

- Phishing Email: estos son los correos electrónicos que tienen intenciones maliciosas. Ya sea un mensaje aparentemente normal con un archivo adjunto infectado o uno que engaña a los lectores para que hagan clic en una URL falsificada que captura sus credenciales de inicio de sesión.
- Vishing (Phishing de voz): es cualquier forma de phishing que se realiza por teléfono. Estos son mensajes de correo de voz que le piden que vuelva a llamar para tomar medidas inmediatas y, a menudo, aprovechan el miedo para recibir devoluciones de llamada.
- Smishing (Phishing por SMS): este puede ser un mensaje de texto que le indica que se atrasa en un pago y que debe pagar en el enlace adjunto para evitar un cargo por atraso, en el que el pirata informático captura su información de inicio de sesión o datos bancarios. O es un número falso que se hace pasar por alguien del gobierno que le envía recursos COVID-19 con un enlace cargado de malware infectado.

Pretexting: si bien el phishing es el proceso de intentar adquirir información confidencial, el pretexting es la historia falsa que el mal actor teje durante el phishing. Es la narrativa que inventan basándose en su conocimiento investigado de ti para engañarte y hacerte creer en su legitimidad. Los pretextos comunes implican hacerse pasar por alguien que conoces u otra fuente confiable, con una razón claramente explicada por la que te piden información o que tomes una acción.

- Caso Famoso: en la década de 1960, Frank Abagnale.

Baiting: se produce cuando un ciberdelincuente pone algo tentador frente a nosotros, con la esperanza de que actúes. Esto podría ser un anuncio de que ganaste un Iphone por ser el visitador número 1.000.000 o un documento etiquetado como "Confidencial". A veces, el delincuente ni siquiera le pedirá que haga clic en él, con la esperanza de que su propia curiosidad se haga cargo.

Tailgating: estos atacantes suelen tener un pretexto inteligente, “vestido como un repartidor que lleva cajas o como una cara amistosa con una docena de donas para el

personal", creando un falso sentido de confianza para dejarlos pasar por la puerta de la empresa u organización, detrás de su objetivo.

Quid Pro Quo: en latín significa "algo por algo" y es una técnica de ingeniería social en la que el ciberdelincuente ofrece un beneficio al objetivo a cambio de información o acceso. Podría ser alguien que se hace pasar por un miembro de un equipo de IT y dice que necesita la contraseña de una computadora para realizar una actualización necesaria del sistema o la promesa de una descarga de música gratuita si se suscribe a un servicio de transmisión falso.

Scareware Ocurre: cuando las personas son bombardeadas con amenazas y alarmas falsas. Una de las más comunes son los avisos en internet, en los que "informan" que el equipo está infectado con un virus y se debe instalar algún programa para proteger la información personal de posibles espías cibernéticos. Al momento de la instalación, el estafador podrá tener acceso a la computadora de la víctima, que ya ha caído en la trampa.

Educación en seguridad para el usuario final

Prácticas preventivas

- Mantener el software actualizado: activando las actualizaciones automáticas del sistema para su dispositivo, asegurándonos de que el navegador web de su escritorio utilice actualizaciones de seguridad automáticas, y mantener actualizados los complementos de su navegador web como Flash, Java, etc.
- Usar protección antivirus y cortafuegos: el software de protección antivirus (AV) ha sido la solución más común para combatir ataques maliciosos ya que evita que el malware y otros virus maliciosos ingresen a su dispositivo y pongan en peligro sus datos. Y el uso de un firewall también es importante a la hora de defender sus datos contra ataques malintencionados ya que ayuda a detectar piratas informáticos, virus y otras actividades maliciosas que ocurren en Internet y determina qué tráfico puede ingresar a su dispositivo.
- Usar contraseñas seguras y usar una herramienta de gestión de contraseñas: se debe considerar evitar la compleja mezcla de letras mayúsculas, símbolos y números, y en su lugar, optar por algo más fácil de usar pero con al menos ocho caracteres y una longitud máxima de 64 caracteres, también no usar la misma contraseña dos veces, la contraseña debe contener al menos una letra minúscula, una letra mayúscula, un número y cuatro símbolos, etc.
- Utilizar autenticación de dos factores o de varios factores: la autenticación de dos factores o de múltiples factores es un servicio que agrega capas adicionales de seguridad al método estándar de contraseña de identificación en línea. Sin la autenticación de dos factores, normalmente ingresaría un nombre de usuario y una contraseña. Pero, con dos factores, se le pedirá que ingrese un método de

