APUNTES: FUNDAMENTOS DE LA SEGURIDAD INFORMÁTICA (PARTE 2)

RESUMEN (PARTE 1)

Hablaremos de seguridad, pero no solamente desde una perspectiva tecnológica. La seguridad como concepto es mucho más antiguo que la informática o las computadoras. No es algo que haya nacido con la automatización ni con la llegada de los sistemas de información. Es un concepto que existe desde que la humanidad existe.

Cuando avancemos con el curso, veremos imágenes e historias que demuestran cómo, desde tiempos prehistóricos, los seres humanos ya pensaban en proteger sus bienes, su territorio, su vida y a su familia. En ese entonces no se trataba de proteger información como hoy, sino de proteger algo más vital: la supervivencia.

A lo largo de la historia, siempre han existido dos elementos básicos: **activos** y **amenazas**.

- Un **activo** es cualquier cosa que tenga valor para una persona u organización, y por lo tanto debe protegerse, cuidarse y administrarse correctamente.
- Una **amenaza** es cualquier factor externo que pueda poner en riesgo ese activo.

*Por ejemplo, en la prehistoria, uno de los activos más valiosos era la **vida** y la **familia**. Las amenazas eran múltiples: el hambre, los depredadores y las condiciones climáticas.*

Cuando hablamos de **vulnerabilidades**, nos referimos a las debilidades o características propias que hacen a un activo susceptible de sufrir daños.

> *Por ejemplo, en la antigüedad, las personas eran vulnerables porque carecían de refugios seguros o de herramientas para defenderse. Vivir en espacios abiertos, sin protección, aumentaba esa vulnerabilidad.*

Con el paso del tiempo, los activos que las sociedades protegían fueron evolucionando: ya no era solo la vida física, también surgieron el conocimiento, la religión, los recursos y las riquezas como elementos de valor. Y, por lo tanto, también cambiaron las amenazas y las formas de protección.

> *Por ejemplo, las civilizaciones antiguas protegían conocimientos a través de símbolos, códices y lenguajes cifrados que no cualquiera podía comprender. Esto era una forma de proteger la información, al igual que hoy usamos algoritmos y claves.*

Más adelante, en la Edad Media, las ciudades, los gobiernos y los reinos implementaron nuevas estrategias de protección. Desde castillos, murallas y fortificaciones hasta métodos más sofisticados, siempre existió la necesidad de proteger aquello que consideraban valioso.

Analogía de la Seguridad: De los Castillos a la Seguridad Informática

En los marcos de seguridad, es común usar la metáfora de los **castillos medievales** para ilustrar cómo protegemos nuestros sistemas hoy en día. Así como en la antigüedad se construían castillos en lugares estratégicos —en riscos, cerca de ríos o rodeados de murallas— para defender las posesiones más valiosas como la "habitación del rey" o el "tesoro real", en la actualidad las empresas protegen sus datos y recursos más importantes de la misma forma, solo que en el mundo digital.

La seguridad informática se encarga de **amurallar nuestros sistemas** mediante capas de protección, buscando evitar que los atacantes o amenazas puedan dañar, robar o destruir la información valiosa de instituciones y empresas.

Este enfrentamiento constante entre **atacantes y defensores** se ha transformado en una especie de guerra, donde lo que se defiende ya no es solamente territorio o personas, sino información: archivos, planes estratégicos, recursos, operaciones, ubicaciones y cualquier dato sensible que podría ser explotado si cayera en manos equivocadas.

El Origen de la Criptografía y la Esteganografía

Esta necesidad de proteger la información no es nueva. Ya en la antigüedad, durante las guerras, las civilizaciones desarrollaron técnicas para ocultar mensajes:

- Los **griegos y romanos** usaban tablillas cubiertas con cera, donde escribían un mensaje sobre la madera, lo cubrían con cera, y al parecer era solo una tablilla en blanco. Solo quien conocía el truco sabía que debía raspar la cera para revelar el mensaje oculto.
- Otra técnica consistía en **tatuar mensajes secretos en la piel** de mensajeros, quienes esperaban que el cabello creciera para ocultarlo, y al llegar al destino, se rapaban para revelar la información.

Estas técnicas son las raíces de lo que hoy conocemos como **criptografía** y **esteganografía**, disciplinas fundamentales en la seguridad informática, encargadas de ocultar y proteger la información mientras viaja de un punto a otro.

Aplicación del Principio de Sun Tzu

Estratega militar **Sun Tzu** en *El arte de la guerra*:

"Si conoces al enemigo y te conoces a ti mismo, no debes temer el resultado de cien batallas."

En seguridad informática esto se traduce en dos principios básicos:

1. **Conocerse a uno mismo**: implica entender profundamente la infraestructura, aplicaciones, debilidades y fortalezas de tus sistemas.
2. **Conocer al enemigo**: significa mantenerse informado sobre las amenazas, vulnerabilidades, técnicas y tácticas que los atacantes pueden usar contra tus sistemas.

Quien domina ambos aspectos está en una posición mucho más fuerte para defender sus recursos, anticipar ataques y diseñar estrategias efectivas de protección.

Lo que antes eran fortalezas físicas hoy son **firewalls, sistemas de autenticación, cifrado de datos, VPN y monitoreo constante**. Y aunque las herramientas han cambiado, el objetivo sigue siendo el mismo: proteger aquello que es valioso y evitar que caiga en manos no autorizadas.

La seguridad no solo es una cuestión técnica, sino también de estrategia y conocimiento.

Conocer al Enemigo y al Terreno

- **Importancia de conocer al adversario**:
  Así como en las guerras físicas, en el ciberespacio conocer quién puede atacarte es clave:
  Hackers, delincuentes cibernéticos o atacantes patrocinados por estados.
- **Conocer el terreno (tu sistema)**:
  Si no conoces bien tu infraestructura, es imposible defenderla.
  Hay que saber:
    - Qué dispositivos tienes.

