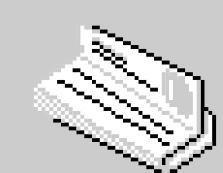
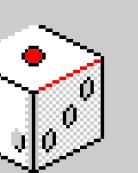
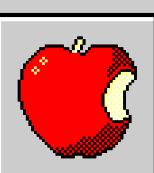


Ingeniería Inversa



Pineda Galindo Ricardo Angel
Asignatura: Sistemas Operativos
Profesor: Ing. Gunnar Eyal Wolf Iszaevich



11:11PM

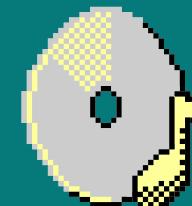
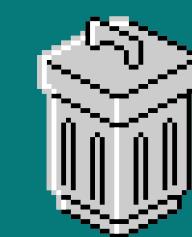
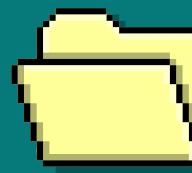
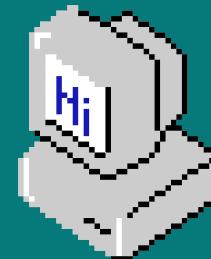
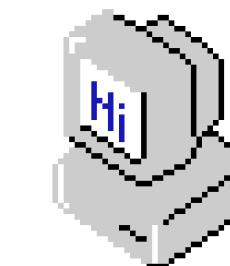
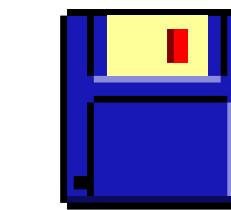


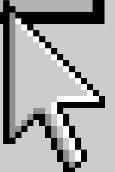
TABLA DE CONTENIDOS



Definición



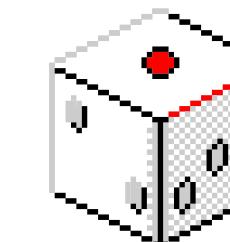
Tipos de Análisis



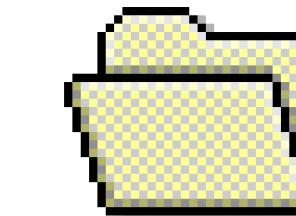
Start



Aplicaciones



Técnicas de
ofuscación



Herramientas



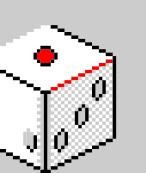
DEFINICIÓN



El enfoque es muy distinto en todas las áreas donde se utiliza



Es el proceso llevado a cabo con el objetivo de obtener información o un diseño a partir de un producto u objeto, con el fin de determinar cuáles son sus componentes, de qué manera interactúan entre sí y cuál fue el proceso de fabricación.



[Back to Agenda Page](#)



Ciberseguridad

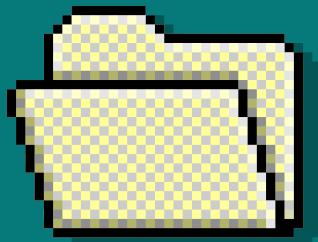


[Back to Agenda Page](#)

```
[0x00000000]> pd
    0x00000000    90        nop
    0x00000001    90        nop
    0x00000002    6800009c00 push 0x9c0000 ; 0x009c00
    0x00000007    e8c7ace37b call 0x7be3acd3
        0x7be3acd3(unk)
    0x0000000c    bb04009c00 mov ebx, 0x9c0004
    0x00000011    8903      mov [ebx], eax
    0x00000013    e81903f47b call 0x7bf40331
        0x7bf40331()
    0x00000018    bb08009c00 mov ebx, 0x9c0008
    0x0000001d    8903      mov [ebx], eax
    0x0000001f    bb00009c00 mov ebx, 0x9c0000
    0x00000024    c60300    mov byte [ebx], 0x0
-> 0x00000027    68e8030000 push 0x3e8 ; 0x0000003e8
    0x0000002c    e81124e37b call 0x7be32442
        0x7be32442(unk)
=< 0x00000031    ebf4      jmp 0x100000027
    0x00000033    90        nop
    0x00000034    ff        invalid
    0x00000035    ff        invalid
    0x00000036    ff        invalid
    0x00000037    ff        invalid
```



Tipos de análisis



Análisis Estático

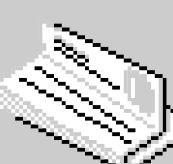
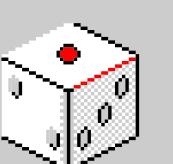
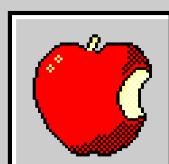
Consiste en analizar el código en ensamblador sin ejecutar el binario asociado. No existe riesgo de infección puesto que no es necesario ejecutarlo.