autenticación adicional, como un código de identificación personal, otra contraseña o incluso una huella digital. Con la autenticación de múltiples factores, se le pedirá que ingrese más de dos métodos de autenticación adicionales después de ingresar su nombre de usuario y contraseña.

- Obtener más información sobre las estafas de suplantación de identidad, desconfíe mucho de los correos electrónicos, las llamadas telefónicas y los folletos.
- Proteger nuestra información confidencial de identificación personal (PII): la información de identificación personal (PII) es cualquier información que pueda ser utilizada por un ciberdelincuente para identificar o localizar a una persona. La PII incluye información como nombre, dirección, números de teléfono, datos de nacimiento, número de seguro social, dirección IP, detalles de ubicación o cualquier otro dato de identidad físico o digital.
- Utilice sus dispositivos móviles de forma segura: creando un código de acceso móvil difícil, instalar aplicaciones de fuentes confiables, mantener el dispositivo actualizado, realizar copias de seguridad, etc.
- Haga una copia de seguridad de sus datos con regularidad.
- No usar wifi público: sin usar una red privada virtual (VPN).
- Revisar cuentas e informes crediticios en línea con regularidad para ver si hay cambios.

Prácticas de seguridad recomendadas

_ A diferencia de otros ataques cibernéticos, la ingeniería social no está orientada a hacerle daño a computadoras y celulares, sino a la manipulación psicológica y emocional de las personas para obtener algo a cambio. A continuación, se detallan algunas recomendaciones para no ser víctima de estas estafas:

- Nunca entregar credenciales de acceso a plataformas.
- No compartir información sensible en tus redes sociales.
- Evitar abrir correos y archivos adjuntos de fuentes sospechosas. Si no se conoce al remitente, no responder el correo hasta verificar su autenticidad.
- Si se recibe correos con ofertas, regalos o beneficios tentadores, pensar dos veces antes de hacer clic y aceptarlos. Validarlos antes de ultima.
- Contactarse directamente con el remitente que está pidiendo tal información sensible, para verificar su identidad.
- Actualizar el software y antivirus de la computadora constantemente, para evitar archivos maliciosos.
- Eliminar el historial y caché de la computadora para que no recuerde las credenciales de acceso a plataformas.
- Seguir las políticas y consejos de seguridad de nuestra empresa.
- Monitorear constantemente nuestros perfiles sociales y cuentas bancarias para confirmar que todo está en orden.

- Evitar conectarte a redes wifi públicas para navegar por internet.

Herramientas de prevención

Endpoint: es cualquier dispositivo informático que esté conectado a una red (algunos ejemplos son equipos de escritorio, portátiles y dispositivos móviles).

Endpoint Protection Platform (EPP): es el antivirus tradicional, y es una solución de seguridad diseñada para detectar y bloquear amenazas a nivel de dispositivo. Incluye funciones de antivirus, antimalware, prevención de intrusiones (IPS), prevención de pérdida de datos (DLP), y los más avanzados de prevención de exploits, tecnología anti-ransomware, etc. Las herramientas de un antivirus tienen un enfoque preventivo y utilizan firmas para identificar amenazas. Es un mecanismo de defensa de primera línea y es efectivo para bloquear sobre todo amenazas conocidas.

- Se centra únicamente en la prevención en el perímetro. Tiene como objetivo evitar que las amenazas ingresen en la red.