- Qué servicios están corriendo.
- Qué aplicaciones y configuraciones manejas.
- Qué tráfico es legítimo y cuál no.

2️⃣ Evolución de la Seguridad y las Amenazas

- Antes, la seguridad era física: proteger castillos, fronteras y armas.
- Luego, con las guerras mundiales:
  La **criptografía** jugó un rol clave.
  - Ejemplo: la máquina **Enigma** en la Segunda Guerra Mundial.
  - Romper los códigos enemigos cambió el curso de la guerra.
- Hoy en día:
  - Las batallas son **cibernéticas**, no de soldados sino de hackers.
  - Los ataques patrocinados por gobiernos son comunes.
  - Se usa la inteligencia artificial para atacar y defender.

3️⃣ Criptografía y su Desarrollo

- **Antes:**
  Cifrado simple, por ejemplo la **escítala**:
  un método griego que usaba un palo para cifrar mensajes.
- **Segunda Guerra Mundial:**
  - La máquina **Enigma** cambió la historia al encriptar mensajes alemanes.
  - Los británicos rompieron ese cifrado gracias a matemáticos y analistas.
- **Actualidad:**
  - Algoritmos avanzados que combinan matemáticas, teoría del caos, teoría de la información.

  Computación cuántica:
  Puede romper los sistemas de cifrado actuales en segundos, pero también servirá para crear cifrados más fuertes.

4️⃣ La Carrera entre Ataques y Defensas

- Cada vez que aparece una nueva solución de seguridad, tarde o temprano alguien desarrolla una forma de vulnerarla.
- La seguridad es una **competencia constante** entre atacantes y defensores.
- Ejemplo de ataques históricos:
    - 2015: hackeo a **Hacking Team** (empresa italiana de software espía) — les robaron 400GB de datos sensibles.
    - 2017: ataque masivo de ransomware **WannaCry** y **Petya**, afectó a grandes empresas y gobiernos.
- La **seguridad absoluta no existe**.
- La mejor defensa es:
    - Conocimiento continuo.
    - Actualización constante.
    - Preparación para detectar y mitigar ataques.
- Los atacantes siempre buscan la mínima debilidad; no importa si eres una persona, empresa o estado.

SEGURIDAD DE LA INFORMACIÓN

Seguridad de la Información: Más que solo herramientas

La seguridad de la información consiste en proteger la **confidencialidad, integridad y disponibilidad** de los datos. Es una función estratégica del negocio, guiada por el **riesgo**, que involucra **tecnología, procesos, políticas y personal**, bajo la responsabilidad del CISO (Normalmente esta gestión la realiza el director de seguridad de la información (CISO) de la empresa).

SEGURIDAD INFORMÁTICA

La seguridad informática (IT security) es la implementación técnica de medidas como antivirus, firewalls y detección de intrusos para proteger la información, actuar ante incidentes y garantizar la continuidad operativa ante fallas.

-

- Según RAE, seguridad es la "**cualidad de seguro**". Buscamos ahora **seguro** y obtenemos **"libre y exento de todo peligro, daño o riesgo**".
- **La Seguridad como Proceso, No como Producto**

  La seguridad no es un software o una herramienta que se instala y listo. Es un **proceso transversal** que afecta a **todos los activos** y **todas las actividades** de una empresa. Cada organización tiene **necesidades diferentes** según sus objetivos, estructura y entorno.

- **"Un conjunto de métodos y herramientas destinados a proteger la información y por ende los sistemas informáticos ante cualquier amenaza, un proceso en el cual participan además personas"**.

SEGURIDAD DE LA INFORMACIÓN Y SEGURIDAD INFORMÁTICA

La **seguridad de la información** es estratégica y se enfoca en proteger los activos informativos a nivel organizacional. La **seguridad informática** es táctica y operativa, centrada en medidas técnicas. Ambas deben alinearse y complementarse para integrar la seguridad en la cultura y operación del negocio.

En muchas organizaciones existe una separación preocupante entre los niveles que conforman la gestión de la seguridad. Esta desconexión puede debilitar la efectividad global del sistema de protección:

- **Nivel estratégico**: enfocado en el cumplimiento de normativas, certificaciones (como ISO **27001**), auditorías y políticas corporativas. Aquí **se toma la seguridad como parte de la gestión organizacional y se alinea con los objetivos de negocio.**
- **Nivel táctico**: traduce la estrategia en planes de acción concretos, **define controles técnicos, configura arquitecturas seguras y establece procesos de respuesta**. Es el puente entre la visión estratégica y la ejecución técnica.
- **Nivel operativo**: involucra al personal que ejecuta directamente las tareas de seguridad: **técnicos, analistas, administradores, pentesters y blue/red teamers**. Son quienes implementan, monitorean y responden a las amenazas en el día a día.

📊 Esta relación se representa en el siguiente diagrama:

La pirámide muestra cómo la **Seguridad de la Información** se posiciona en el plano estratégico y la **Seguridad Informática** en el plano táctico-operativo, con una relación de retroalimentación constante entre niveles.

🔍 El problema surge cuando estos niveles no están integrados: algunas organizaciones se enfocan exclusivamente en "cumplir" formalmente con los estándares, pero descuidan las acciones técnicas reales necesarias para proteger sus activos. El reto es **articular una visión integrada y coherente** que combine lo normativo, lo técnico y lo operativo, para construir una verdadera cultura de seguridad

- Algunas organizaciones **solo cumplen** por obligación.
- Ejemplo: los bancos reportan pruebas de seguridad por obligación legal, pero no siempre **corrigen las debilidades reales**.
- Se contratan servicios que entregan reportes solo para aparentar cumplimiento, sin resolver problemas.

4️⃣Posturas de Seguridad

- Existen dos grandes **filosofías**:
    1. 🛡️ **Seguridad Defensiva**: proteger, cerrar puertas, configurar firewalls, antivirus, sistemas de detección (IDS/IPS).
    2. 🔓 **Seguridad Ofensiva**: buscar vulnerabilidades, pruebas de penetración, hacking ético, simulaciones de ataque.