Análisis Dinámico

Consiste en estudiar las acciones del binario durante su ejecución, cabe destacar que cuando se ejecuta el binario, la máquina donde se ejecuta se verá infectada.

101
011

BIN



[Back to Agenda Page](#)

Técnicas de Ofuscación

- Ofuscar las cadenas de caracteres

Malware HerpesNet

```
rootbsd@lab:~/strings herpesnet.exe
[...]
tcerfhygy
uggc://qq.mrebkpbqr.arg/urecarg/
7497806rpp6p19836n17n3p2pq084000
uggc://jjj.mrebkpbqr.arg/urecarg/
sgc.mrebkpbqr.arg
uggc://sex7.zvar.ah/urecarg/
hcybnq@mrebkpbqr.arg
hccvg
ujsdedbngfgjhhuugfgfujd
rffggghooo
Ashfurncsmx
[...]
```

```
rootbsd@lab:~/strings herpesnet.exe | rot13
[...]
gpresultl
http://dd.zeroxcode.net/herpnet/
74978b6ecc6c19836a17a3c2cd0840b0
http://www.zeroxcode.net/herpnet/
ftp.zeroxcode.net
http://frk7.mine.nu/herpnet/
upload@zeroxcode.net
uppit
hwfqfqooatstwuuhtstshwq
essttubbb
Nfusheapfzk
[...]
```

