



Tecnológico Nacional de México

Instituto Tecnológico de Tlaxiaco

Carrera: Ingeniería en Sistemas Computacionales

Materia: Seguridad Y Virtualización

Actividad: Reporte de Practica 5

Alumnos:	Feria Ortiz Eduardo Tomas	21620095
	Reyes Peña Isaí	21620053
	Zárate Reyes Irving	20620166

Grupo: 7US

Catedrático: Ing. Osorio Salinas Edward

Heroica Ciudad de Tlaxiaco.
Domingo, 06 de octubre de 2024.





Índice

Práctica. Protección contra ataques.....	3
1. Crear un programa que simule un ataque de fuerza bruta.	3
2. Crear un programa que simule un ataque de denegación de servicio.....	5
3.- Posibles acciones que se pueden realizar para prevenir este tipo de ataques.....	12
Prevención de ataques de fuerza bruta	12
Prevención de ataques de denegación de servicio (DoS)	14
Investigación.....	16
Ataque de fuerza bruta	16
Ataque de denegación de servicio (DoS)	17
Ataque económico de denegación de servicio (EDoS)	17
Ataque de denegación de servicio distribuido (DDoS).....	19
Ataque de denegación de servicio por agotamiento de recursos	21
Ataque de denegación de servicio por saturación de ancho de banda.....	22
Conclusión.....	23
Bibliografía:.....	24

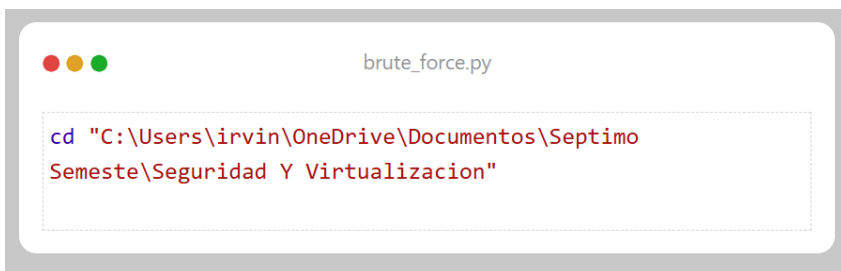


Práctica. Protección contra ataques

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

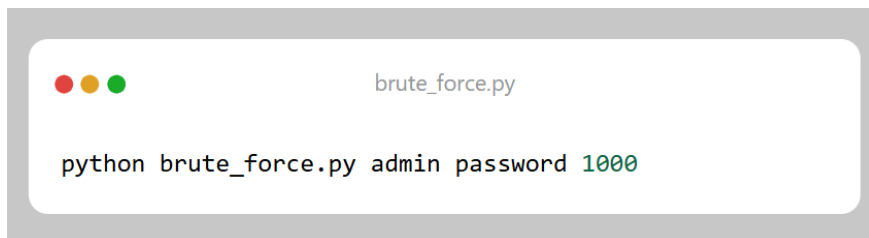
Comenzando con nuestra práctica, creamos dos archivos en Python, para comenzar a trabajar. Nos apoyamos también del código disponible en el repositorio de GitHub, lo modificamos a nuestra conveniencia y necesidades, para posteriormente ejecutarlo.

Navegamos por nuestra terminal hasta la carpeta que contenía a nuestro archivo .py.



```
brute_force.py  
  
cd "C:\Users\irvin\OneDrive\Documentos\Septimo  
Semestre\Seguridad Y Virtualizacion"
```

Una vez en ella ejecutamos la siguiente línea de código para mandar a llamar nuestro programa y de igual manera proporcionarle los parámetros.



```
brute_force.py  
  
python brute_force.py admin password 1000
```

En esta línea podemos apreciar los parámetros proporcionados:

- admin es el nombre de usuario.

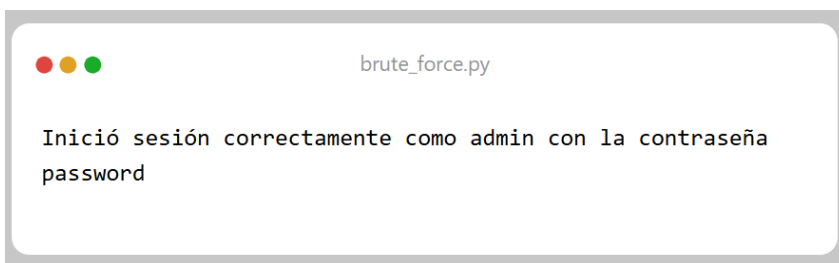


- password es la contraseña que estamos probando.
- 1000 es el número de intentos máximos permitidos.