Endpoint Detection and Response (EDR): es una herramienta que proporciona monitorización y análisis continuo del endpoint y la red. La finalidad es identificar, detectar y prevenir amenazas avanzadas (APT) con mayor facilidad. Proporciona herramientas adicionales para buscar amenazas desconocidas. Es posible realizar un análisis forense y responder de manera rápida y efectiva a los ataques. La tecnología EDR detecta ataques que nuestro antivirus ha pasado por alto. Monitoriza y evalúa todas las actividades de la red (eventos de los usuarios, archivos, procesos, registros, memoria y red). Y detecta ataques informáticos en tiempo real, y permite tomar medidas inmediatas si es necesario.

- Está enfocado en amenazas avanzadas, las diseñadas para evadir la primera capa de defensa y que logran penetrar en la red. Detecta esa actividad y contiene al adversario antes de que pueda moverse lateralmente en la red.

Redes sociales

_ Las redes sociales son tecnologías interactivas que permiten la creación o el intercambio de información, ideas, intereses profesionales y otras formas de expresión a través de comunidades y redes virtuales.

Delitos en redes sociales

Grooming: o engatusamiento, se refiere a aquellas prácticas online de ciertos adultos para ganarse la confianza de un/a menor con fines de satisfacción sexual. Esta práctica está íntimamente relacionada con la pornografía infantil, la pederastia y deriva en casos de abuso sexual. Los acosadores se ponen en contacto con menores, ya sea por las redes

sociales, por chat o videochat o por foros de contactos. En la gran mayoría de los casos falsifican su identidad con el fin de conseguir un encuentro con el menor.

Cyberbullying: se produce cuando un/a menor atormenta, amenaza, humilla o molesta a otro/a mediante Internet, teléfono móvil, videoconsolas u otras tecnologías. El ciberbullying puede ser el paso siguiente al acoso sobre una persona o un punto de inicio difícilmente perceptible por padres y educadores.

Sexting: es una práctica consistente en el intercambio de contenidos personales de tipo sexual (fotografías o videos) por medio de teléfonos móviles o Internet. Estos envíos suelen producirse entre parejas o bien con el fin de intentar seducir a otra persona, pero el peligro viene dado en que muchas veces se suele contactar con gente desconocida. La extorsión con hacer públicas esas fotografías propicia graves problemas psicológicos al menor (que también afecta a adultos), que en algún caso han acabado en suicidio.

_ Algunas prácticas de seguridad recomendadas en redes sociales son:

1. Borrar cualquier imagen explícita que se nos envíe.
2. No distribuir imágenes explícitas.
3. Ignorar o rechazar cualquier solicitud de otros de imágenes inapropiadas.
4. Bloquear a personas que incomodan como hablan (o con lo que envían).
5. Discutir las consecuencias de tomar, enviar o reenviar una foto sexual.
6. Nunca tomarse fotos que no quieras que vean todos (tus compañeros de clase, profesores, familiares, amigos y tu empleador).
7. Antes de presionar enviar, recordar que no se puede controlar a dónde puede viajar esta imagen.

Seguridad en Aplicaciones Web

Aplicación web

_ Las aplicaciones web son un tipo de software que se codifica en un lenguaje soportado por los navegadores web y cuya ejecución es llevada a cabo por el navegador en Internet.

Funcionamiento y arquitectura

_ La arquitectura de la aplicación web define las interacciones entre las aplicaciones, los sistemas de middleware y las bases de datos para garantizar que varias aplicaciones puedan trabajar juntas. Las aplicaciones web se basan en una arquitectura cliente/servidor por un lado está el cliente (el navegador o explorador) y por otro lado el servidor. Existen diversas variantes de la arquitectura básica según cómo se implementan las diferentes funcionalidades de la parte servidor. El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se

solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó

Modelo MVC

_ El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Se utiliza el MVC porque nos permite separar los componentes de nuestra aplicación dependiendo de la responsabilidad que tienen, Esto significa que cuando hacemos un cambio en alguna parte de nuestro código, esto no afecte otra parte del mismo.

- Modelo: se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. todo eso va aquí, en el modelo.
- Vista: son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.
- Controlador: se encarga de manejar instrucciones, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.