Técnicas de Ofuscación

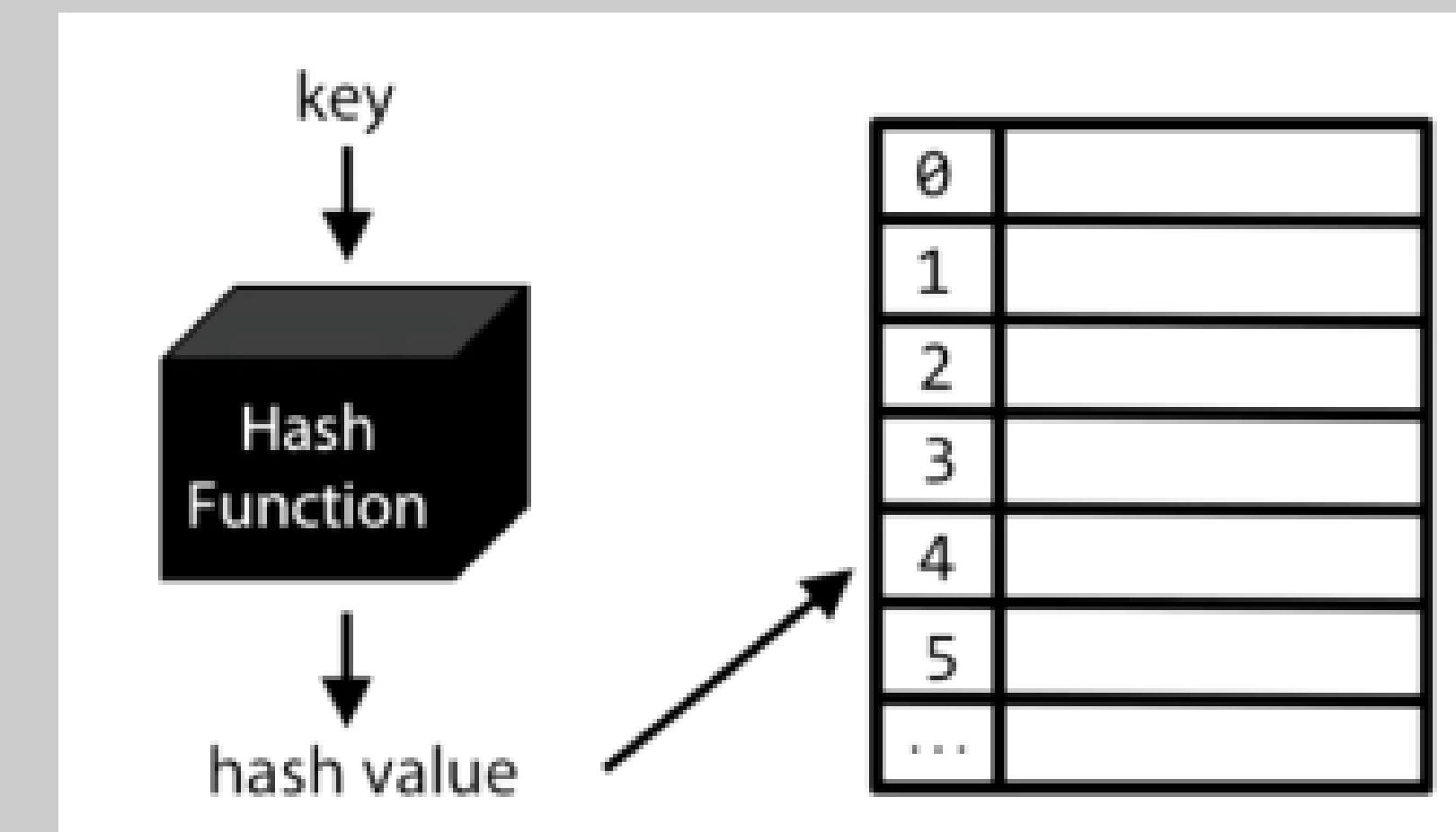
- API de Windows

Malware Duqu

```
00401E34      push    eax
00401E35      push    5FC5AD65h
00401E3A      push    [ebp+viewonNTDLL]
00401E3D      push    [ebp+NTDLLmodhdl]
00401E40      push    [ebp+imports]
00401E43      call    FUN_4017E2
```

Argumentos importantes

- 5FC5AD65
- ebp+viewonNTDLL



Técnicas de Ofuscación

- Packers

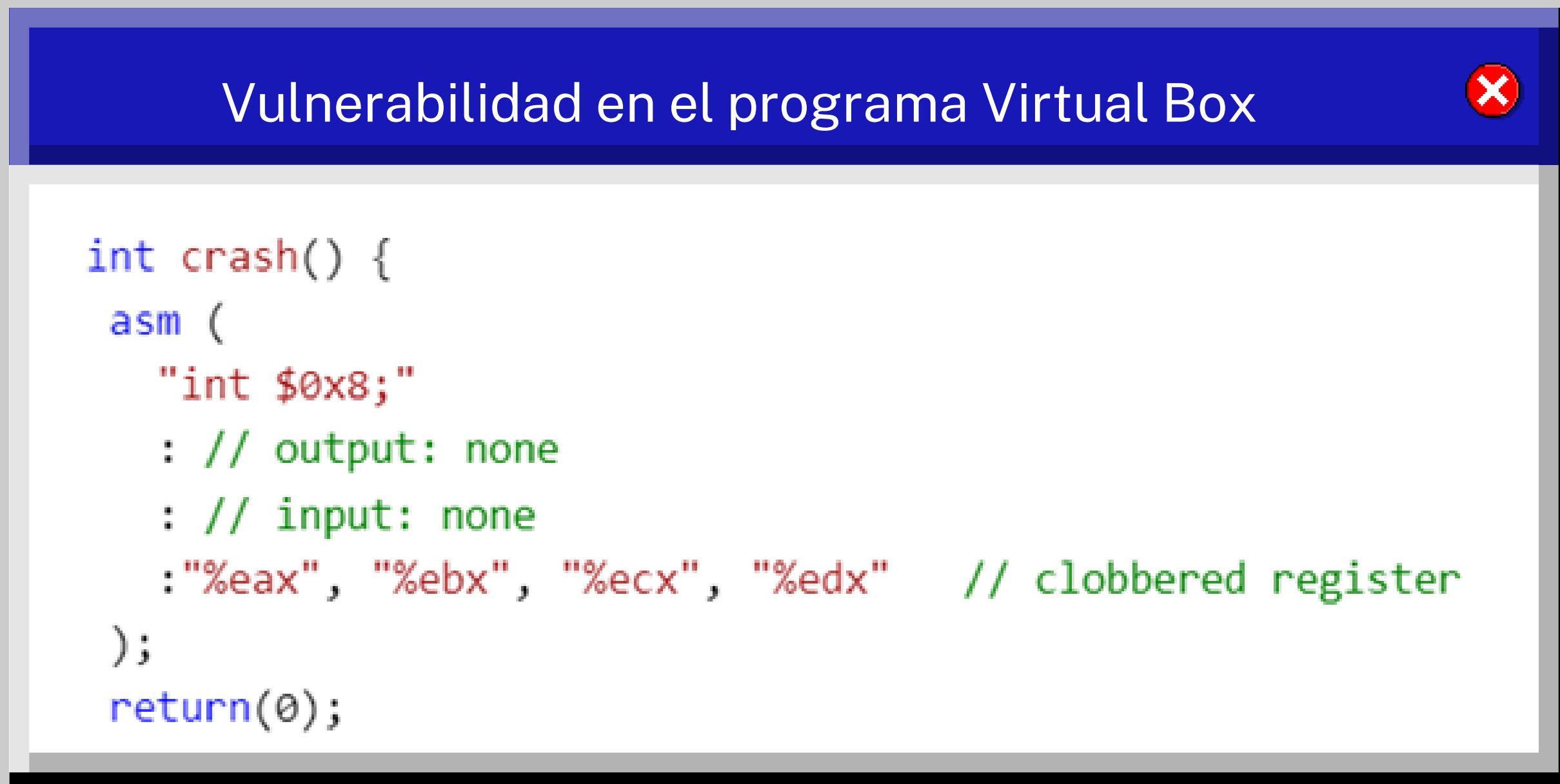
Para identificar packers se puede utilizar una herramienta llamada YARA, la cual contiene firmas de packers

```
rootbsd@lab:~$ yara yara/packer.yara code_upx.exe
UPXv20MarkusLaszloReiser code_upx.exe
UPXV200V290MarkusOberhumerLaszloMolnarJohnReiser code_upx.exe
UPX20030XMarkusOberhumerLaszloMolnarJohnReiser code_upx.exe
UPX290LZMAMarkusOberhumerLaszloMolnarJohnReiser code_upx.exe
```