La **seguridad informática** enfrenta un desafío constante: los sistemas siempre son susceptibles a vulnerabilidades. Cada nueva configuración, parche o actualización puede introducir brechas inesperadas. Mientras que un **atacante** solo necesita encontrar una falla para comprometer un sistema, los defensores deben anticiparse y cerrar todas las posibles puertas de entrada, lo que convierte esta labor en una tarea compleja y desigual.

SEGURIDAD DEFENSIVA

Enfoque centrado en la configuración técnica de sistemas de seguridad (como antivirus, firewalls, IDS/IPS) y en la detección de amenazas y gestión de incidentes.

SEGURIDAD OFENSIVA

Enfoque basado en análisis de vulnerabilidades y pruebas de penetración (Pentesting), que evalúa la seguridad tanto tecnológica como humana mediante metodologías de Ethical Hacking.

Dentro del ámbito de la **seguridad ofensiva** existen diversos perfiles, comúnmente identificados por categorías simbólicas conocidas como "**sombreros**". Los **White Hats** son hackers éticos que actúan dentro del marco legal para mejorar la seguridad. En contraste, los **Black Hats** persiguen fines maliciosos, explotando sistemas con intenciones delictivas. Entre ambos se encuentran los **Grey Hats**, quienes pueden realizar acciones no autorizadas, aunque sin intenciones maliciosas claras. También están los **Script Kiddies**, usuarios inexpertos que utilizan herramientas sin comprender su funcionamiento. Finalmente, otros perfiles relevantes incluyen **hacktivistas**, **crackers**, **pentesters** y **expertos en exploits**, cada uno con roles específicos dentro del ecosistema de la ciberseguridad.

---

7. Cultura de Seguridad

- Más allá de usar herramientas, se debe:
  - Entender qué hace cada herramienta.
  - Tener conciencia de que la seguridad es un proceso continuo.
  - Fomentar el trabajo conjunto entre la parte técnica y la parte estratégica.

Al descargar **herramientas de pentesting**, es fundamental reflexionar sobre su propósito y su funcionamiento real. Muchas de estas herramientas, especialmente las que circulan en foros o repositorios públicos, no siempre hacen lo que prometen.

> *Por ejemplo, en el pasado circularon cargas maliciosas atribuidas a un actor conocido como Floyd, cuyo exploit destacaba por ser fácilmente reutilizable, aunque requería intervención manual para ser efectivo. Este tipo de exploit era aplicable a sistemas vulnerables como* ***Windows XP****,* ***Windows 2000****, y también versiones más recientes como* ***Windows 10*** *y* ***Windows 11****, similares a las brechas aprovechadas por el malware* ***WannaCry****. A primera vista, parecía sencillo: se descargaba, se ejecutaba, y se esperaba que funcionara. Sin embargo, en muchos casos los resultados no coincidían con las expectativas. A diferencia de los ataques automáticos tradicionales, este código necesitaba ser adaptado y configurado, ya que era público y no venía listo para el ataque directo. Por ello, muchos profesionales lo consideraron más como una prueba de concepto incompleta que como una herramienta lista para producción.*

Este ejemplo refuerza la importancia de **analizar críticamente cada herramienta** antes de usarla: verificar su origen, revisar su código (si es posible), entender sus limitaciones y no asumir que su simple descarga implica funcionalidad o seguridad.

Es fundamental preguntarse: ¿por qué ocurre esto? ¿Realmente se ha leído el código o simplemente **se ha confiado en lo que decía una fuente sin verificarlo?** Muchas veces**, se ejecuta una herramienta sin revisar qué hace**, **a qué servidores se conecta**, qué **bibliotecas usa** o qué **dependencias** incluye. Esta falta de análisis puede llevar a consecuencias graves, tanto funcionales como de seguridad. Este tipo de cuestionamientos es precisamente lo que diferencia a un profesional de un "**script kiddie**". Mientras que los "bajan-ejecutan" actúan sin criterio —disparando herramientas sin saber qué hacen ni por qué—, los **profesionales** se caracterizan por ser **precavidos, analíticos y conscientes de cada paso**. No se trata solo de ejecutar un exploit, sino de **entenderlo, evaluarlo y, cuando es posible, modificarlo**. Esa actitud crítica y metódica es la que define a un verdadero experto en seguridad ofensiva. No basta con descargar una herramienta y asumir que funcionará como se promete. En muchos casos, será necesario revisarla, adaptarla o incluso corregir errores en el código fuente. Esta práctica no solo es común, sino esencial en el entorno real del pentesting. **Personalizar herramientas, entender su lógica interna y ajustarlas para que funcionen correctamente demuestra un nivel técnico superior.** Esas pequeñas modificaciones marcan la diferencia entre quien simplemente ejecuta y quien domina lo que hace.

⚠️ Riesgos del Software Gratuito y la Publicidad Oculta

Otro punto importante es el riesgo asociado al uso de software gratuito, especialmente el que se descarga sin verificación. Algunos programas pueden incluir ***adware***,(**muestra anuncios de forma no deseada o excesiva** en tu dispositivo) funciones no documentadas, o incluso *troyanos* que abren puertos sin consentimiento del usuario. En el **mejor** de los casos, **muestran publicidad**; en el

peor, **comprometen la integridad del sistema**. Muchas personas priorizan la facilidad y la inmediatez sin considerar las implicancias de seguridad, ignorando que al no usar software licenciado o auditado, están exponiéndose a riesgos innecesarios.

El **software libre** representa una gran ventaja, especialmente para quienes tienen recursos limitados.

> *Por ejemplo, en lugar de pagar por licencias costosas como las de **SPSS**, existen alternativas gratuitas y muy potentes, como **R**. R es un software estadístico de código abierto que ofrece enormes capacidades, aunque requiere mayor dedicación: implica escribir más código, leer documentación y estudiar su funcionamiento.*

Aquí es donde cada persona debe evaluar sus prioridades y capacidades. Algunos prefieren la comodidad de interfaces gráficas y funciones automatizadas, aunque eso implique asumir costos o licencias cerradas. Otros optan por herramientas libres, aún sabiendo que la curva de aprendizaje puede ser más exigente.

