#CyberSBC2024

Fundamentos de la Gestión de Incidentes (Malware)

Ricardo J. Rodríguez
Universidad de Zaragoza (España)

Cybersecurity Summer Bootcamp

LEÓN - 2024

8 al 18 julio de 2024

León, España

Organizado por:

GOBIERNO DE ESPAÑA · MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA · SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

OEA Más derechos para más gente

Con la colaboración de:

Canadá

AYUNTAMIENTO DE LEÓN

LEÓN CONGRESOS

universidad de León

$whoami

- **Ricardo J. Rodríguez**
  - PhD en Informática e Ingeniería de Sistemas
  - Profesor Titular en la Universidad de Zaragoza
  - **Investigador en ciberseguridad**, especialmente en:
    - Análisis de software
    - Forense digital (en particular, en memoria)
    - Seguridad de sistemas
  - **Grupo de investigación DisCo**
    - RME-DisCo: https://reversea.me
    - Síguenos en Twitter y Telegram! @reverseame
  - **E-mail**: rjrodriguez@unizar.es
    - Contáctame si tienes dudas después del taller
  - **Página web personal**: http://www.ricardojrodriguez.es

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

https://bit.ly/sbc24-analisis-malware

1. Introducción
2. Conceptos previos
3. Metodología de análisis de malware
4. Análisis estático
5. Análisis dinámico
6. Fases de ataque del malware
7. Casos de estudio
8. Conclusiones

Organizado por:
GOBIERNO DE ESPAÑA · MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA · SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD
OEA Más derechos para más gente

Con la colaboración de:
Canadá
AYUNTAMIENTO DE LEÓN
LEÓN CONGRESOS
universidad de León

1. Introducción

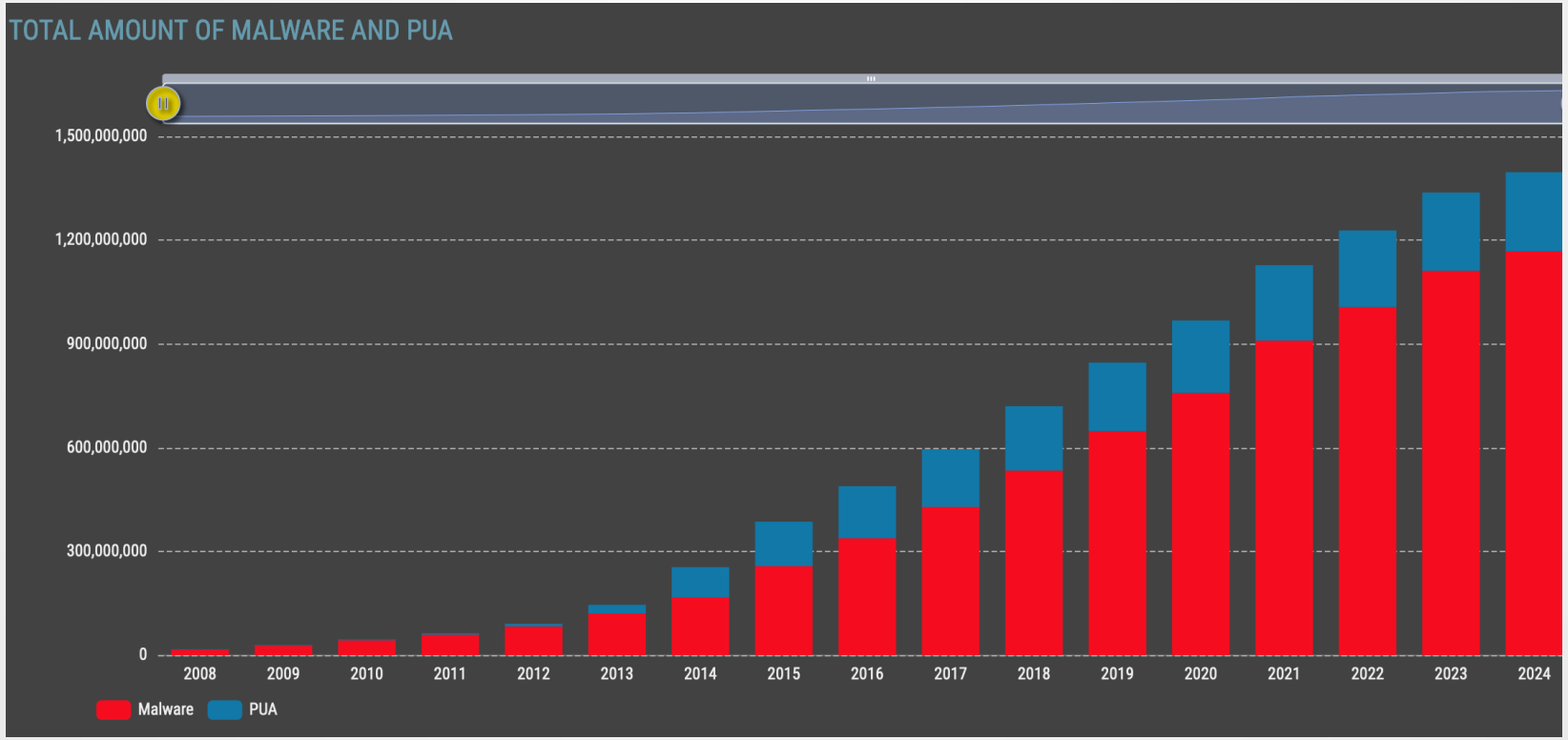
1. Introducció

¿Qué es *malware*?

- Malware: ***mal**icious soft**ware***
  - Código especialmente di
- Distintas características
- ¿Principal objetivo?
  - Fácil: **ganar dinero**
  - Algunos números:
    - ZeuS (+ de $100M recono
    - Citadel, Dridex: £20M en F