En este caso, el binario analizado coincidió con 4 firmas upx

Técnicas de Ofuscación

- Anti VM



Vulnerabilidad en el programa Virtual Box

```
int crash() {
    asm (
        "int $0x8;"           // instruction
        : // output: none
        : // input: none
        : "%eax", "%ebx", "%ecx", "%edx" // clobbered register
    );
    return(0);
}
```

The screenshot shows a debugger window titled "Vulnerabilidad en el programa Virtual Box". The assembly code within the window is as follows:

```
int crash() {
    asm (
        "int $0x8;"           // instruction
        : // output: none
        : // input: none
        : "%eax", "%ebx", "%ecx", "%edx" // clobbered register
    );
    return(0);
}
```

Técnicas de Ofuscación

- Anti VM

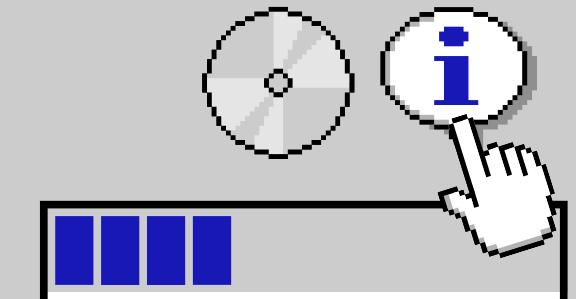
Vulnerabilidad en el programa Virtual Box 

Unspecified vulnerability in the Oracle VM Virtual Box component in Oracle Virtualization 3.2, 4.0, and 4.1 allows local users to affect availability via unknown vectors related to VirtualBox Core. NOTE: The previous information was obtained from the October 2012 CPU. Oracle has not commented on claims from another vendor that this issue is related to "incorrect interrupt handling."

(CVE - CVE-2012-3221, 2024)



Herramientas

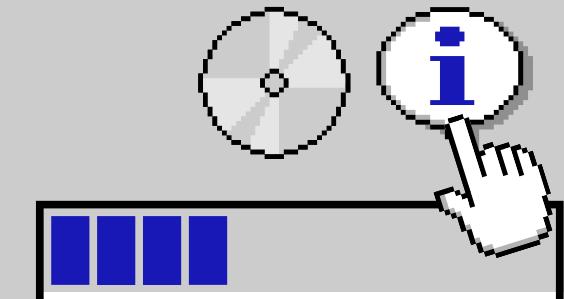


- Desensamblador: Convierte el código de lenguaje máquina a lenguaje ensamblador para que pueda ser más legible para las personas.
- Decompilador o compilador inverso: Recrea un código en lenguaje de alto nivel a partir del lenguaje máquina binario.
- Depurador: Es un programa que permite el control de otro programa, dando opción a analizarlo paso a paso para poder ir viendo el estado de las variables, del uso de memoria y afectación del programa en el equipo.
- Editor Hexadecimal: Es un programa que permite editar archivos binarios.
- Detector de packers: Este tipo de herramienta muestra si el programa está protegido por un packer y que packer es .
- Sandbox: Es un entorno controlado y aislado en el cual se ejecuta un programa para poder estudiar su comportamiento.