Lo importante es tomar decisiones informadas. Si alguien no conoce los riesgos o limitaciones, es comprensible que tome el camino fácil. **Pero si sabe lo que está haciendo, también sabe hasta dónde puede llegar**... o incluso dónde están los verdaderos peligros, por decirlo metafóricamente, "dónde está el cementerio".

Entonces, recomendación siempre para todos, no solamente para esto, sino para cualquier cosa que ustedes hagan o se enfrenten. Es como la información que buscan: revisan ustedes, está ChatGPT y todo lo demás, y si le creen todo lo que hay, pues no, pues van a pedir falsos.

🤔 Limitaciones en Pruebas Éticas

En el ámbito del **pentesting** o **hacking ético**, existen varias limitaciones que condicionan el alcance real de las evaluaciones. Aunque se les llama "pruebas", en realidad implican un análisis profundo de vulnerabilidades, no solo una verificación superficial. Sin embargo, una de las principales desventajas es el **control restringido del entorno**.

> *Por ejemplo, cuando una empresa contrata un pentest, suele decir: **"Tenemos 100 servidores, pero solo puedes evaluar estos 20."** Esto significa que el análisis es **limitado por decisión del cliente**, lo cual puede dejar activos críticos fuera del alcance. Además, las **restricciones de tiempo** son comunes: **"Tienes 10 días para hacerlo."** En muchos casos, el tiempo se reduce drásticamente, dificultando una evaluación completa y profunda. Incluso sucede que, durante el proceso, el propio cliente interrumpe con afirmaciones como: **"Ya encontramos una vulnerabilidad, así que ahí nomás."***

Esto demuestra que, aunque se trate de pruebas éticas y controladas, **los límites impuestos pueden comprometer los resultados** o dejar riesgos latentes sin identificar.

🚫 Límites en la Exploración de Vulnerabilidades

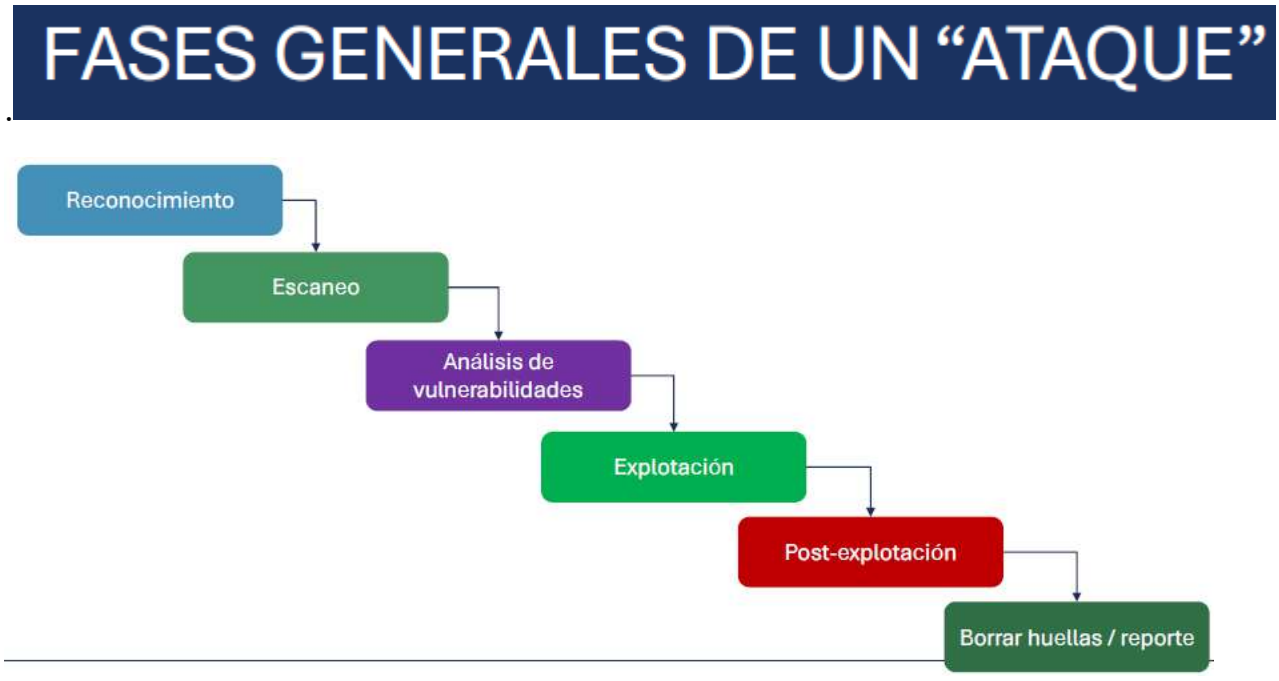
En muchas ocasiones, al identificar una vulnerabilidad —

*por ejemplo, una de **ejecución remota de código (RCE)**—, la respuesta inmediata por parte del cliente suele ser contundente: **"Hasta ahí nomás. No sigas."***

Este tipo de restricciones no responde a una falta de capacidad técnica, sino que forman parte de las **políticas internas del cliente** o de las **cláusulas del contrato de servicios**. Una vez que se ha demostrado la existencia de una falla, se solicita detener el análisis, incluso si existen indicios de que se puede profundizar o escalar el hallazgo.

*La instrucción es clara: **"Con eso basta. Ya está. Solo era para el informe técnico o académico."***

Aunque el equipo esté preparado para continuar la investigación y explorar otras capas del sistema, el trabajo debe darse por concluido conforme a los términos acordados. Estas limitaciones son comunes en pruebas de penetración y auditorías de seguridad, y suelen obedecer a **motivos legales, éticos o contractuales**. En este contexto, no se trata de una decisión técnica, sino de **respetar el alcance autorizado**, algo fundamental en el ejercicio profesional del hacking ético.



