# Evolución del Malware

**Marcos Fernando Reyes**[1]

[1] **Facultad de Ingenierías, Universidad, mreyes@unisangil**

RESUMEN

**En la evolución del mundo computacional, desde los primeros ensayos tecnológicos, la humanidad ha buscado automatizar los diferentes procesos cotidianos, pero paralelamente en ese trasegar, se ha dado el paso camuflado del malware o virus, apareciendo casi por accidente, en una serie de eventos que se relatan en este documento y se van convirtiendo en una opción de vida para aquellos perfiles cibernéticos conocidos como black hat; con el paso del tiempo las tecnologías modernas buscan ser ya impermeables ante las amenazas, sin embargo, los creadores de malware siempre buscan dar un paso adelante, así como se relacionan tecnologías como Android, el siguiente paso que están dando las amenazas es la nube, la computación convergente, y el ecosistema digital en general, donde muy seguramente muchos delitos dejaran de ser presenciales en su mayoría y se convertirán en cibernéticos, haciendo al delincuente menos visible ante la sociedad, en su búsqueda de ocultarse en la red y buscando borrar las evidencias.**

***Palabras Clave*— Android Tamer, Ingeniería Inversa, Lenguaje Ensamblador Virus.**

ABSTRACT

**In the evolution of the computational world, since the first technological trials, humanity has sought to automate the different daily processes, but in parallel in that transfer, the camouflaged step of malware or viruses has been taken, appearing almost by accident, in a series of events that are recounted in this document and become a life choice for those cyber profiles known as black hat; With the passage of time modern technologies seek to be already impervious to threats, however, malware creators always seek to take a step forward, as well as related technologies like Android, the next step that threats are taking is the cloud, convergent computing, and the digital ecosystem in general, where many crimes will most likely cease to be in person and become cybernetic, making the offender less visible to society, in their quest to hide in network and looking to erase the evidence.**

***Key Words* Android Tamer, Reverse Engineering, Language Assembly Virus.**

## I. INTRODUCCIÓN

Desde épocas atrás, el ser humano ha buscado crear algo tan extraordinario como el cerebro humano, lo cual ha sido una misión maratónica y aunque esté empezando a vislumbrar algunas luces lejanas al final del túnel, se está solo en el principio de esa aventura.

En la segunda guerra mundial, usaron un desarrollo "tecnológico", para obtener una victoria, gracias a "Bombe", la máquina creada por Alan Turing, complementado con algunas operaciones de inteligencia, fueron vitales para que los aliados pudiesen descifrar los mensajes alemanes, conociendo sus movimientos, acciones y planes, permitiéndoles posteriormente elaborar estrategias de contra inteligencia militar, y poder ganar la guerra, derrotando al ejército nazi comandado por Hitler. Desde ese entonces, la máquina de Turing, "Bombe" es considerada la base de la informática moderna. [1]

Las primeras computadoras eran grandes salones llenos de pantallas, teclados y circuitos, que al principio servían para sumar y restar, las cuales buscaban imitar la capacidad del cerebro humano de crear cosas nuevas, por lo tanto esas máquinas monumentales, con el tiempo fueron reduciendo su tamaño físico, y aumentando su capacidad lógica, realizando operaciones mucho más complejas, cálculos avanzados, aplicaciones que realizaban simulaciones, proyecciones y diferentes tareas, se hablaba de tarjetas perforadas, semiconductores y procesadores, luego de disquetes de 5 ¼ y de 3 ½, que almacenaban menos de un bit de información, pero que en ese momento eran lo máximo. [2]

Las computadoras siguieron evolucionando, mejorando sus sistemas operativos, sus core fueron más robustos, se desarrollaban vertiginosamente aplicaciones por medio de consola, la conocida pantalla azul de c++ [3], turbo pascal [4], cobol [5], fortran [6], entre otros, los códigos eran bastante extensos, aun no existía la programación orientada a objetos, se tenía conocimiento de assembler [6], pero era casi imposible pensar en una simulación en 3D.

