# RED TEAM OPS DEVELOPER II

# [RTOD II]

## SPARTAN-CYBERSECURITY

# ADVERTENCIA

**Derechos de Autor:** Todos los derechos sobre este material están reservados. Queda estrictamente prohibido reproducir, distribuir, transmitir, adaptar o modificar, ya sea de manera total o parcial, cualquier parte de esta publicación, sin el consentimiento previo y por escrito del autor. Esto incluye, pero no se limita a, cualquier forma de fotocopia, grabación, emisión, almacenamiento digital o cualquier otro tipo de reproducción o sistema de recuperación y transmisión de información.

**Propósito Educativo:** El contenido presente en este material se proporciona exclusivamente con fines educativos, informativos y de promoción de prácticas éticas. Se enfatiza que todas las técnicas y prácticas mencionadas deben ser aplicadas únicamente en entornos controlados y designados para pruebas, y siempre con los permisos adecuados.

**Exención de Responsabilidad:** Aunque este contenido se presenta con una intención educativa y ética, los autores, editores y distribuidores no pueden garantizar que todos los usuarios del mismo harán un uso adecuado. Por ello, nos desligamos completamente y no asumimos ninguna responsabilidad por cualquier acto indebido, ilegal o inmoral realizado por cualquier individuo o entidad que utilice este material, así como por cualquier daño o perjuicio resultante de dichas acciones.

**Notificación:** Cualquier persona que obtenga o use este material reconoce y acepta estos términos y condiciones, y entiende plenamente las posibles consecuencias legales de la violación de estos términos.

# TABLA DE CONTENIDO

6. Manejo de errores
    a. Panic
    b. Result
7. ¡Uso de ASM! En Rust
    a. Inline Assembly
8. Instalación del marco C2
    a. Instalación de Havoc2 y despliegue
9. Conceptos de soluciones de defensa
    a. Que es un antivirus
    b. Que es un EDR
    c. Diferencias entre avs-edr
10. Windows API
    a. Que es Windows api
    b. Bibliotecas comunes (kernel32.dll, user32.dll, advapi32.dll,ntdll.dll)
    c. Manipulación de memoria (virtualloc, readprocessmemory, writeprocessmemory) conceptos
11. Api Calls
    a. Diagrama de flujo
12. Shellcode
    a. Que es shellcode
    b. Generador de shellcode para los ejercicios (donut)
    c. Desarrollo de proyecto versátil para el envio de shellcode remoto
    d. Implementación de cifrado en el shellcode para evitar análisis estático – (IPS)
13. HtmlSmuggling
    a. Que es htmlSmuggling
    b. Generación payload en template customizada
    c. Creación de herramienta para automizar el smuggling
    d. Caso de uso Practico con payload

14. Técnicas ofensivas y de evasión en Rust
    a. Uso de ensamblador en rust para interactuar con la api de Windows
    b. Shellcode encrypt memory heap usando asm
    c. Cifrado shellcode en aes
    d. Creación de un cargador(loader) en aes
    e. Loader remoto webserver shellcode
15. Xor cipher en rust
    a. Creación de proyecto para cifrar shellcode con key en hexadecimal
16. Inyectando y ejecutando shellcode en memoria
    a. Introducción a a la api de Windows para la gestión de memoria
    b. Descripcion de virtualloc propósito y argumentos
    c. Conceptos de MEM_COMMIT, PAGE_EXECUTE_READWRITE
17. Inyección del shellcode en Partes
    a. Division del shellcode en dos mitades.
    b. Objetivo de dividir el shellcode
    c. Uso de .offset para ajustar direcciones en memoria
    d. Demostración practica inyectando el shellcode en partes
18. Recuperación de la dirección base y función ntdll
    a. Vision general sobre Ntdll
    b. Introducción a la función GetModuleHandle
    c. Creación de get_ntdll_api
    d. Definicion y estructura de NtAllocateVirtualMemory
    e. Explicación de la función mem:trasmute
    f. Obtención de ntdll base usando assembly
19. Patch Amsi
    a. Patch básico de amsi