🔍 1. Reconocimiento

Fase inicial donde se recopila información sobre el objetivo (empresa, red, sistemas). Puede ser pasivo (sin interactuar con el objetivo) o activo.

*Buscar dominios asociados a una empresa usando herramientas como **WHOIS**, **Shodan** o **Google Dorking** para obtener IPs públicas y servicios expuestos.*

📡 2. Escaneo

Identificación activa de puertos abiertos, servicios en ejecución y versiones de software del sistema.
*Utilizar **Nmap** para descubrir que el puerto **22 (SSH)** está abierto en una IP del objetivo, indicando un posible vector de ataque.*

🔍 3. Análisis de Vulnerabilidades

Evaluar los servicios detectados para identificar fallos o configuraciones inseguras que puedan ser aprovechadas.

*Escanear con **Nessus** o **OpenVAS** y detectar que el servidor web Apache 2.4 tiene una vulnerabilidad crítica (por ejemplo, CVE-2021-41773).*

💥 4. Explotación

Usar la vulnerabilidad identificada para obtener acceso no autorizado o ejecutar código malicioso.

*Usar **Metasploit** para explotar la vulnerabilidad del Apache 2.4 y conseguir una shell remota en el servidor.*

🛠️ 5. Post-explotación

Consolidar el acceso, escalar privilegios, mover lateralmente y recolectar información valiosa del sistema comprometido.

*Buscar archivos con contraseñas guardadas, capturar tokens o intentar moverse a otros equipos en la red interna usando **Mimikatz** o **BloodHound**.*

🧹 6. Borrar huellas / Reporte

En un entorno ético: documentar todo en un reporte técnico para la empresa. En un entorno malicioso: borrar logs y ocultar rastros para no ser detectado.

*Elaborar un **informe detallado** con evidencias, pasos reproducibles, impactos y recomendaciones de mitigación. En pruebas reales, evitar dañar o dejar puertas abiertas.*

🛡️ Análisis de Vulnerabilidades: Límites y Realidades

En muchos escenarios, el análisis de vulnerabilidades se detiene justo después de confirmar la falla, sin llegar a la explotación ni a las fases de post-explotación. Me ocurrió recientemente con un caso en el que se detectó **acceso no autenticado a un servicio expuesto**. Al revisar, era posible volcar las credenciales de la base de datos asociadas al usuario del servicio. Sin embargo, el cliente indicó de inmediato: **"Como ya se encontró la vulnerabilidad, hasta ahí nomás."** ¿Dónde estaba exactamente la falla? La clave fue la **ausencia total de autenticación**. Bastaba acceder a la dirección, al puerto del servicio, y se obtenía acceso directo sin ninguna validación. A pesar de ello, no se autorizó continuar.

Esta situación lleva a una pregunta común: **"¿Hasta dónde se podría llegar si se explotara esto?"** Y la respuesta posible es: **a datos sensibles, a información crítica, incluso a credenciales reutilizadas en otros servicios**. Pero si no se permite avanzar, ni el equipo de seguridad ni el cliente conocerán con certeza el verdadero impacto. El problema es que **un atacante real no esperará esa autorización**: si encuentra la brecha, explotará todo lo que pueda sin restricciones.

🔍 Servicios Vulnerables y Descuidos Comunes

El servicio en cuestión era conocido históricamente por vulnerabilidades, pero aún así estaba en uso sin protección adecuada. Lo preocupante es que **este tipo de servicios poco comunes o poco documentados suelen ser ignorados**, no por su complejidad, sino por la falta de políticas de control estrictas.

Muchas organizaciones fortalecen sus sistemas principales: contraseñas robustas para bases de datos, políticas de acceso duro al sistema operativo, etc. Sin embargo, **descuidan pequeños servicios auxiliares o componentes menos visibles**, donde justamente se esconden las puertas traseras. Esto **no es una defensa en profundidad**, sino una seguridad desigual, lo cual termina siendo un riesgo serio. En seguridad, **no basta con proteger los puntos más visibles**. Lo crítico también está en lo olvidado.

Al realizar ejercicios, buscar información o resolver desafíos técnicos, es importante adoptar un enfoque mental adecuado. Cuando se les recomiende realizar una actividad práctica, no la aborden simplemente como una tarea más. **Pónganse en el rol del atacante**, no del defensor. Este cambio de mentalidad les permitirá anticipar escenarios reales, comprender mejor las amenazas y, sobre todo, desarrollar un pensamiento estratégico más completo.

Una pregunta clave que deben hacerse es: **¿Qué haría un atacante en esta situación?** Muchas veces, se cae en suposiciones como: *"No, seguramente ese servicio está cerrado. Imposible que un banco lo tenga expuesto."* Y por ese tipo de prejuicios, se omiten pruebas que podrían revelar

vulnerabilidades reales. No se trata de ser pesimistas, sino de ser **curiosos y persistentes**. Siempre pregunten: ***¿qué más podría haber ahí***? Esa es la mentalidad ofensiva que se necesita en un entorno real de análisis de seguridad.

📚 Conocimiento Técnico Detallado

Otro aspecto fundamental es el conocimiento técnico. Si no conocen un servicio o un puerto específico, **investiguen, lean, prueben**. La falta de familiaridad no debe ser una excusa para detenerse. Como se mencionó antes, si no conoces el terreno en el que estás operando, es probable que pases por alto información crucial o que cometas errores importantes. Entender a fondo los sistemas, protocolos y servicios es lo que permite actuar con eficacia. Y si no sabes, **no lo veas como una limitación**, sino como una oportunidad de aprendizaje.

⚔️ Defensa y Ofensiva: Roles que se Complementan

Dentro del ámbito de la ciberseguridad, los roles ofensivos y defensivos no son opuestos, sino **complementarios**. El **objetivo** de las **pruebas ofensivas** no es simular, sino **emular al atacante real** para **probar la efectividad de las defensas**. No se trata de una competencia entre equipos, sino de una **colaboración estratégica**. Tanto los **defensores** como los **ofensores** trabajan hacia el mismo fin: **mejorar la postura de seguridad de la organización**.