## II. RETROSPECTIVA DESDE MS-DOS HASTA LA ACTUALIDAD

### *A. A través del tiempo*

Primer Autor: mreyes@unisangill.edu.co, Esp. Seguridad Informática - Ingeniería de Sistemas, Unisagil.

En la época aún del Ms-Dos, los virus eran un tanto diferentes de los actuales, el alcance y objetivo principal de los atacantes en ese momento tecnológico era el de destruir la información, o quizás el de transmitir mensajes filosóficos, religiosos o políticos.

Un ejemplo fue el virus Walker, una persona que salía caminando de lado a lado de la pantalla mientras la información seleccionada por el usuario aparecía y desaparecía. Esta era la época en donde el disquete era el principal dispositivo extraíble y contagiaba equipos a través de su uso, dando pie a que aparezcan los primeros antivirus, buscando dar solución a dichas problemáticas, que en ocasiones también buscaban afectar el sector 0 del disco duro y causar pérdidas de información irreparables. [8]

Con los Cd's y los primeros antivirus a gran escala, por un tiempo se tuvo una sensación de tranquilidad, hasta cuando las vulnerabilidades en los sistemas operativos, servidores y páginas web empezaron a hacerse más evidentes, siendo atacados y hackeados múltiples servidores, aprovechando las vulnerabilidades del joven joomla [9], del código html puro y de los sistemas operativos configurados por default.

Las aplicaciones para pagos por internet trataban de generar confianza, mientras los piratas informáticos, crackers y hackers black hat se divertían robando tarjetas de crédito y haciendo transacciones de forma ilícita, generando mucho temor en los usuarios del ciberespacio. [10]

Se había entrado en una nueva era, la de la web, de la información ágil y fácil de transportar, llevando grandes cantidades de información en un pequeño dispositivo, aunque muy costoso al principio para la época, pero una muy valiosa novedad, ya no eran muchos disquetes ni cds, ya eran memorias de determinada cantidad de gigabytes que facilitaban el trabajo de los habitantes de la web, satisfaciendo esa necesidad.

Hoy en día, esas grandes capacidades se han trasladado a las memorias sd y más específicamente a los celulares, la nube y el ecosistema digital, en la actualidad podemos hablar de un ecosistema en el cual convive la información, en el que todo está relacionado entre sí, donde un correo electrónico es la llave para abrir muchas puertas, espacios en drive, copias de seguridad de whatsapp, listas de contactos, agendas, fotos, videos, información en general que una vez es almacenada en el móvil, empieza a compartirse a través de su nube por las diferentes cuentas.

### *B. Evolución de las Amenazas*

Se realiza una breve reseña histórica de algunas de las amenazas más resonantes [11] desde sus principios hasta el momento: en esta reseña recordamos algunos malware de la época de los 80 y los 90 [12], también algunos virus "raros" a través de la historia [13], con objetivos, origen e impactos muy discutibles, los orígenes del famoso ransomware [14], malware programados con temáticas como la política, la religión, la sociedad o el arte [15], y la historia de los virus en una forma muy general [16].

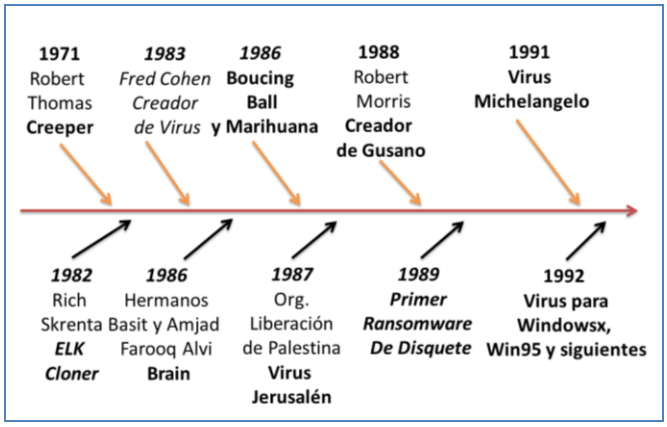


Fig 1. Evolución de las Amenazas 1

En la figura 1 se puede confirmar como en la primera etapa del malware, los primeros atacantes facilitaban la información para identificarse y que todo el mundo supiera quien era el autor de cada ataque, pero desde mitad de los años 90 aproximadamente debido al endurecimiento de las leyes y castigos para los daños y perjuicios informáticos, los atacantes se han ocultado mucho más dejando ver específicamente la información de sus creaciones, pero ocultando en lo posible sus identidades.