Componentes de la arquitectura de una aplicación web DNS

Domain Name System (DNS): es un sistema fundamental que ayuda a buscar un nombre de dominio y una dirección IP, y de esta manera, un servidor en particular recibe una solicitud enviada por un usuario.

Load Balancer: al dirigir las solicitudes entrantes a uno de los múltiples servidores, el balanceador de carga envía una respuesta a un usuario.

Web App Servers: este componente procesa la solicitud de un usuario y envía documentos (JSON, XMK, etc.) a un navegador. Para realizar esta tarea, generalmente se refiere a las infraestructuras de back-end, como la base de datos, el servidor de caché, la cola de trabajos y otras.

Databases: proporciona instrumentos para organizar, agregar, buscar, actualizar, eliminar y realizar cálculos.

Caching Service: el servicio de almacenamiento en caché proporciona almacenamiento de datos, lo que permite almacenar y buscar datos. Siempre que un usuario obtiene información del servidor, los resultados de esta operación se almacenan en caché. Entonces, las solicitudes futuras ya que permite hacer referencia al resultado anterior para hacer un cálculo mucho más rápido.

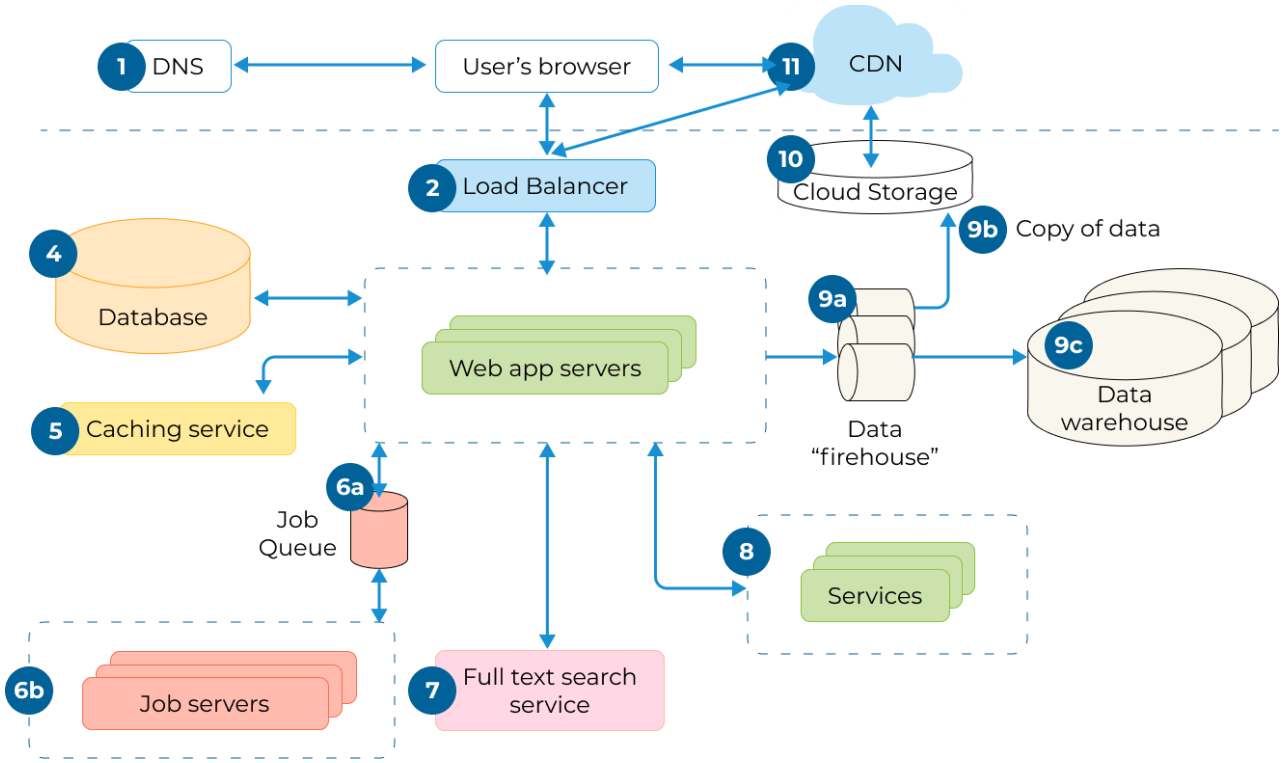
Job Queue (optional): la cola de trabajos consiste en que cuando un trabajo debe completarse, pasa a la cola de trabajos y se opera según un cronograma.