Lamentablemente, en la práctica, esta relación muchas veces se malinterpreta. En algunos entornos, quienes ejercen un rol **ofensivo** son vistos como "**los malos**". Nada más alejado de la realidad. Su función es **señalar debilidades**, **no para criticar, sino para fortalecer**. Incluso hay ocasiones en las que el equipo ofensivo **no logra vulnerar el sistema**, y eso también es un **buen resultado: significa que la defensa está funcionando correctamente**, al menos dentro del tiempo y los recursos asignados.

🔐 Fundamentos de Seguridad Informática

1. Evaluaciones de Seguridad:

Una Fotografía Temporal

Las pruebas de seguridad, como los análisis de vulnerabilidades o las auditorías, ofrecen una visión limitada en el tiempo: son solo una “fotografía” del estado de seguridad en un momento específico. Esta fotografía puede quedar desactualizada rápidamente, ya que en un periodo de 3 a 6 meses pueden producirse cambios que alteran completamente el panorama: instalación de **parches**, **actualizaciones** del sistema, **cambios de configuración**, **migración de servicios**, **nuevas versiones** de software o la **integración de nuevo código**. Por ello, es fundamental considerar la seguridad como un proceso continuo y no como una tarea puntual.

---

2. Equipos Azul y Rojo:

Defensa y Simulación Ofensiva
Para fortalecer su postura de seguridad, se recomienda que las organizaciones adopten un enfoque dual basado en los equipos Azul y Rojo. El **equipo Azul** se encarga de la **defensa y monitoreo continuo de los sistemas**, mientras que el **equipo Rojo** **emula el comportamiento de atacantes reales** para poner a prueba esas defensas. La práctica regular de ejercicios controlados entre ambos equipos permite detectar brechas, evaluar la capacidad de respuesta y mejorar los controles internos.

---

3. Cultura de Seguridad y el Valor de la Ofensiva
En muchas empresas ya se ha institucionalizado la existencia de roles ofensivos y defensivos. Los perfiles **ofensivos**, sin embargo, **requieren más que habilidades de ataque**: deben poseer **conocimientos sólidos en desarrollo de software, análisis de aplicaciones y comprensión profunda de arquitecturas y protocolos**. Si bien no es obligatorio ser programador para dedicarse a la seguridad ofensiva, tener **habilidades de codificación** ofrece una ventaja significativa al momento de identificar y explotar vulnerabilidades de forma controlada y ética.

---

4. Servicios Profesionales Externos: Pentest vs Red Team
Las empresas que buscan fortalecer su seguridad también pueden recurrir a servicios externos especializados, como las **pruebas de penetración (pentesting)** o los **ejercicios de Red Team**. Mientras que el pentest se enfoca en revisiones técnicas bajo un alcance claramente definido, el Red Team realiza simulaciones ofensivas más realistas, sin alertar a los defensores, para evaluar tanto las tecnologías como los procesos y la respuesta humana. Ambas estrategias son esenciales para identificar debilidades antes de que los atacantes reales las exploten.

---

5. Aplicaciones y Seguridad Durante la Pandemia
Durante la pandemia, muchas organizaciones se vieron obligadas a lanzar aplicaciones rápidamente, priorizando la operatividad sobre la seguridad. Esto trajo como consecuencia sistemas con fallas críticas, interfaces expuestas y formularios web mal protegidos. Un caso común fue el de las plataformas de “Mesa de Partes Virtuales”, que en muchos casos actuaban como verdaderas coladeras de seguridad debido a la ausencia de pruebas y controles previos.

---

6. Proceso de Revisión de Seguridad
Cualquier nuevo software o actualización debe pasar por un proceso formal de revisión de seguridad. Esto incluye la evaluación de vulnerabilidades conocidas, el análisis de posibles escenarios de

explotación y la aplicación de medidas de mitigación. Aunque es imposible garantizar una seguridad absoluta, este enfoque permite **reducir drásticamente la superficie de ataque** y disminuir el impacto potencial ante un incidente.

---

7. Limitaciones y Ventajas del Atacante

Una diferencia clave entre los servicios éticos y los ataques reales es el **límite de tiempo**. Los servicios profesionales como el pentesting o el Red Teaming están restringidos a contratos de corta duración: 10, 15, 30 días o, en casos más extensos, hasta 3 meses. En contraste, un atacante real no tiene tales restricciones: puede permanecer en una red durante meses o incluso años sin ser detectado, aprovechando al máximo cada brecha no identificada. Esta diferencia subraya la importancia de realizar auditorías frecuentes y contar con capacidades de detección y respuesta activa.

El video aborda temas de ciberseguridad y cómo han evolucionado las amenazas en el tiempo. Aquí hay un resumen de los puntos principales:

1. **Evolución de los virus**: Hace 13 o 14 años, los virus eran fácilmente detectables y su propósito era evidente, como mostrar al autor. Hoy en día, los virus están diseñados para pasar desapercibidos, recolectando información sin que el usuario se dé cuenta.
2. **Diversificación de las amenazas**: Las formas de entregar malware se han diversificado. Antes, un archivo adjunto era suficiente para infectar una máquina, pero ahora se utilizan diferentes canales y métodos para propagar los virus.
3. **Seguridad en transacciones bancarias**: Aunque las transacciones bancarias están cifradas, si un dispositivo está infectado, se pueden capturar los datos de la tarjeta y lo que el usuario teclea. La autenticación de dos factores (2FA) se ha implementado como medida adicional, pero no es completamente infalible. Los ciberdelincuentes también han encontrado formas de evadir este sistema.
4. **Riesgos asociados al teléfono móvil**: Los teléfonos móviles se han convertido en un objetivo principal para los ciberdelincuentes, ya que contienen mucha información sensible. El robo de dispositivos ahora es más peligroso que antes debido a la cantidad de datos que almacenan.
5. **Evasión del doble factor**: Existen métodos para evadir la autenticación de dos factores, como alterar el flujo de la aplicación o predecir el código de autenticación. Algunas aplicaciones tienen fallos de seguridad que permiten eludir este mecanismo de protección.
6. **Pruebas de seguridad**: Las pruebas de seguridad, como la inyección de código (SQL injection) o ataques de scripting, son esenciales para detectar vulnerabilidades en los sistemas y aplicaciones. La correcta implementación de la autenticación de dos factores y otras medidas es crucial para proteger las plataformas.