1. Introducción

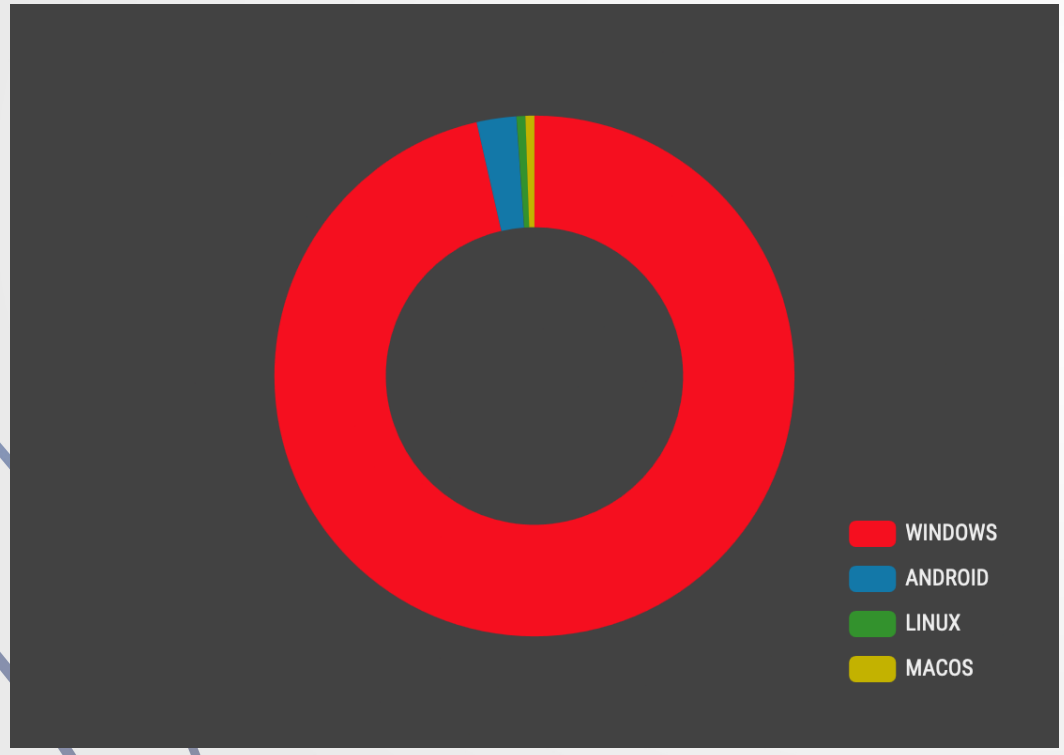
Algunas estadísticas

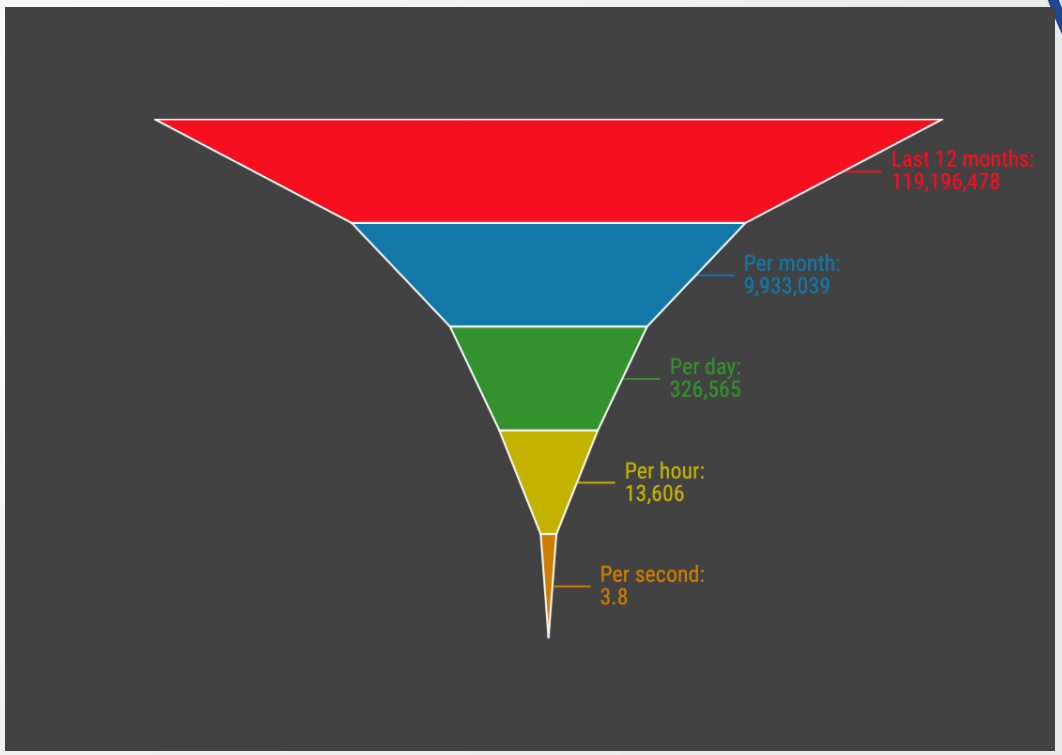


Fuente: https://portal.av-atlas.org/malware

1. Introducción

Algunas estadísticas



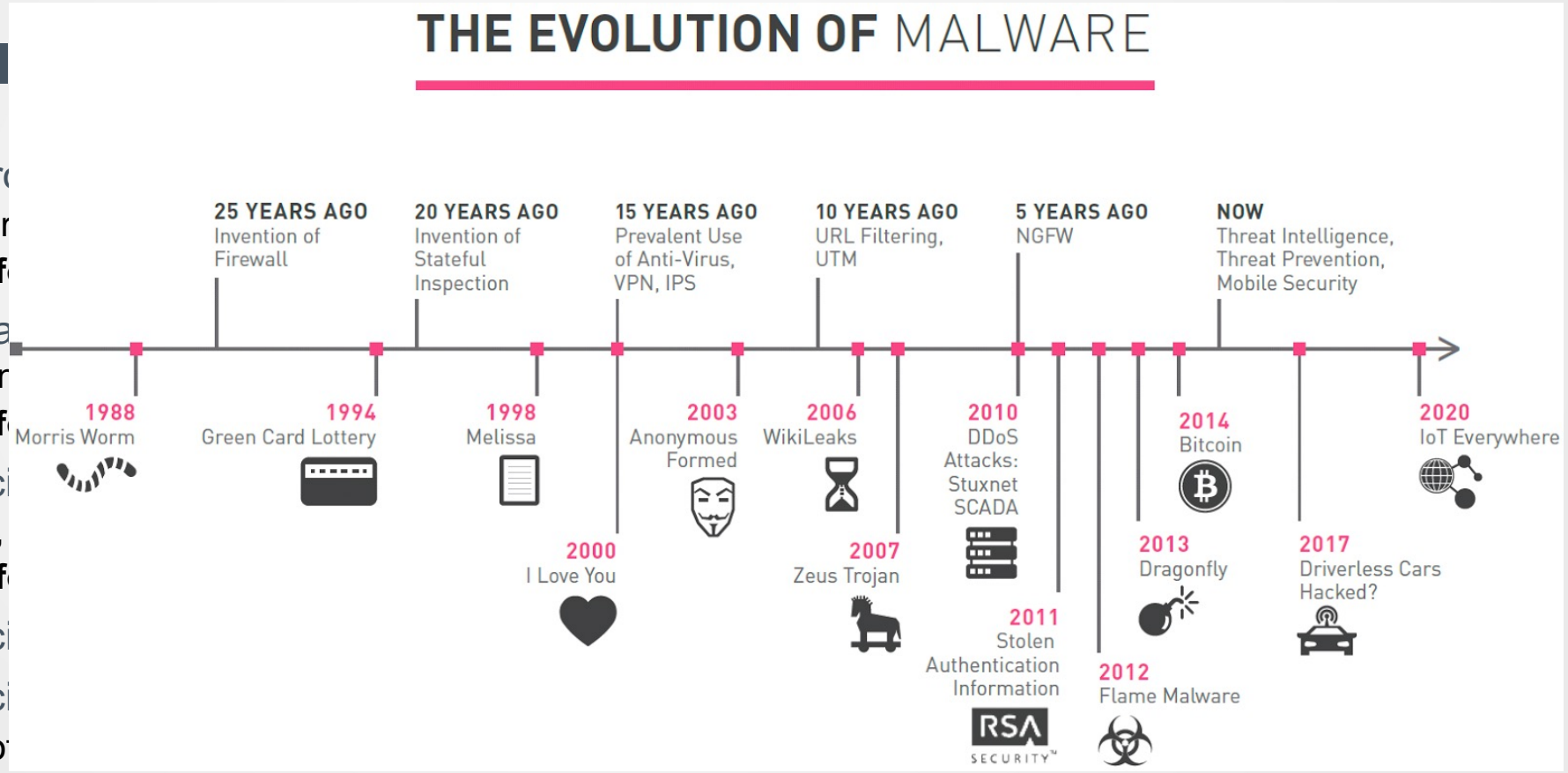


Fuente: https://portal.av-atlas.org/malware