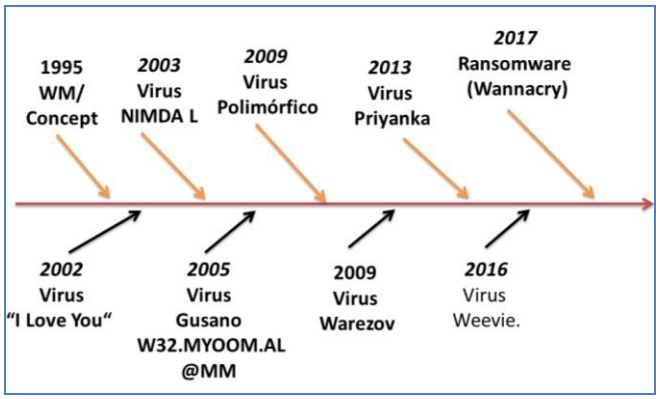


Fig 2. Evolución de las Amenazas 2

## III. INGENIERÍA INVERSA

La ingeniería inversa es un concepto que busca analizar el código fuente de las aplicaciones, con el fin de comprender su funcionamiento, interacción con los dispositivos físicos, comprender los métodos y procesos de desarrollo utilizados, en muchos casos como guía para el desarrollo de nuevos proyectos. También puede ser utilizada para analizar

vulnerabilidades, problemas, incompatibilidades, o simplemente por curiosidad tecnología de explorar y descubrir cómo funcionan las cosas. [17]

La ingeniería inversa como muchas de las creaciones que han existido en el mundo fue algo pensado con la intención de aportar positivamente al desarrollo y bienestar del ser humano, pero así como ha sucedido a través de los años, las fuerzas oscuras o aquellos que están del lado de la sombra también la han sabido utilizar a su favor, de formas anti éticas, en donde han aprovechado para obtener información, dinero u otros beneficios recurriendo a dichas prácticas, con la creación de virus [18], exploits, aprovechando vulnerabilidades en los sistemas informáticos.

Hoy en día, las vulnerabilidades de los sistemas informáticos se analizan con scanner, pero en un principio muchos personajes en las sombras iniciaban esas experiencia a prueba y error, instalando aplicaciones, sistemas operativos, introduciéndose en su núcleo y buscando a profundidad los fallos a causa de los errores de los programadores, abriendo esos códigos y explorándolos, para posteriormente construir aplicaciones maliciosas que aprovechasen estas ventajas.

En la evolución del malware, se han identificado algunas prácticas de los atacantes informáticos, tales como: crackerar aplicaciones (en esta lista han entrado por muchos años las encartas, officce, maltlab, corel drawn, video juegos y muchas otras aplicaciones muy populares y comunes en el mercado), vender la información robada, e incluso extorsionar a las víctimas a cambio de su información cifrada, como lo hacen con el Ransomware [19].

Con el tiempo algunas plataformas en especial las de juegos han sido pioneros en montar sus plataformas en la nube y de esta forma combatir la piratería, obligando así a los usuarios a pagar por los servicios de nube para acceder a sus servicios, ya sin tener instaladores directamente en muchos de los casos o con los requerimientos de pago para usar en línea las aplicaciones.

De esa forma solo los clientes registrados en la nube, pueden aprovechar todas sus propiedades de dichas aplicaciones, ayudando a reducir en cierta forma las pérdidas económicas para sus empresas, sin embargo hay que aclarar que los atacantes siempre están buscando permanentemente la forma de vulnerar sus objetivos, y ya han existido muchos casos de creaciones de malware que han vulnerado plataformas web, generando caos entre sus usuarios.