[Back to Agenda Page](#)

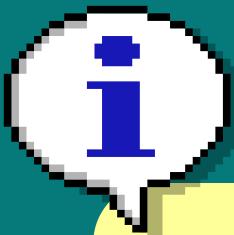
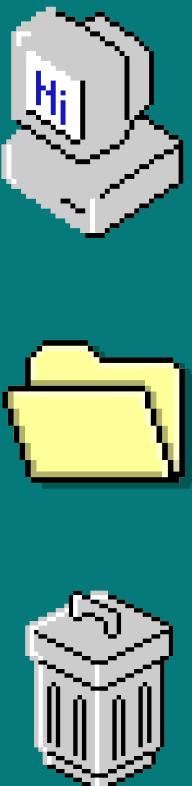


Herramientas (Ejemplos)

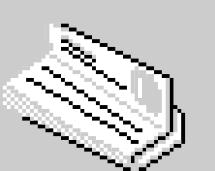
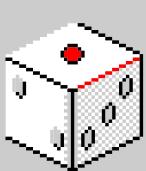
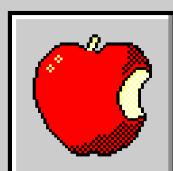


- Ghidra: Es una herramienta de **análisis estático** hecha pública por la NSA (Agencia Nacional de Seguridad) a inicios del 2019. Puede **desensamblar** archivos binarios y contiene un **decompilador** que permite obtener un pseudocódigo C del binario que se está analizando.
- Immunity Debugger: Es un **depurador** de 32 bits para Windows que permite ver las instrucciones ejecutadas, el estado de la memoria y de los registros. La ventaja que tiene es que permite el lenguaje de Python para automatizar ciertas tareas.
- IDA: Es una herramienta muy completa que permite realizar tanto análisis estático como dinámico, también tiene desensamblador, decompilador, debugger, capacidad para renombrar variables, crear scripts para automatizar procesos, etc.
- x64dbg: Es un **debugger** que soporta arquitecturas de 64 y 32 bits.
- HxD: Es un **editor hexadecimal**

[Back to Agenda Page](#)

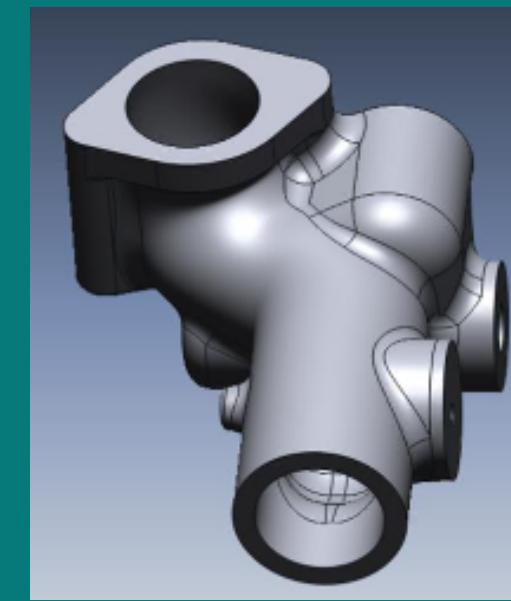
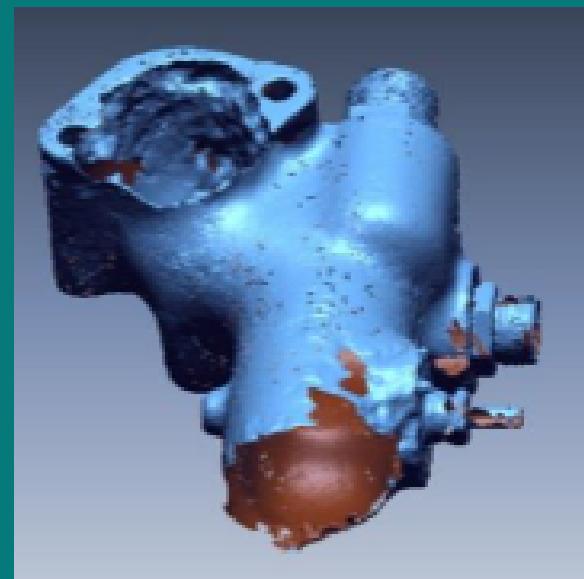


IMPORTANTE

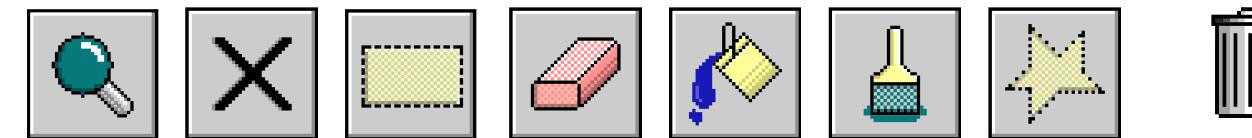


[Back to Agenda Page](#)