1. Introducción

Historia del

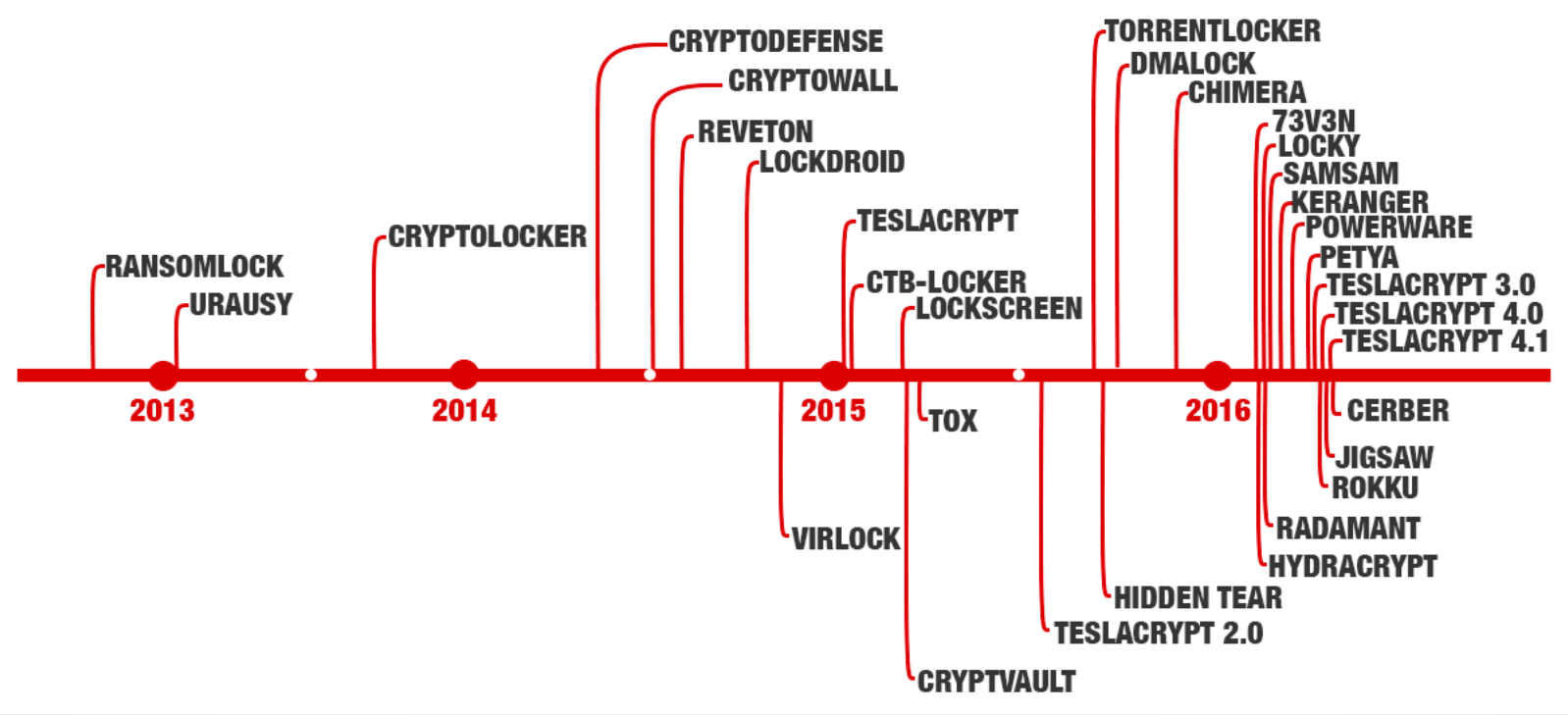
- 1970s: primer
  - Creeper
  - **Más inf**
- 1980s: prueba
  - Elk Clor
  - **Más inf**
- 1990s: evoluci
  - Vienna,
  - **Más inf**
- 2000s: evoluci
- 2010s: evoluci
  - Más so