Full-Text Search Service (optional): muchas aplicaciones web admiten la función de búsqueda por texto o la llamada solicitud, y luego, una aplicación envía los resultados más relevantes a un usuario. Con la ayuda de palabras clave, busca los datos necesarios entre una gran cantidad de documentos.

Services: los servicios no son tan visibles entre otros componentes de la aplicación web, pero la aplicación web y otros servicios interactúan con ellos.

Data Warehouse: este es un almacén electrónico donde generalmente una empresa u organización mantiene una gran cantidad de información.

Content Delivery Network (CDN): se ocupa del envío de archivos HTML, archivos CSS, archivos JavaScript e imágenes.



Tipos de Ataques más frecuentes

_ Los sitios web son constantemente atacados. Los hackers buscan, ya sea comprometer la red de una corporación o a los usuarios finales, accediendo al sitio web y obligándolos a realizar drive-by downloading (descarga involuntaria de software de ordenador proveniente de Internet). A continuación vamos a ver cuáles son los ataques de aplicaciones web más comunes hoy en día:

Cross Site Scripting: tipo de vulnerabilidad informática o agujero de seguridad típico de las aplicaciones Web, que puede permitir a una tercera persona inyectar en páginas web visitadas por el usuario, código JavaScript o en otro lenguaje similar. Es posible encontrar

esta vulnerabilidad en aplicaciones, que tengan entre sus funciones, presentar la información en un navegador web u otro contenedor de páginas web. Este puede ser utilizado para robar información delicada, secuestrar sesiones de usuario, y comprometer el navegador, subyugando la integridad del sistema. Tenemos dos tipos de ataques:

- Directa: consiste en insertar código HTML peligroso en sitios que lo permitan; incluyendo etiquetas como como <script> o <iframe>.
- Indirecta: consiste en modificar valores que la aplicación web utiliza para pasar variables entre dos páginas, sin usar sesiones, y sucede cuando hay un mensaje o una ruta en la URL del navegador, en una cookie, o cualquier otra cabecera HTTP. En donde un atacante en realidad trataría de colocar un script en una URL que robe las cookies de la víctima.

_ Una opción de mitigación es HTML Entity Encode, que es útil cuando se busca información de desconfianza dentro del body de HTML, como un , pero no funciona cuando esta información está dentro de un <script> .

Inyección SQL: es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos. La vulnerabilidad radica en la incorrecta comprobación o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código SQL. Se inserta o "inyecta" código SQL invasor dentro del código SQL programado, a fin de alterar el funcionamiento normal del programa y lograr así que se ejecute la porción de código "invasor" incrustado, en la base de datos. Al ejecutarse la consulta en la base de datos, el código SQL inyectado también se ejecutará y podría hacer un sinnúmero de cosas, como insertar registros, modificar o eliminar datos, autorizar accesos e, incluso, ejecutar otro tipo de código malicioso en el computador.

_ Algunas mitigaciones son validar las entradas del usuario, desinfectar los datos limitando los caracteres especiales, utilizar procedimientos almacenados en la base de datos, cifrado (mantenga sus secretos en secreto), no mostrar más de lo necesario en mensajes de error, crear declaraciones preparadas y queries parametrizadas, monitoreo continuo de declaraciones SQL, etc.

Buffer Overflow: es un error de software que se produce cuando un programa no controla adecuadamente la cantidad de datos que se copian sobre un área de memoria reservada (buffer).