Este video destaca cómo las amenazas de seguridad han evolucionado, pero también resalta que las medidas de protección, como la autenticación de dos factores, siguen siendo esenciales, aunque no infalibles.

Este segmento aborda temas de seguridad digital, las transformaciones que han ocurrido en el uso de tecnologías y los riesgos asociados, así como la importancia de los tres pilares fundamentales de la seguridad informática: confidencialidad, integridad y disponibilidad. Aquí está un resumen de los puntos clave:

1. **Reacio al uso de aplicaciones bancarias**: El amigo mencionado en el texto se muestra reacio a hacer transacciones bancarias en línea. Sin embargo, los bancos están obligando a los usuarios a realizar transacciones a través de sus aplicaciones móviles, lo que aumenta la exposición a riesgos.
2. **Evolución de las redes sociales y aplicaciones**: En el pasado, se usaban plataformas como Hi5 y Yahoo, pero con el tiempo las redes sociales y aplicaciones como WhatsApp han ganado popularidad. El número de aplicaciones y servicios conectados a internet ha crecido considerablemente, lo que también ha incrementado los riesgos de seguridad.
3. **Malware sofisticado en infraestructuras críticas**: Se menciona el uso de malware como Stuxnet, que fue utilizado para atacar instalaciones industriales, como plantas nucleares y redes de electricidad. Este tipo de ataque está dirigido a sistemas críticos que controlan infraestructuras vitales, como el agua potable y los servicios de comunicación.
4. **Internet de las Cosas (IoT)**: Los dispositivos conectados a Internet, conocidos como IoT, también han sido un objetivo para los ciberdelincuentes. Estos dispositivos pueden estar involucrados en ataques a sistemas industriales o infraestructuras críticas, lo que aumenta el riesgo de ciberataques.
5. **Tres pilares de la seguridad informática**: Se resaltan los tres principios fundamentales de la seguridad: confidencialidad, integridad y disponibilidad.
    - **Confidencialidad**: Se refiere a asegurar que solo usuarios autorizados puedan acceder a información privada. Ejemplo: tu correo electrónico, que es confidencial.
    - **Integridad**: Implica que los datos no sean alterados sin autorización. Ejemplo: un certificado digital que ha sido modificado, pierde su integridad.
    - **Disponibilidad**: Aunque no se profundiza en el texto, este principio asegura que los datos y servicios estén accesibles cuando se necesiten.

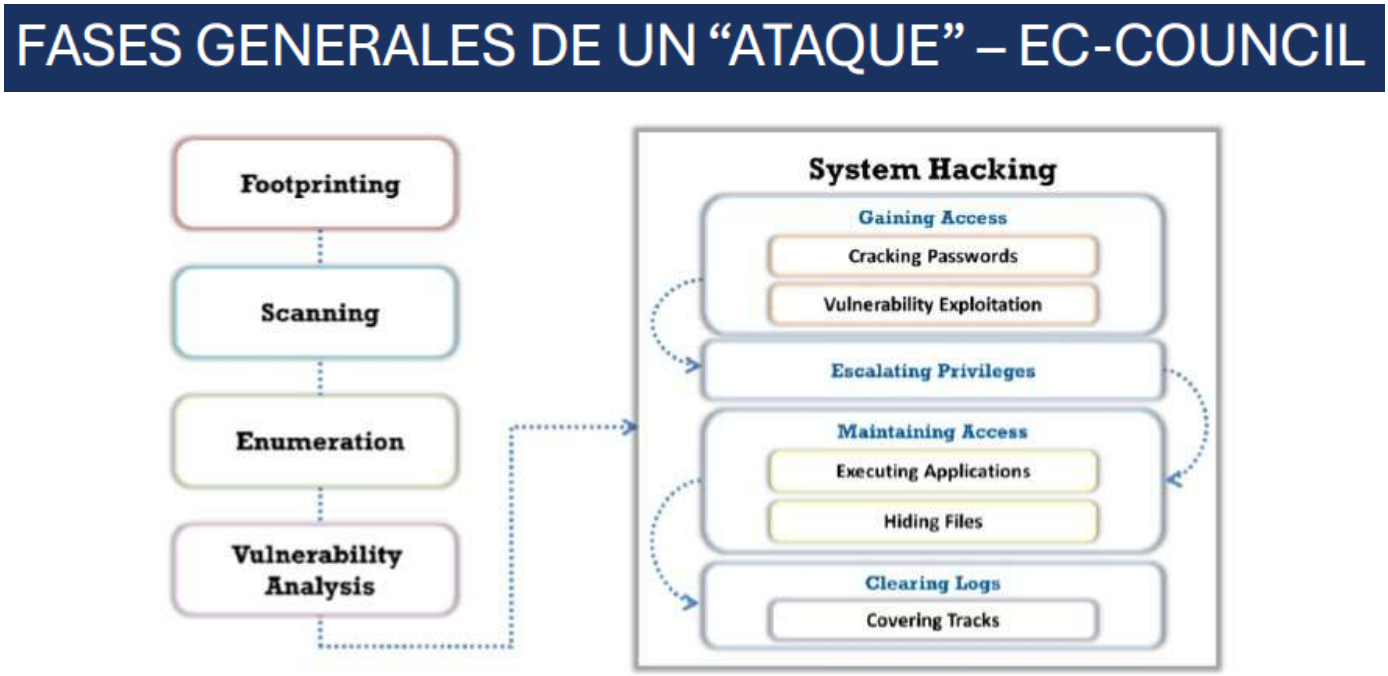
Este resumen enfatiza cómo la evolución de la tecnología, la sofisticación de los ataques y el incremento de dispositivos conectados a internet han elevado los riesgos de seguridad, destacando la importancia de proteger la confidencialidad, la integridad y la disponibilidad de los datos.