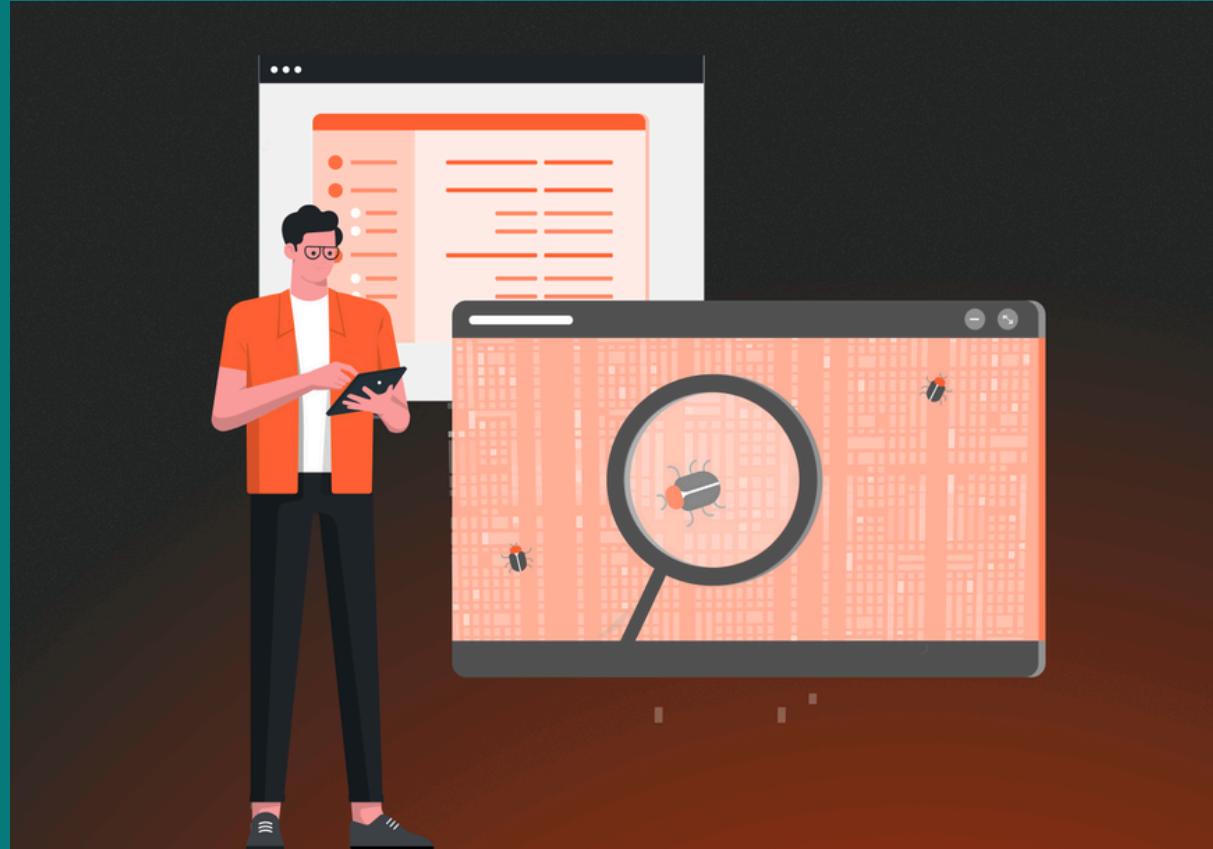
En la industria



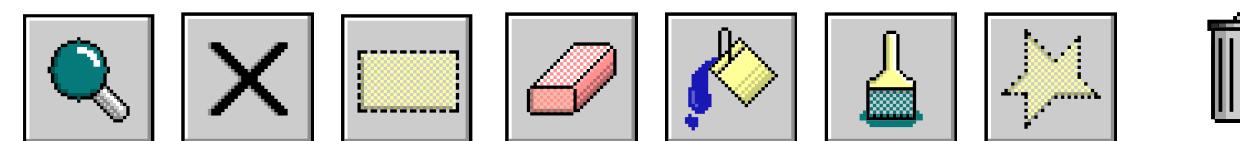
- Recreación de productos que no tienen dibujos 2D o datos CAD 3D para poder reproducirlos.
- Problemas con el fabricante de equipos originales (OEM): Si el OEM ya no opera o ha perdido las medidas de diseño, entonces a través de la ingeniería inversa se puede obtener información vital del producto para continuar con la fabricación de ese objeto.
- Análisis de la competencia: Cualquier organización puede analizar los productos de la competencia y así poder robar sus conocimientos.
- Objetos antiguos y a medida: Cuando no hay información sobre las dimensiones de un objeto, excepto el elemento físico en sí, la forma más rápida es utilizando la ingeniería inversa porque cuando el producto tiene una forma orgánica puede ser difícil diseñarlo en CAD.