Fuente: Check Point security Report, 2014

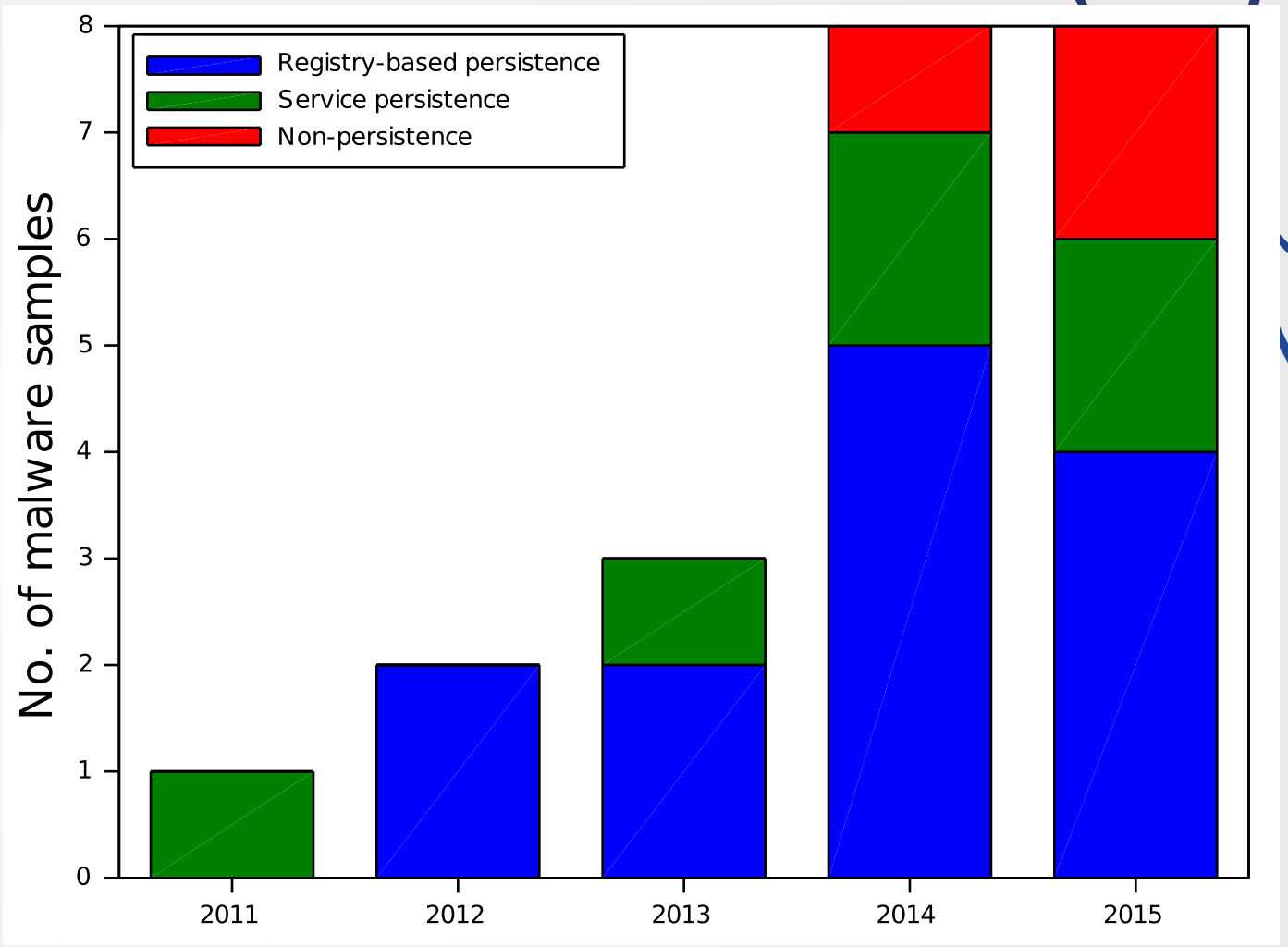
1. Introducción

Evolución del ransomware



1. Introducción

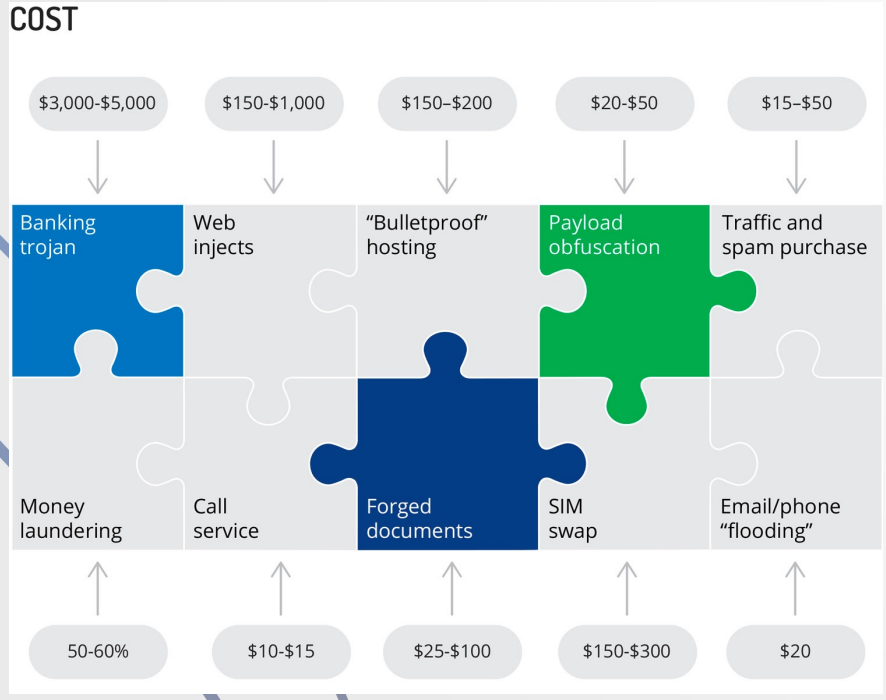
Otros tipos de malware

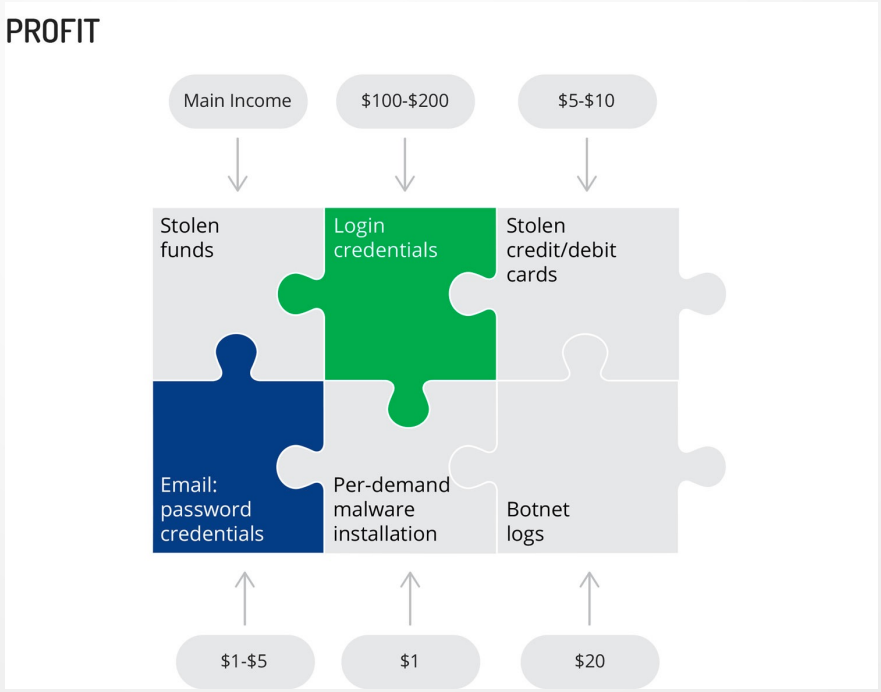


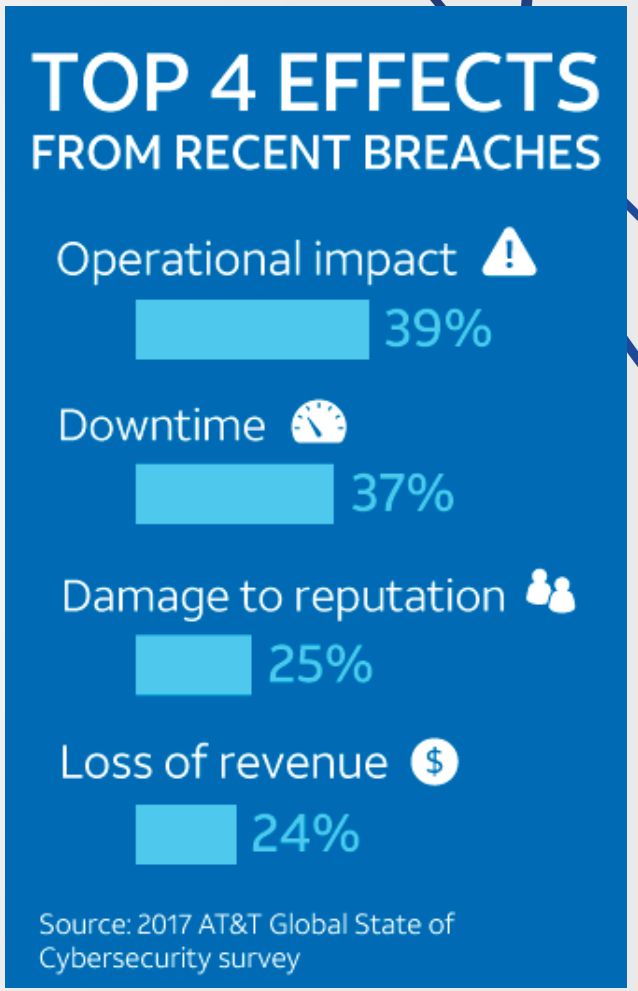
Fuente: Rodríguez, R. J. "Evolution and Characterization of Point-of-Sale RAM Scraping Malware". Journal in Computer Virology and Hacking Techniques, vol. 13, no. 3, pp. 179-192, 2017. DOI: 10.1007/s11416-016-0280-4