20. Técnicas ofensivas y de evasión en golang
    21. Cifrado
        a. Que es el cifrado
        b. Proyecto de cifrado usando aes 256
        c. Xor
    22. Obfuscacion
        a. Que es la obfuscacion
        b. Proyecto obfuscador personalizado para shellcode
    23. Encoding
        a. Base64
        b. Base32
        c. Hex
    24. Patch De ETW usando golang
        a. Que es ETW
        b. Aplicaciones y usos de etw en sistemas Windows
        c. Diseño y estructura de parche básico
    25. Patch de Amsi usando Golang
        a. Que es amsi y para que se utiliza
        b. Como funciona el amsi
        c. Casos de estudio
        d. Evasión de amsi
    26. Técnicas anti sandbox
        a. Que es un sandbox
        b. Detección de dispositivos
        c. Búsqueda de procesos específicos
        d. Retrasos de ejecución
        e. Verificación de red

27. Process inyección shellcode
    a. Alocacion de memoria en procesos remotos
    b. NtAllocateVirtualMemory
    c. Escritura en memoria de procesos remotos
    d. GetProcAddr
    e. VirtualProtect
    f. WriteProcessMemory
    g. CreateRemoteThreadEx
28. Simple Reverse Shell
    a. Creación Shell reverse básica
    b. Conversión y ejecución Shell en dll
29. Heap
    a. Que es la memoria heap
    b. Asignación y liberación de la memoria
    c. Asignación de un mensaje cifrado en memoria
    d. Debug process hacker regiones de memoria
    e. Shellcode en diferentes regiones
    f. Encryptacion del shellcode en el heap con AES 256
30. Module Stomping
    a. Que es module stomping
    b. Inyección de código
    c. Module overloading DLL
    d. Inyección amsi.dll
    e. Inyección bcrypt.dll
    f. Debug x64dbg internal
31. Unhooking
    a. Que es unhook
    b. Métodos de unhook
    c. Identificación de dll enganchadas
    d. Detección de ganchos ingeniería inversa X64dbg
    e. Api Unhooking
    f. Unhook ntdll completa

32. Casos de estudio ataques
    a. Unhook bitdefender licenciado
    b. Bypass bitdefender shellcode Havoc2
    c. Bypass kapersky shellcode Havoc2
    d. Ingeniería inversa x64dbg debug ganchos
33. Shellcode ipv4
    a. Que es RtlIpv4StringToAddressA?
    b. Implementación.
    c. Creación de convertidor de shellcode  a ipv4
    d. Creación loader para ejecución de shellcode ipv4
34. Proyecto Websocket Shell reverse
    a. Que es websocket
    b. Implementación
    c. Controladores
    d. Struct información básica del sistema
    e. Validación api key
    f. Cliente
    g. Servidor
    h. Despliegue
35. Droppers
    a. Que es un dropper
    b. Diseño de un dropér básico
    c. Diseño de un droper avanzado
36. Bibliografía y referencias
37. Herramientas, artículos
38. Anexos: ejercicios
39. Practicas recomendadas tras cada modulo

# DETALLES DEL CURSO

**Material entregable:**

- ✓ Acceso a más de 20 horas de contenido en MATERIAL GRABADO.
- ✓ Acceso al código fuente de cada proyecto.
- ✓ Acceso a un grupo exclusivo del curso para interactuar con los demás estudiantes y el profesor.
- ✓ Certificado del RTOD-2 por parte de Spartan-Cybersecurity.

**Costo del curso:** $200 USD

**Comunícate con el área de ventas para conocer nuestros descuentos:**

**WhatsApp:** https://wa.link/j265a0

**Telegram:** https://t.me/Spartan_Cybersecurity

**CUPOS LIMITADOS.**