Ciberseguridad



- Análisis de malware: El ingenier@ especializado utiliza distintas técnicas de ingeniería inversa para desmontar y comprender el funcionamiento interno de programas maliciosos.
- Retroingeniería: Se emplea para obtener información sobre tecnología patentada o productos existentes en el mercado con el objetivo de desarrollar productos similares o compatibles. Esta práctica es legal siempre y cuando no infrinja los derechos de propiedad intelectual.
- Identificar y comprender vulnerabilidades en software o sistemas.
- Destrucción de protección anticopia: Algunas personas utilizan estas técnicas para poder copiar videojuegos o copiar música bajo la protección de un sistema anticopia (DRM, Digital Rights Management).
- Interoperabilidad: Los desarrolladores utilizan las técnicas de reverse engineering para reescribir tareas que permitan utilizar los productos sobre otras plataformas diferentes a la soportada por el fabricante (es el caso de muchos drivers en Linux).
- Control de calidad de las funciones del software

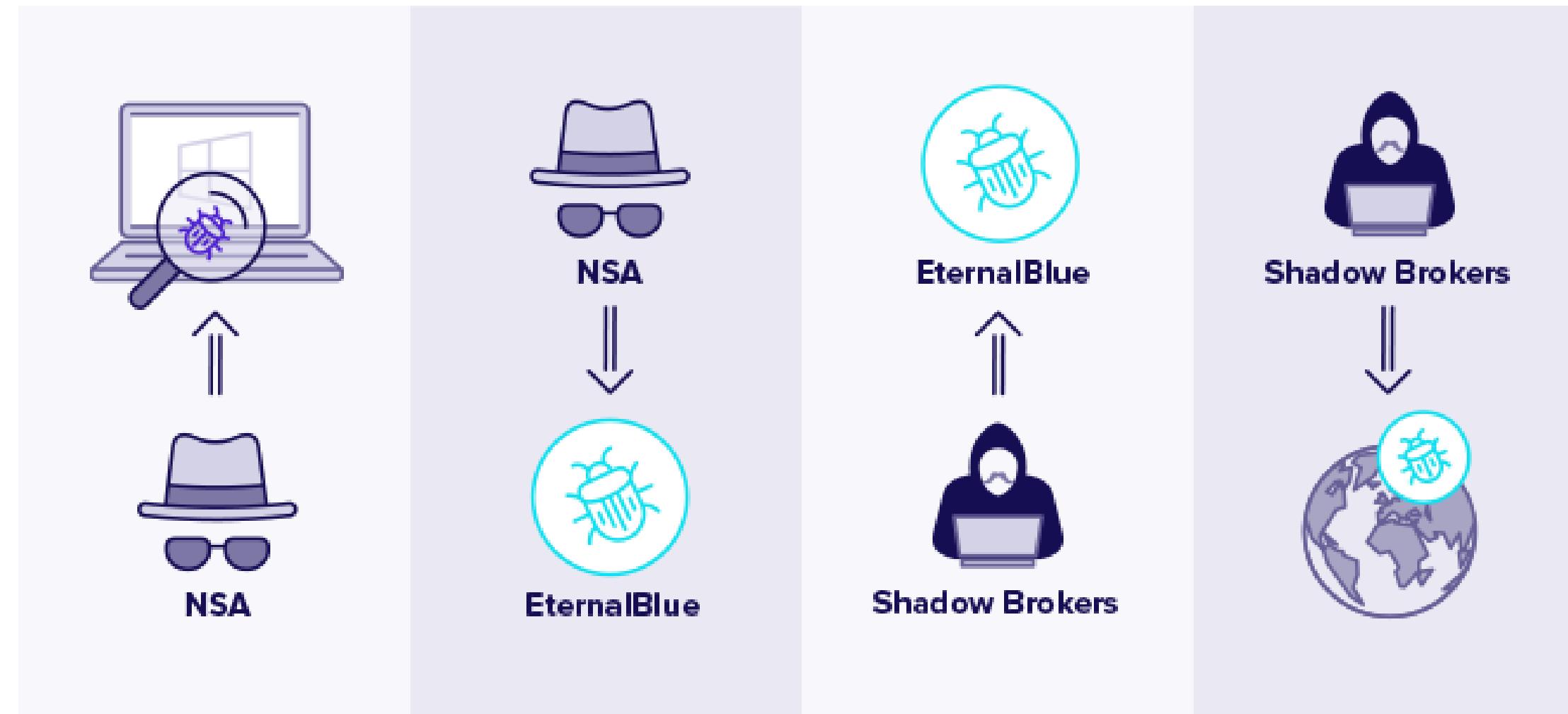


Caso de estudio: Malware WannaCry

Se produjo el 12 de mayo de 2017 e infectó a más de 230 mil computadoras en 150 países, afectando a instituciones estatales y empresas a nivel mundial.