### *A. Depuradores, Desensambladores, Decompiladores y lenguaje ensamblador (Assembler).*

Es importante identificar estos conceptos, ya que han sido fundamentales en el transcurso de la historia para los programadores de software, y de igual manera para los desarrolladores de malware, logrando así, aplicar la ingeniería inversa por medio de estas aplicaciones, para editar los códigos y lanzar las aplicaciones maliciosas, aprovechando esas vulnerabilidades que lograban encontrar, gracias al estudio cuidadoso del código producido por los programadores.

Los depuradores son aplicaciones que permiten hacer seguimiento a un código paso a paso, por ejemplo, para Windows: OllyDbg y WinDBG, los cuales son gratuitos y son muy útiles para quienes realizan procesos de ensamblado y desensamblado. [17]

Los desensambladores son programas que traducen los binarios de código máquina a un lenguaje ensamblador, permitiendo así que un programa o software que ya está empaquetado o compilado en un .exe, .dll, .jar, .vbs, bas, o cualquier extensión, y así poder ver el código fuente del mismo, y dependiendo del tipo de desensamblador puede obtenerse el código en assembler, o en el mismo lenguaje en el cual fue construido.

Por ejemplo existen desensambladores para Java o para Visual, que permite abrir el proyecto al igual como lo haría el desarrollador original, realizando alguna modificación y volviendo a compilarlo.

Algunos desensambladores conocidos son: IDA Pro, PE explorer, IDA Pro Freeware 4.1, Bastard Disassembler, Ciasdis, entre otras herramientas que pueden llegarse a encontrar en el mercado y que tienen un objetivo general de mostrar en claro el código fuente, pero cada uno con sus características. [17]

Un decompilador es una herramienta que transforma código en ensamblador o código máquina en código fuente en lenguaje de alto nivel. También existen decompiladores que transforman lenguaje intermedio en código fuente de lenguaje de alto nivel.

Los decompiladores son parecidos a los desensambladores pero llevan el proceso un importante paso más allá. Algunos decompiladores conocidos son: DCC Decompiler, Boomerang Decompiler Project, Reverse Engineering Compiler (REC). [17]

El lenguaje ensamblador tiene sus características particulares y diferenciadas de un lenguaje de alto nivel, pues assembler es la comunicación más cercana a la máquina, también llamada de bajo nivel, por ende envía prácticamente órdenes directas a los componentes de hardware que deben ser ejecutados.

En la actualidad el lenguaje ensamblador no es tan popular

en los ingenieros de sistemas actuales, puesto que se están usando directamente herramientas más light, de desarrollo rápido, y ya de mucha más cercanía con el usuario final. [17]

**Push:** Guarda el valor en la pila.
**Pop:** Recupera valor de la pila.
**Mov (dst, src):** Copia el operador "src" en el operador "dst".
***Lea:*** (reg, src): Copia una dirección de memoria en el registro destino (ej: EAX).
***Add:*** (o1, o2): Suma los dos operadores y los almacena en el operador uno.
**Sub:** (o1, o2): Resta el valor del segundo operador sobre el primero y lo almacena en el primer operador.
**Inc:** Incrementa en 1 el valor del operador indicado.
**Dec:** Decrementa en 1 el valor del operador indicado.
**And:** El resultado es 1 si los dos operadores son iguales, y 0 en cualquier otro caso.

Fig 3. Acciones comunes dentro del lenguaje ensamblador.

En la fig 4, se recopilan y listan algunos componentes afines a ese lenguaje máquina como son los siguientes:

***EAX (Accumulator register):*** Utilizado tanto para realizar cálculos, cómo para el almacenamiento de valores de retorno en "calls".

***EDX (Data register):*** Extensión de EAX, utilizada para el almacenamiento de datos en cálculos más complejos.

***ECX (Count register):*** Utilizado en funciones que necesiten de contadores, como por ejemplo bucles.

***EBX (Base register):*** Se suele utilizar para apuntar a datos situados en la memoria.

Fig 4. Componentes lenguaje ensamblador

*B. Android Tamer*

Una vez en este artículo se ha hecho un breve recorrido por la ingeniería inversa y el malware, realizando un seguimiento a su evolución, y la mutua correlación entre ellos, podemos ver que como a través de los años, la tecnología al igual que los atacantes y las amenazas, se aprovechan de las diferentes técnicas para acechar a sus víctimas, transformándose y adaptándose al medio.