1. Introducción

Estimación de beneficios del cibercrimen



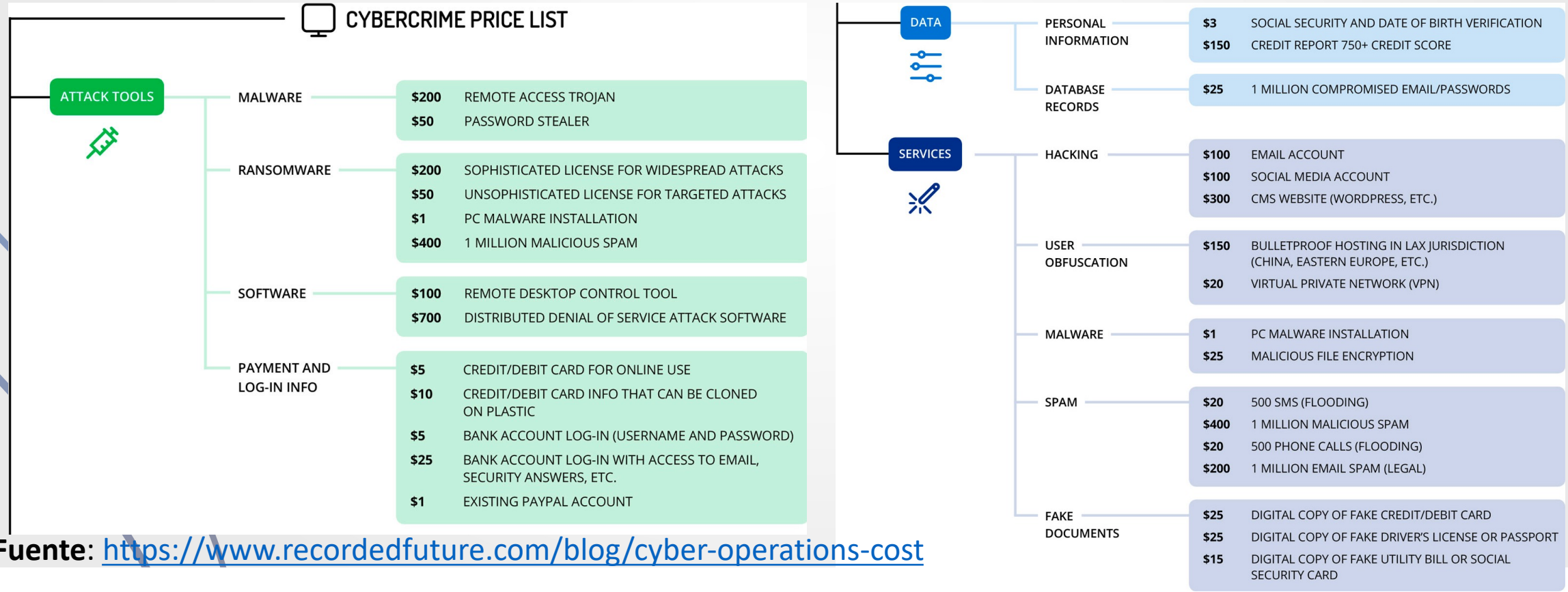




Fuente: https://www.recordedfuture.com/blog/cyber-operations-cost

1. Introducción

Estimación de beneficios del cibercrimen



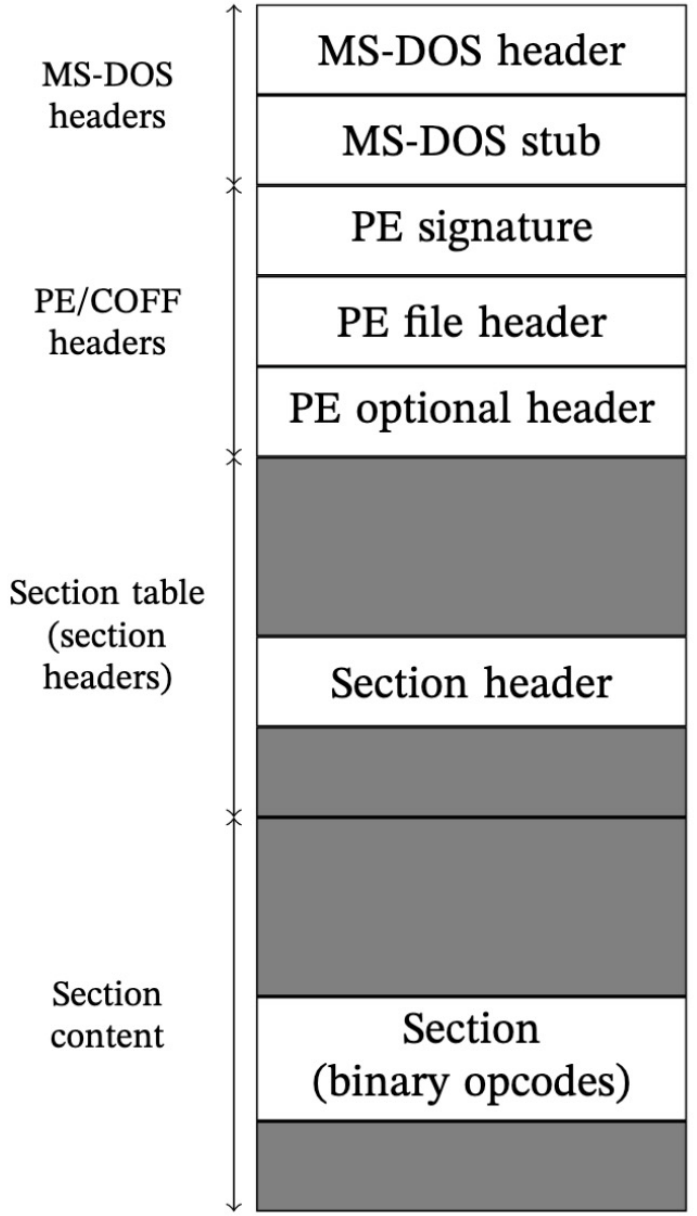
Fuente: https://www.recordedfuture.com/blog/cyber-operations-cost

2. Conceptos previos

2. Conceptos previos

Estructura de ficheros ejecutables en Windows

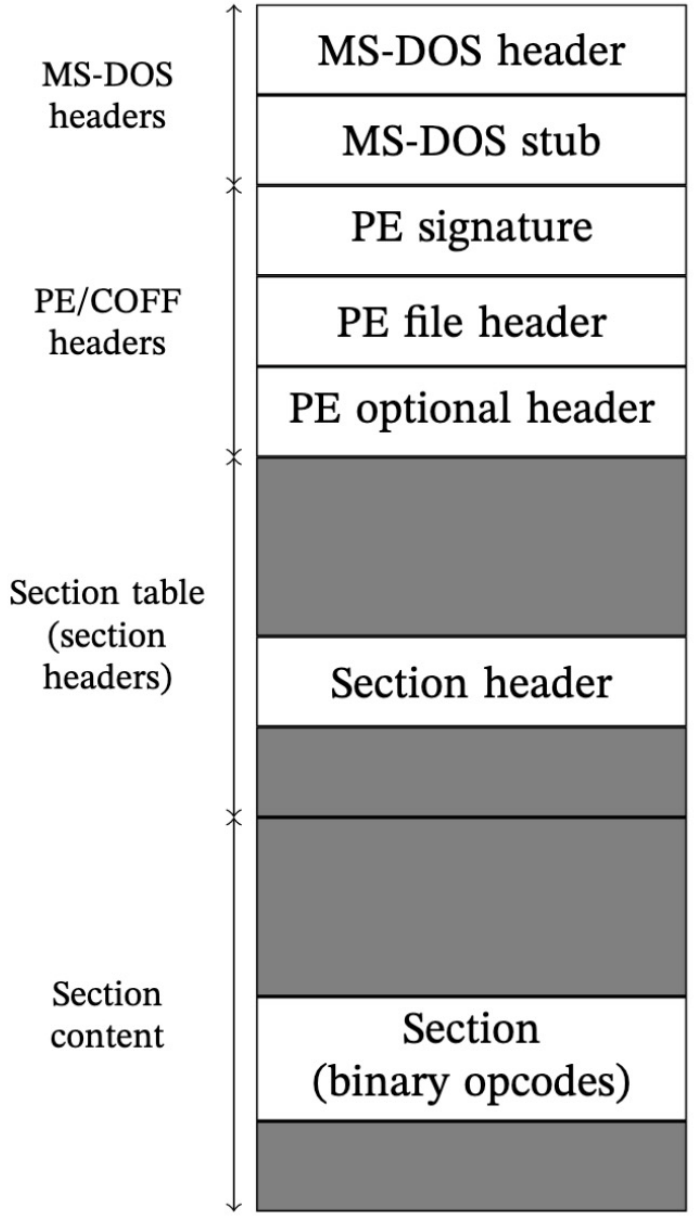
- Desde Windows NT 3.1
- **PE: Portable Executable**
  - Estructura de datos definida en WinNT.h (Microsoft Windows SDK)
  - Tres partes: cabeceras MS-DOS, cabeceras PE/COFF, cabeceras de secciones
- **Cabeceras de MS-DOS**
  - Primeros 64 bytes
  - e_magic: MZ (Mark Zbikowski)
  - e_lfanew: offset a cabeceras PE/COFF