En esta transcripción, se profundiza en los tres pilares fundamentales de la seguridad informática: **confidencialidad**, **integridad** y **disponibilidad**, explicando cada uno con ejemplos. Aquí tienes un resumen de los puntos clave:

1. **Confidencialidad**: Se refiere a proteger la información para que solo los usuarios autorizados puedan acceder a ella. En el ejemplo dado, se menciona la confidencialidad de la información en un sistema de gestión académica, donde solo los usuarios con privilegios (como docentes) pueden modificar ciertos datos, como las notas.
2. **Integridad**: Se asegura que la información no sea alterada de manera no autorizada. Si un dato es modificado sin la debida autorización, pierde su integridad. El ejemplo dado menciona que un docente tiene el privilegio de modificar las notas, pero no cualquier persona puede hacerlo sin autorización.
3. **Disponibilidad**: Se refiere a que la información debe estar disponible cuando sea necesaria para los usuarios autorizados. Se discute un ejemplo sobre los horarios de disponibilidad de un sistema, donde el sistema está disponible de 7 a 3, y si no está disponible fuera de ese horario, no se afecta la disponibilidad según los requisitos establecidos.

También se plantea la situación en que un sistema esté disponible solo en un horario específico, como los servicios bancarios que solo operan en ciertas horas. La disponibilidad se mide según los requisitos establecidos y no necesariamente debe ser 24/7.

En resumen, **confidencialidad** asegura el acceso restringido, **integridad** garantiza que los datos no se alteren sin permiso y **disponibilidad** asegura que los datos estén accesibles cuando se necesiten, de acuerdo con los requisitos establecidos.



🧭 1. Footprinting (Reconocimiento)

Primera fase en la que se recolecta información del objetivo sin interactuar directamente. Por ejemplo, se puede buscar subdominios con herramientas como *crt.sh*, investigar registros WHOIS o encontrar sistemas expuestos usando Shodan.

📡 2. Scanning (Escaneo)

Se identifican puertos abiertos, servicios activos y versiones de software. Una herramienta como Nmap puede revelar que los puertos 22 (SSH) y 80 (HTTP) están abiertos, y que se ejecuta Apache 2.4.49.

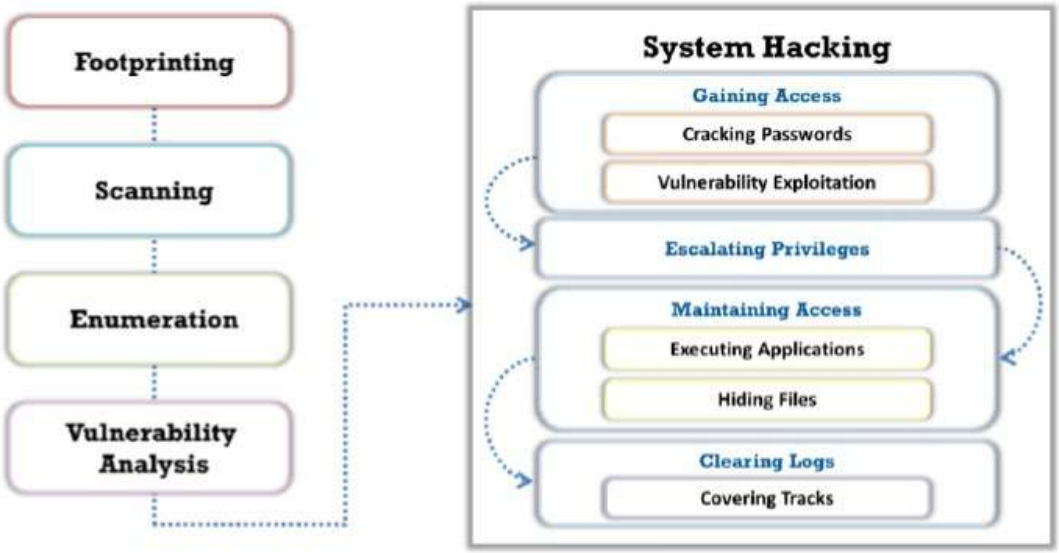
📜 3. Enumeration (Enumeración)

Se extrae información detallada de los servicios descubiertos. Por ejemplo, con enum4linux se listan usuarios de un servidor Samba, o con SNMPwalk se obtiene la configuración de red de un equipo.

🧪 4. Vulnerability Analysis (Análisis de Vulnerabilidades)

Se identifican fallas en los servicios encontrados. Con Nessus o OpenVAS se puede detectar que la versión de Apache usada tiene una vulnerabilidad crítica (como CVE-2021-41773).

💻 System Hacking



🔓 Gaining Access (Obtener acceso)

Aquí se ejecuta el ataque que permite ingresar al sistema. Por ejemplo, se puede usar Metasploit para explotar Apache y abrir una shell remota, o hacer fuerza bruta de contraseñas SSH.

⬆️ Escalating Privileges (Escalar privilegios)

Se intenta pasar de usuario común a administrador. Esto puede lograrse explotando una falla del sistema operativo (como Dirty COW en Linux) o usando Mimikatz para robar credenciales de administrador.

🔄 Maintaining Access (Mantener el acceso)

Se instalan mecanismos para volver a ingresar después. Esto incluye dejar un webshell en un servidor o crear una cuenta oculta con permisos elevados.

🧹 Clearing Logs (Borrar huellas)

Se eliminan rastros del ataque. Por ejemplo, se borra el historial de comandos en bash, se limpian los logs del sistema o del visor de eventos en Windows.