Ahora aparece el concepto de ecosistema digital, en el cual tenemos involucradas redes físicas, lógicas, hardware físico local y en la nube, software de escritorio y de nube, servicios de nube, y muchos más componentes que conviven en ese mundo digital, en donde el eslabón más débil de la cadena de la seguridad de la información sigue siendo el ser humano. [20]

Mencionando lo que es la evolución de las amenazas, se puede hablar de la ingeniería inversa aplicada a la inseguridad de la información en aplicaciones móviles, ya cuando muchos pensaban que se habían acabado muchas amenazas que se veían en un Windows, Linux o Apple, ahora la amenaza también es para usuarios de dispositivos móviles, en sus diferentes sistemas operativos.

En este documento se hace referencia a android y a herramientas de decompilación como android tamer, en la cual un atacante puede descargar una aplicación, decompilarla, modificarla, agregar algo particular, como enviar información del usuario a un correo electrónico, un servidor, o causar daños en el rendimiento del equipo. [20]

A continuación se detalla el proceso para decompilar una aplicación Android [21] y volverla a insertar dentro de un celular directamente o en determinado caso como lo podría hacer un atacante instalando el apk en el celular sin que el usuario se entere que es una aplicación que ha sido modificada

**Proceso:**

**a. Paso 1.** Instalar la aplicación Android Tamer.

**b. Paso 2.** Conectar el móvil al pc y conectarlo a la máquina virtual.

**c. Paso 3.** Buscar la aplicación que se desea descargar y bajar el apk, o conseguir el apk previamente de cualquier otro lugar.

**d. Paso 4.** Con el apk objetivo en la máquina virtual, abrir APKTOOL [22] y se procede a decompilar el archivo apk con el comando

apktool d nombreapp.apk

**e. Paso 5.** Se extrae el archivo clases.dex [23] (clases de la app) con dex2jar [24].

d2j-dex2jar.sh nombreapp.apk

**f. Paso 6.** Con JD-GUI se abre el proyecto con todas sus clases, como cualquier proyecto java, y poderse editar. Un atacante podría modificar por ejemplo el archivo AndroidManifest.xml [25]. (contiene todos los permisos de la aplicación), podría insertar alguna acción o proceso para enviar información oculta a un servidor o a un correo electrónico, mientras el usuario no lo nota. [21]

**g. Paso 7.** Una vez se ha terminado se realiza el mismo proceso a la inversa, se vuelve a compilar el paquete, con las modificaciones que el pentester [26] o el atacante haya realizado, e insertándolo de nuevo en el celular por medio de sus técnicas de ingeniería social y observar su resultado. [21]

apktool b nombreapp.apk

## IV. Retos de la Seguridad de la Información

Los retos venideros de la seguridad informática son muy amplios, teniendo en cuenta el internet de las cosas y la domótica, la tendencia en torno hacia el control de todo de forma remota y forma móvil.

Si los atacantes logran infiltrar esos mecanismos domóticos del futuro, será como un virus en el centro de cómputo de una planta nuclear, con un impacto incalculable para la sociedad, la empresa y las personas del común. Todo el ecosistema digital quedará expuesto, tarjetas de crédito, aplicaciones, espacios en la nube, componentes de la casa como el acceso físico desde un móvil, entre otros riesgos.

Implementar estrategias para que el desarrollo ágil no se confunda con desarrollo inseguro y se apliquen los métodos y buenas prácticas necesarias para crear desarrollos cada vez más rápidos, pero seguros y confiables. (HoseDex2Jar).

## V. Conclusiones

La tecnología es evolutiva, así como lo hace el atacante, a medida que se obtienen mayores logros tecnológicos, también se abren nuevas brechas que pueden ser aprovechadas por los intrusos, autopistas de nuevas oportunidades para los atacantes, que están siendo aprovechadas, y posiblemente traerán muchos dolores de cabeza para los encargados de la seguridad de la información.