Caso de estudio: Malware WannaCry



- Agencia de Seguridad Nacional de los Estados Unidos (NSA)
- Protocolo Server Message Block (SMB)

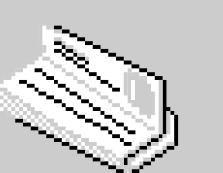
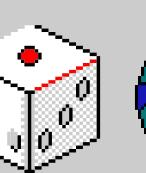
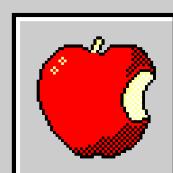
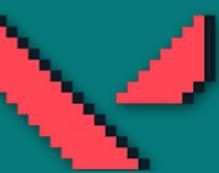
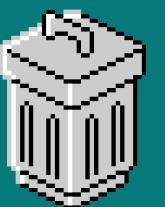
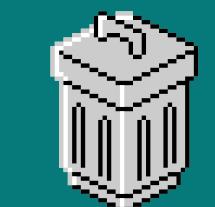
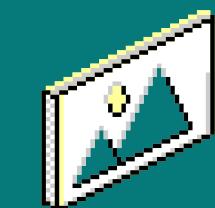
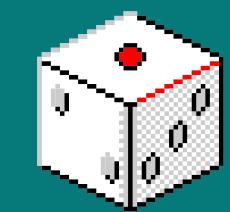
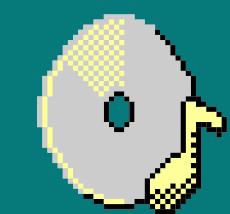
Caso de estudio: Malware WannaCry



Marcus Hutchings

<http://www.iuquerfsodp9ifjaposdfjhgosurijfaewrwegwea.com/>

Conclusiones



[Back to Agenda Page](#)

Gracias por su
atención!



Referencias

- Cortés, A. (2021). Estudio del malware WannaCry utilizando técnicas de reversing (Tesis de grado). Universidad de Alcalá Escuela Politécnica Superior.
https://ebuah.uah.es/dspace/bitstream/handle/10017/49194/TFG_Cortes_%20Arranz_2021.pdf?sequence=1&isAllowed=y
- Ingeniería inversa para maquinaria y procesos. (2023). ALTERTECNIA. <https://altertecnia.com/ingenieria-inversa-maquinaria-y-procesos/>
- La ingeniería inversa de software. (2020). IONOS.
<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/ingenieria-inversa-de-software/>

[Back to Agenda Page](#)

Referencias

- Lares, D. (2022). Malware Sandboxing and Reverse Engineering Overview. Medium. <https://medium.com/nerd-for-tech/malware-sandboxing-and-reverse-engineering-overview-66f6ad84e8c9>
- Latto, N. (2020). ¿Qué es WannaCry?. Avast. <https://www.avast.com/es/es/c-wannacry> • Meza, P. (2023). Reversing. IHack. <https://ihack.red/reversing/>
- Qué es y para qué sirve la ingeniería inversa. (2023). Asorcad. <https://asorcad.es/blog/que-es-y-para-que-sirve-la-ingenieria-inversa/>
- **Rascagneres, P. (2020). Seguridad informática y malwares: análisis de amenazas e implementación de contramedidas (2nd ed.). Editorial ENI**

[Back to Agenda Page](#)

Referencias

- REDACCIÓN. (2020). Aplicaciones industriales más comunes de la ingeniería inversa. Tecnología Para La Industria.
<https://tecnologiaparalaindustria.com/aplicaciones-industriales-mas-comunes-de-la-ingenieria-inversa/>
- Serra, J. (2021). Técnicas y herramientas la ingeniería inversa. Tecnología++. <https://blogs.uoc.edu/informatica/es/ingenieria-inversa-que-es-herramientas-y-tecnicas/>
- yadav, G. (2021). A brief introduction to Packing and Obfuscation.
<https://medium.com/ax1al/packing-and-obfuscation-fe6b03bbc267>

[Back to Agenda Page](#)