2. Conceptos previos

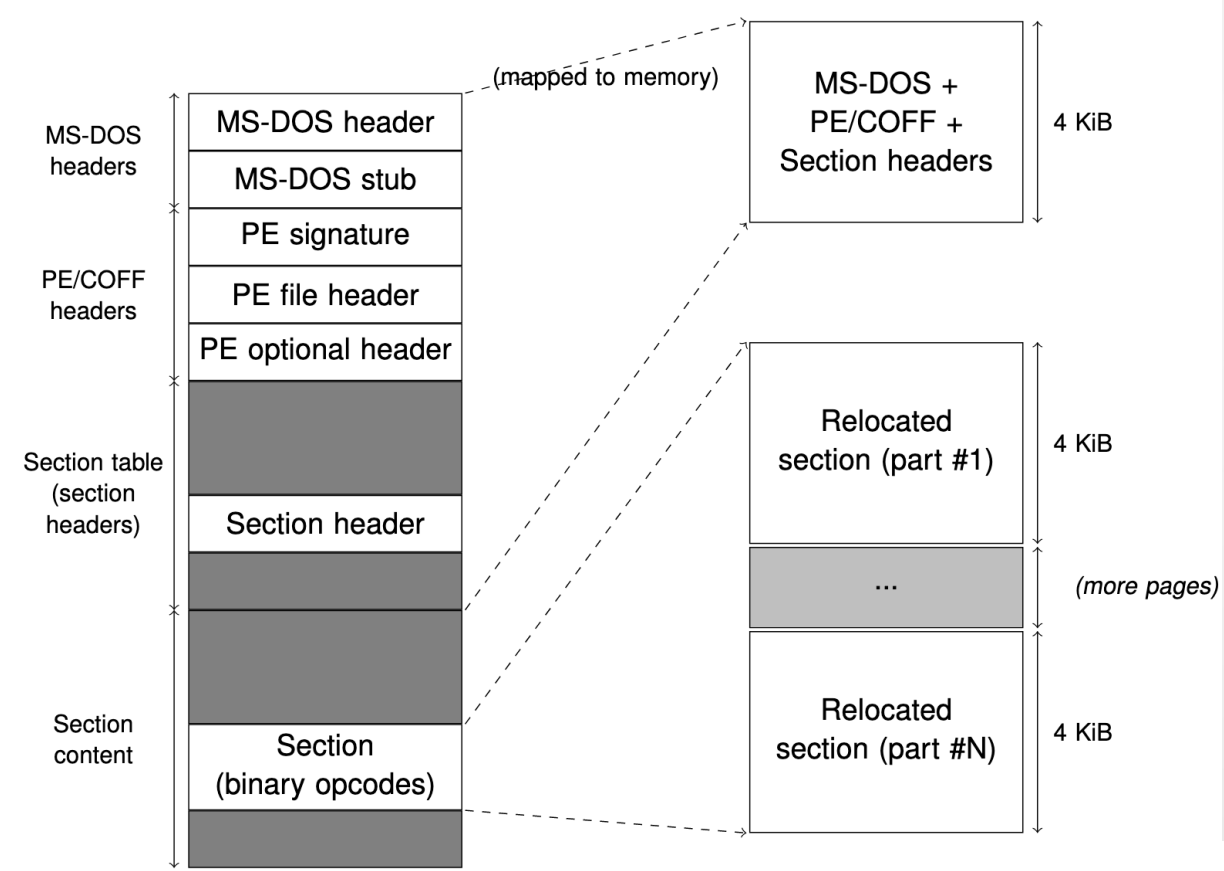
Estructura de ficheros ejecutables en Windows

- **Cabeceras PE/COFF**
  - **Signatura PE** (“PE\0\0”)
  - **Cabecera archivo PE**
    - Define máquina destino, número de secciones, características, etc.
  - **Cabecera PE opcional**
    - Opcional para algunos archivos objeto
    - Campos de interés: ImageBase, BaseOfCode, AddressOfEntryPoint
    - DataDirectory: tabla de directorios. Cada entrada tiene un significado
- **Cabeceras de sección**
  - Estructura IMAGE_SECTION_HEADER
  - Secciones comunes: .text/.code, .rdata/.rodata, .data, .reloc, …



2. Conceptos previos

Carga de ejecutables en memoria



3. Metodología de análisis de malware

3. Metodología de análisis de malware

Construcción de laboratorio de análisis de malware

- **¿Máquinas virtuales o máquinas físicas?**
  - Snapshots versus discos duros congelados
  - Los entornos virtuales suelen dejar muchos artefactos detectables: *cuidado*
- **Red de comunicación aislada (o filtrada)**
- **Herramientas de análisis de malware**
  - *Veremos más en las siguientes diapositivas…*
- **Herramientas de análisis automático:**
  - On-premise: CAPEv2
  - Cloud: VirusTotal, HybridAnalysis, JoeSandbox, etcétera
- **En la máquina anfitriona** (u otra máquina controlada)
  - Servidor DNS, web, firewall, etcétera

3. Metodología de análisis de malware

- **Tipos de análisis**
  - Análisis estático (*análisis de código muerto o código en frío – **NO** se ejecuta*)
  - Análisis dinámico (*análisis de código vivo o código en caliente – se ejecuta*)
  - Análisis híbrido
- **Normalmente se hará un análisis híbrido**
- **Objetivo fundamental:**
  - Conocer el impacto a las dimensiones de ciberseguridad del malware
  - *¿Qué hace? ¿Cómo lo hace? ¿Desde cuándo? ¿Cómo ha entrado?*
  - *¿Cómo lo quito? ¿Cómo lo evito? ¿A quién comunica?*

4. Análisis estático

4. Análisis estático

Identificación de la muestra: hashes (MD5, SHA-1, SHA-256…)

- **Permiten identificar de manera unívoca al ejecutable analizado**
  - Funciones denominadas one-way, con efecto avalancha (una modificación de un bit en la entrada cambia radicalmente la salida)
  - Cuidado con las funciones de hash débiles (e.g., MD5)
- **Herramientas útiles**: HashTab, md5sum, sha1sum, WinMD5Free, …
- **Objetivo**:
  - Calcular el hash y buscar información de la muestra en Internet
  - Comprobar el hash en VirusTotal o plataformas similares

4. Análisis estático

Recuperación de cadenas

- **Secuencia de ASCII bytes representables (e.g., ‘a’..’z’, ‘A’..’Z’, ‘0’..’9’, y demás símbolos) que acaban en un byte nulo** (*null-byte strings*)
- **Cadenas UNICODE:**
  - Usan UTF-8 para la codificación
  - Los caracteres ASCII se codifican como su codificación ASCII, seguidos de un byte nulo
- Cuando programamos, esas cadenas están **contenidas dentro de la aplicación**
- Resultan **útiles para extraer dominios, direcciones IP, criptocarteras, etcétera**…
  - Uso de expresiones regulares (XXX)
- **Herramienta de interés**: strings
  - Nativa de UNIX
  - Portada a Windows: https://technet.microsoft.com/en-us/sysinternals/bb89743
- **OJO: el malware puede contener las cadenas codificadas o cifradas**
  - Por tanto, no se visualizarán con esta técnica