La evolución tecnológica especialmente será un reto para las empresas y personas que no evolucionen a la par de la tecnología, y permanezcan en ese letargo calcino que no les permita ver las amenazas informáticas que se avecinan, en caso de no tomar las medidas de seguridad necesarias.

El aseguramiento de las infraestructuras informáticas, debe iniciar desde la misma planificación de los proyectos, hasta la entrega y capacitación final con el cliente, y así evitar problemas irremediables desde la raiz.

## Agradecimientos

Agradecimientos del autor Marcos Fernando Reyes a Unisangil por el apoyo brindado para gestar estos logros.

## Referencias

[1] Noragueda, C. (2019). Las matemáticas de Alan Turing acabaron con Hitler. Retrieved from http://www.(URL)
[2] Historia de la computadora - Tecnología & Informática. (2019). Retrieved from http://www.(URL).
[3] Historia del Lenguaje c++ (2019). Retrieved from http://www.(URL).
[4] Savitch, W. (1994). Turbo Pascal. Redwood City, CA: Benjamin/Cummings.
[5] Cobol.¿Que es Cobol?. (2019). Retrieved from http://www.(URL).
[6] Fortran, el primer lenguaje de programación de alto nivel. (2019). Retrieved from http://www.(URL).
[7] Larevistainformatica.com. (2019). Lenguaje de Programación Ensamblador. [online] Available at: http://www.(URL).[Accessed 20 Apr. 2019].
[8] Martín, J., ®, C., Boy, S. and Smith, N. (2019). 23 virus de la época del DOS | Emezeta.COM. [online] Emezeta.com. Available at: http://www.(URL).[Accessed 20 Apr. 2019].
[9] Alonso, C. (2019). BlackBox: Ataques para Joomla, WordPress, Magento o PrestaShop. Retrieved from http://www.(URL).
[10] Alonso, C. (2019). PayPal: Di adiós a tu dinero si te confundes con el e-mail. Retrieved from http://www.(URL).
[11] Esteva (2019). Línea de tiempo -Historia de los virus-. [online] Es.slideshare.net. Available at: http://www.(URL).[Accessed 20 Apr. 2019].
[12] Diocesanos.es. (2019). Nostalgia: Así era la informática en los 80 y 90 | Todos hacemos TIC. [online] Available at: http://www.(URL).[Accessed 20 Apr. 2019].
[13] Diario Sur. (2019). Los cinco virus más raros de la era Internet. [online] Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[14] Testdevelocidad.es. (2019). El primer virus ransomware llegó en 1989 dentro de un disquete. [online] Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[15] Eoi.es. (2019). "Michelangelo" ¿el primer virus por venganza?. [online] Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[16] OpenMind. (2019). La historia de los virus informáticos. [online] Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[17] Ecured.cu. (2019). Ingeniería Inversa - EcuRed. [online] Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[18] Security, I. (2019). Cómo hacer ingeniería inversa de malware|análisis de malware. Retrieved from http://www.(URL).
[19] Academy, B. (2019). BacktrackAcademy: Realizando Ingeniería Inversa a Ransomware. Retrieved from http://www.(URL).
[20] Android Tamer. (2019). Retrieved from http://bit.ly/2IBLjlw
[21] RedInfoCol. (2019). Ingeniería inversa en aplicaciones de Android I - RedInfoCol. [online] Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[22] Academy, B. (2019). BacktrackAcademy: apktool - Una herramienta para reingeniería de archivos apk de Android. [online] Backtrack Academy. Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[23] Muñoz Veliz, O. (2019). BacktrackAcademy: dex2jar - Herramientas para trabajar con archivos .dex y java .class. [online] Backtrack Academy. Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[24] Linux, K. (2019). dex2jar. [online] Kali-linux.net. Available at: http://www.(URL).[Accessed 20 Apr. 2019].
[25] tuprogramacion.com. (2019). ¿Qué es el Android manifest?. [online] Available at: http://www.(URL). [Accessed 20 Apr. 2019].
[26] de la Informática, G. (2019). Android Tamer: Penetration Test en Android. [online] Blog.segu-info.com.ar. Available at: http://www.(URL). [Accessed 20 Apr. 2019].