- Stack Smashing: es un tipo de buffer overflow que es aprovechado por algunos virus y otros programas maliciosos para tomar control sobre una aplicación, o provocar su terminación. Esto sucede cuando, por algún error imprevisto, se ingresa a la pila de la aplicación más datos que los que ésta puede contener, lo que provoca que esta se "desborde" y algunos datos se sobrescriben.

_ Como mitigaciones, la forma más sencilla de prevenir estas vulnerabilidades es simplemente usar un lenguaje que no permita el uso del buffer.

Distributed Denial-of-Service (DDoS): un ataque distribuido de denegación de servicio es cuando un atacante, o atacantes, intentan hacer imposible la entrega de un servicio. Esto se puede lograr frustrando el acceso a prácticamente cualquier cosa, como servidores, dispositivos, servicios, redes, aplicaciones e incluso transacciones específicas dentro de las aplicaciones. Generalmente, estos ataques funcionan ahogando un sistema con solicitudes de datos. Esto podría estar enviando a un servidor web tantas solicitudes para servir una página que se bloquea bajo la demanda. El resultado es el ancho de banda de Internet disponible, la capacidad de la CPU y la RAM se ve abrumada. Hay tres clases principales de ataques DDoS:

1. Los ataques basados en volumen utilizan cantidades masivas de tráfico falso para abrumar un recurso como un sitio web o un servidor. Incluyen ICMP, UDP y ataques de inundación de paquetes falsificados. El tamaño de un ataque basado en volumen se mide en bits por segundo (bps).
2. Los ataques a la capa de aplicación se llevan a cabo inundando aplicaciones con solicitudes creadas con fines malintencionados. El tamaño de los ataques a la capa de aplicación se mide en solicitudes por segundo (RPS).

_ Para cada tipo de ataque, el objetivo es siempre el mismo: hacer que los recursos en línea sean lentos o que no respondan por completo.

_ Como mitigaciones, para prevenir estos ataques se tiene que bloquear el tráfico “malo” antes de que llegue al sitio, también se debe monitorear el comportamiento de los visitantes, bloquear bots maliciosos conocidos y desafiar entidades sospechosas o no reconocidas con pruebas JS, desafío de cookies e incluso captcha.

Brute Force Attack: un ataque de fuerza bruta utiliza prueba y error para adivinar la información de inicio de sesión, claves de cifrado o encontrar una página web oculta. Los hackers trabajan con todas las combinaciones posibles con la esperanza de adivinar correctamente. El "fuerza bruta", significa que utilizan intentos excesivos de fuerza para intentar "forzar" el acceso a cuentas privadas. Tipos de ataques:

- Ataques simples: en donde se intenta adivinar lógicamente las contraseñas , sin la ayuda de herramientas de software u otros medios.
- Ataques de diccionario: son la herramienta más básica en los ataques de fuerza bruta. Se utilizan diccionarios completos y aumentan las palabras con caracteres y números especiales o utilizan diccionarios de palabras especiales, pero este tipo de ataque secuencial es engorroso.
- Ataques de fuerza bruta inversa: revierte la estrategia de ataque al comenzar con una contraseña conocida. Luego, los hackers buscan millones de nombres de usuario hasta que encuentran una coincidencia. Muchos de estos delincuentes

comienzan con contraseñas filtradas que están disponibles en línea a partir de violaciones de datos existentes.

- Relleno de credenciales: si un pirata informático tiene una combinación de nombre de usuario y contraseña que funciona para un sitio web, también lo probará en muchos otros. Dado que se sabe que los usuarios reutilizan la información de inicio de sesión en muchos sitios web, son los objetivos exclusivos de un ataque como este.

_ Como mitigaciones, para evitar este tipo de ataques se tendría que obligar a nuestros usuarios a elegir contraseñas de una cierta longitud con algún carácter especial y/o mayúsculas combinadas con minúscula, también limitar el intento de ingresos de los atacantes a través de un captcha, etc.

_ Otros ataques en aplicaiones web son:

- Fuzzing
- Man-In-The-Middle Attack
- Phishing