4. Análisis estático

Visualización del fichero PE: secciones y cabeceras

- **Comprobar las secciones del ejecutable**
  - Si no se encuentran secciones habituales, puede ser un indicativo de que está ofuscado
- **Útil para identificar la herramienta de ofuscación utilizada**
- **Comprobar los valores de algunos campos de interés:**
  - TLS
  - AddressOfEntryPoint
  - ImageBase
- **Herramientas de interés:**
  - PEiD
  - PE Bear
  - CFF Explorer
  - PE View
  - PE Explorer

4. Análisis estático

Visualización del fichero PE: funciones importadas

- **Comprobar la Import Address Table: APIs importadas**
  - Contiene las funciones (junto con las DLLs que las contienen) presentes en el código compilado
  - *Ojo: que existan, no quiere decir que se ejecuten…*
- **Útil para identificar el posible comportamiento de la aplicación**
  - Las funciones es la forma que tenemos en Windows de interaccionar con los recursos del sistema
- **Pueden aparecer en plano, o no aparecer:**
  - Carga en ejecución en vez de en arranque
  - Distintas formas de ofuscación
    - Presencia de LoadLibraryA/W + GetProcAddress suele indicar carga dinámica
    - Navegación por las DLLs/APIs a través de estructuras internas del SO Windows
- **Herramientas de interés:**
  - Dependency Walker
  - PE Bear
  - CFF Explorer
  - …

4. Análisis estático

Visualización del fichero PE: sección de recursos

- **Comprobar la sección de recursos (si existe)**
  - Contiene datos de sólo lectura contenidos dentro de la aplicación
  - *Ojo: que existan, no quiere decir que se ejecuten…*
- **Suele contiene recursos (e.g., imágenes, cursores, etc.), aunque puede contener otras aplicaciones**
  - Primeros bytes de un PE
  - Sección “gorda”
  - Bytes codificados
- Si se encuentra algo binario, hay que **extraerlo y analizarlo aparte**
- **Herramientas de interés:**
  - ResourceHacker
  - ResEdit

4. Análisis estático

Resumen de funciones típicamente usadas por malware

- **Versiones de funciones en Windows:**
  - ***A / *W:** parámetros cadena ASCII / parámetros cadena UNICODE
  - ***Ex**: versión de función extendida (admite más parámetros)
- **Procesos e IPCs (kernel32.dll)**
  - CreateProcessA, OpenProcess, CreateThread, CreatePipe, CreateNamedPipe, CreateMutex, OpenMutex, CreateToolhelp32Snapshot, CreateRemoteThread, …
- **Archivos (kernel32.dll)**
  - CreateFile, WriteFile, ReadFile, CopyFile, …
- **Registro de Windows (advap32i.dll)**
  - RegOpenKey, RegEnumKey, RegEnumValue, RegDeleteKey, RegQueryInfoKey, …
- **Comunicaciones de red (ws2_32.dll – Winsocks, wininet.dll)**
  - WSAStartup, WSASocket, socket, connect, accept, bind, recv, send, htons, …
  - **urlmon.dll:** URLDownloadToFile

4. Análisis estático

Problemas

- **Indicios, no certezas**
- **Posible falta de resultados**
- **Incertidumbre**

Primer contacto para saber qué puede estar haciendo
(pero no lo sabremos seguro sin un análisis más exhaustivo)

5. Análisis dinámico

5. Análisis dinámico

Interacción con el Sistema Operativo

- **Sistema de ficheros**
  - *¿Qué ficheros está creando o accediendo?*
  - *¿Qué ficheros está modificando o eliminando?*
  - **Herramientas de interés:**
    - DiskMon: registra cualquier actividad con el disco duro de sistema
    - Handle: manejadores abiertos por los procesos (similar a lsof de UNIX)
    - Process Monitor: monitoriza ciertas funciones de la API de Windows
    - WinAPIOverride
    - Microsoft Detours
- **Procesos**
  - *¿Qué procesos está creando o accediendo?*
  - *¿Qué procesos está eliminando?*
  - **Herramientas de interés:**
    - Process Hacker
    - Sysmon: registra todos los eventos de Windows

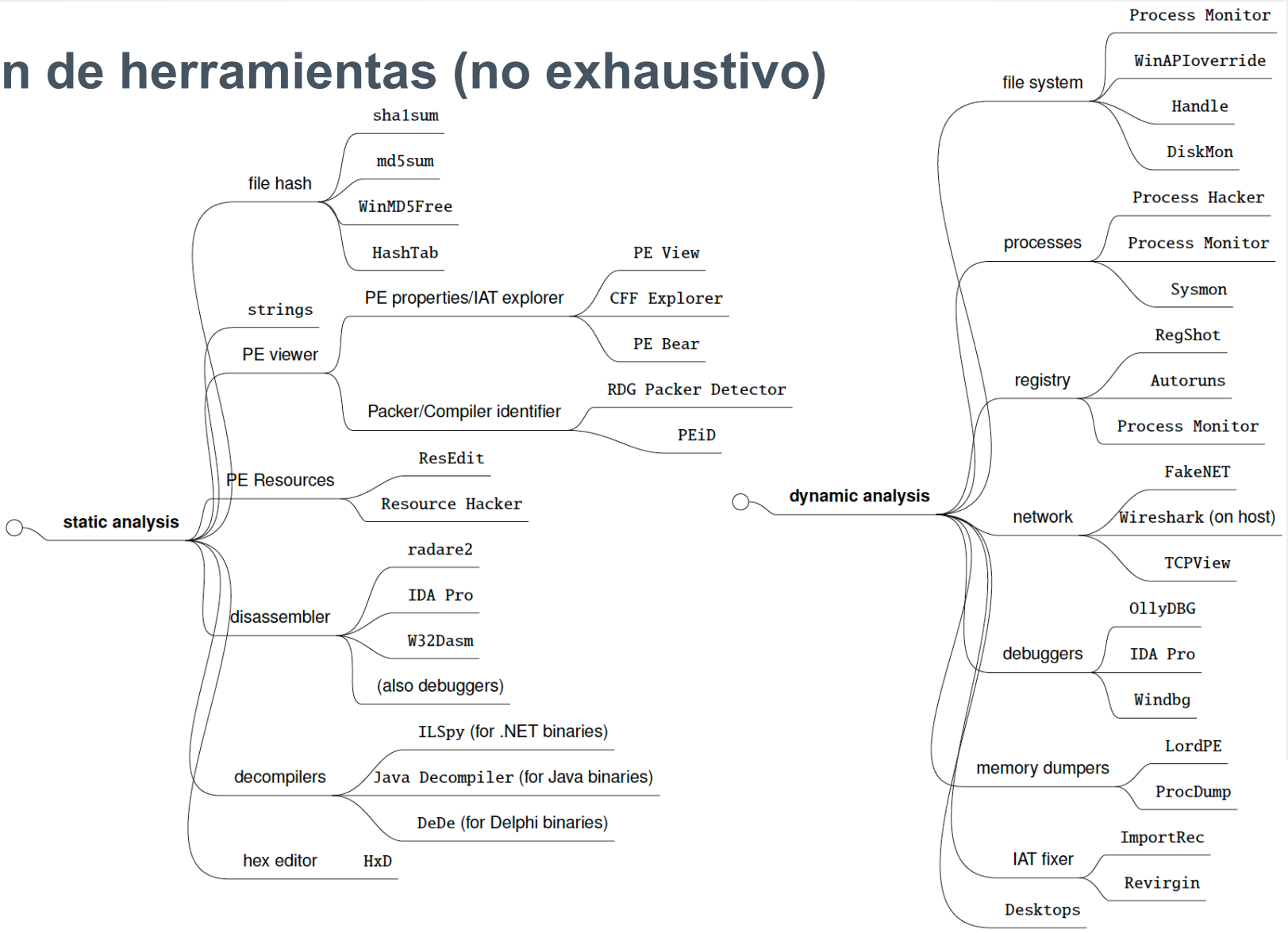
5. Análisis dinámico

- **Interacción con el SO: Registro de Windows**
  - *¿Qué claves de Registro está creando o accediendo?*
  - *¿Qué claves de Registro está modificando o eliminando?*
  - **Herramientas de interés:**
    - Autoruns (de Sysinternals): comprueba los puntos de auto-arranque del sistema (ASEPs)
    - RegShot: captura estado del Registro en un momento dado y permite hacer análisis comparativos

- **Interacción con el exterior: análisis de comunicaciones**
  - *¿Qué dirección DNS intenta resolver? ¿A qué dirección IP conecta?*
  - *¿Dónde está esa IP/dominio? ¿Qué se puede saber sobre ella?*
  - *¿Qué datos se están enviando o recibiendo?*
  - **Herramientas de interés:**
    - Wireshark
    - FakeNET (o servicios similares, para simular servidores)

5. Análisis dinámico

Resumen de herramientas (no exhaustivo)



6. Fases de ataque del malware

6. Fases de ataque del malware

1. Ganando acceso al sistema

- **Interacción con el usuario**
  - Ingeniería social
  - Correos electrónicos (spam)
  - Software descargado de Internet
- **Vulnerabilidades en el software**
  - Aplicaciones no actualizadas
  - Sistema operativo no actualizado
  - Páginas web de terceros comprometidas
  - Servidores internos comprometidos
    - App provisioning
    - Update provisioning

6. Fases de ataque del malware

2. Persistencia (mediante ASEPs)

Windows Auto-Start Extensibility Points	Characteristics: Write permissions	Execution privileges	Tracked down in memory forensics†	Freshness of system	Execution scope	Configuration scope
System persistence mechanisms						
Run keys (HKLM root key)	yes	user	yes	user session	application	system
Run keys (HKCU root key)	no	user	yes	user session	application	user
Startup folder (%ALLUSERSPROFILE%)	yes	user	no	user session	application	system
Startup folder (%APPDATA%)	no	user	no	user session	application	user
Scheduled tasks	yes	any	no	not needed‡	application	system
Services	yes	system	yes	not needed‡	application	system
Program loader abuse						
Image File Execution Options	yes	user	yes	not needed	application	system
Extension hijacking (HKLM root key)	yes	user	yes	not needed	application	system
Extension hijacking (HKCU root key)	no	user	yes	not needed	application	user
Shortcut manipulation	no	user	no	not needed	application	user
COM hijacking (HKLM root key)	yes	any	yes	not needed	system	system
COM hijacking (HKCU root key)	no	user	yes	not needed	system	user
Shim databases	yes	any	yes	not needed	application	system
Application abuse						
Trojanized system binaries	yes	any	no	not needed	system	system
Office add-ins	yes	user	yes	not needed	application	user
Browser helper objects	yes	user	yes	not needed	application	system
System behavior abuse						
Winlogon	yes	user	yes	user session	application	system
DLL hijacking	yes	any	no	not needed	system	system
AppInit DLLs	yes	any	yes	not needed	system	system
Active setup (HKML root key)	yes	user	yes	user session	application	system
Active setup (HKCU root key)	no	user	yes	user session	application	application

†*If the memory is paging to disk, it would be not possible to track down these ASEPs in memory forensics.*
‡*Depends on the trigger conditions defined to launch the program.*

Más información: Uroz, D. & Rodríguez, R. J. "*Characteristics and Detectability of Windows Auto-Start Extensibility Points in Memory Forensics*". Digital Investigation, 2019, 28, S95-S104, https://doi.org/10.1016/j.diin.2019.01.026

6. Fases de ataque del malware

2. Persistencia (mediante ASEPs)

- **Ejemplos de Runkeys típicas de malware:**
  - HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
  - HKLM\Software\Microsoft\Windows\CurrentVersion\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

6. Fases de ataque del malware

3. Comportamiento dañino

- **Distintas formas de importer funciones:**
  - **Importación estática** *(funciones presentes en la aplicación)*
    - Visibles en la parte de análisis estático (tabla IAT)
  - **Importación dinámica** *(funciones no presentes en la aplicación)*
    - Las funciones de Windows se resuelven en tiempo de ejecución
    - Distintas formas de importar de manera dinámica una función
    - Normalmente:
      - LoadLibrary (carga una biblioteca DLL y devuelve su dirección base)
      - GetProcAddress (devuelve la dirección de la función solicitada)
    - Las funciones se pueden resolver por nombre o por ordinal
    - *Existe otra forma más compleja de resolver funciones (usada habitualmente en explotación)*

6. Fases de ataque del malware

3. Comportamiento dañino

- **Downloaders**
  - Primera etapa de una infección exitosa
  - Descarga otro malware e instala ASEP para persistencia
- **Robo de información**
  - APIs relacionadas con ficheros (búsqueda de ficheros con cierta extension o nombre)
- **Exploradores de memoria de otros procesos**
  - Buscan información de interés en otros procesos y la extraen (a disco o red)
- **Ransomware**
  - Iteración en directorios y ficheros (apertura, lectura, escritura)

6. Fases de ataque del malware

3. Comportamiento dañino

- **Keyloggers**
  - Colocan una función de hook, que puede ser global (de sistema) o específica a un hilo
  - *Recuerda que Windows es un sistema operativo dirigido por eventos…*
  - Eventos típicos filtrados: WH_CALLWNDPROC, WH_CBT, WH_DEBUG, WH_GETMESSAGE, WH_KEYBOARD, WH_MOUSE, WH_MSGFILTER
- **Inyectores de Código**
  - Acceden a la memoria de otro proceso y ejecutan el código dañino desde allí
  - 3 métodos para hacerlo: carga de DLL remota, función hook, escritura de código binario
- **Conexión con servidor de mando y control (C&C)**
  - Winsocks (≈ psockets)
  - Wininet: abstrae sesiones HTTP y FTP

7. Casos de estudio

7. Casos de estudio

https://bit.ly/sbc24-analisis-malware

- **Descarga muestras disponibles de carpeta “muestras”**
- **Sigue las instrucciones detalladas en el PDF del taller**
- **Intenta responder a las siguientes preguntas:**
  - *¿Qué hace el malware en el sistema?*
  - *¿Cómo lo clasificarías?*
- **Realiza primero un análisis estático**
- **Realiza luego un análisis dinámico**

8. Conclusiones

8. Conclusiones

- **Malware: malicious software**
- Problema creciente y cada vez más sofisticado
- Análisis de malware: *determinar qué (y cómo) está afectando a las dimensiones de ciber*
- **Tipos de análisis**
  - Estático: analizar las propiedades y características del ejecutable
  - Dinámico: observar la interacción con el Sistema Operativo y con el exterior
- **¿Esto es todo?**
  - **No. Análisis estático y dinámicos avanzados**
    - Uso de técnicas de análisis de software: *CFG, análisis taint, ejecución simbólica, depuración, etcétera*
- ***¿Y qué pasa en otros Sistemas Operativos?***
  - Cambian las herramientas de análisis, pero el objetivo y la metodología son las mismas

Cybersecurity Summer Bootcamp

LEÓN - 2024

8 al 18 julio de 2024

León, España

#CyberSBC2024

incibe.es/eventos/summer-bootcamp

Organizado por:

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

OEA Más derechos para más gente

Con la colaboración de:

Canadá

AYUNTAMIENTO DE LEÓN

LEÓN CONGRESOS

universidad de León