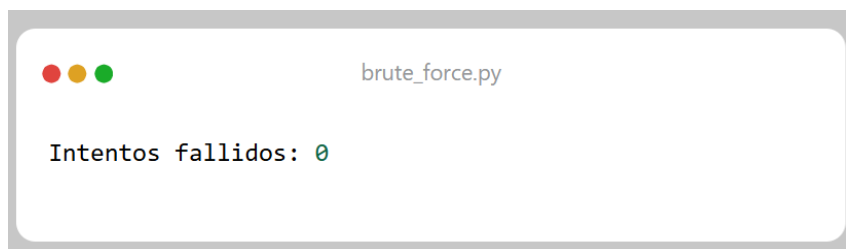
Y como respuesta a esto obtuvimos las siguientes líneas:

```
PS C:\Users\irvin> cd "C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion"
PS C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion> python brute_force.py admin password 1000
Inició sesión correctamente como admin con la contraseña password
Intentos fallidos: 0
Tiempo transcurrido: 0.00 segundos
Combinaciones intentadas: 1
```

Esto nos indica que el programa intento iniciar sección con las credenciales proporcionadas y lo logro en el primer intento. Con lo cual nos muestra los siguientes mensajes:



Esta línea es la esperada y nos muestra que la sesión se logro de manera exitosa.



Con esta línea comprobamos que el inicio de sección se dio de manera exitosa desde el primer intento.

```
brute_force.py

Tiempo transcurrido: 0.00 segundos
```

Esta línea nos muestra el tiempo de ejecución del programa.

```
brute_force.py

Combinaciones intentadas: 1
```

En esta última línea nos indica que solo se intente una vez para lograr el inicio de sección.

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

Para esta segunda parte de la práctica, creamos un nuevo archivo de Python, igualmente apoyando del script proporcionado en GitHub como ejemplo y guía para la realización.

```
PS C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion> python dos.py "127.0.0.1" 80 1000
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
Error al enviar solicitud: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión
```



Después de algunos ajustes y ejecución nos encontramos con un problema.

```
brute_force.py

Error al enviar solicitud: [WinError 10061] No se puede
establecer una conexión ya que el equipo de destino denegó
expresamente dicha conexión
```

Los resultados de nuestra ejecución eran todos errores al enviar la solicitud.

```
brute_force.py

python dos.py "127.0.0.1" 80 1000
```

Y esto se debía a que estamos enviando la IP 127.0.0.1 para localhost. Después de una investigación pudimos concluir que era necesario crear un servidor web que estuviera corriendo.

Decimos designar al puerto 80 para esta función.

```
hello-world.js

cd C:\Users\irvin\Desktop
```



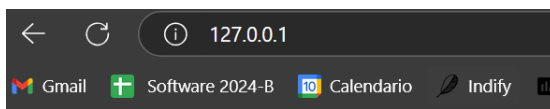


Navegamos por el directorio al lugar donde deseábamos levantar el servidor en nuestro caso elegimos el escritorio.

```
hello-world.js  
  
python -m http.server 80
```

Posterior a eso, ejecutamos esta línea de código, la cual nos permitió iniciar el servidor.

Para corroborar que el servidor estaba funcionando fuimos al navegador y colocamos la dirección ip asignada



Directory listing for /

- [.android/](#)
- [.bash_history](#)
- [.cache/](#)
- [.dotnet/](#)
- [.emulator_console_auth_token](#)
- [.gitconfig](#)
- [.gradle/](#)
- [.Icecream Ebook Reader/](#)
- [.nbi/](#)
- [.nuget/](#)
- [.packettracer](#)

Nuestro servidor ya estaba listo.



```
Símbolo del sistema - python × + v
Microsoft Windows [Versión 10.0.22631.4249]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\irvin>cd C:\Users\irvin\Desktop
El sistema no puede encontrar la ruta especificada.

C:\Users\irvin>python -m http.server 80
Serving HTTP on :: port 80 (http://[::]:80/) ...
```

Por último, proseguimos a las pruebas.

Con esta línea de código ejecutamos nuestro programa:

```
brute_force.py

python dos.py "127.0.0.1" 80 1000
```

- 127.0.0.1: Es la dirección IP del servidor.
- 80: Es el puerto del servidor web.
- 1000: Es el número de solicitudes que quieres enviar.

Tuvimos distintos resultados, comenzamos probando con 10, 100, 1,000 y 2,000 solicitudes. Cada una con muy distintos resultados, tal y como se esperaba.




```
PS C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion> python dos.py "127.0.0.1" 80 10
Se enviaron 10 solicitudes
Tiempo transcurrido: 0.5196573734283447 segundos
PS C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion> python dos.py "127.0.0.1" 80 100
Se enviaron 100 solicitudes
Tiempo transcurrido: 9.229613780975342 segundos
PS C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion> python dos.py "127.0.0.1" 80 1000
Error al enviar solicitud: [WinError 10054] Se ha forzado la interrupción de una conexión existente por el host remoto
Se enviaron 1000 solicitudes
Tiempo transcurrido: 89.82048487663269 segundos
PS C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion> python dos.py "127.0.0.1" 80 2000
Se enviaron 2000 solicitudes
Tiempo transcurrido: 188.49538779258728 segundos
PS C:\Users\irvin\OneDrive\Documentos\Septimo Semestre\Seguridad Y Virtualizacion> 
```

En el primer caso, 10 solicitudes fue un número pequeño elegido para corroborar que el código ya funcionara sin problemas y no tuviera problemas.



Las 10 solicitudes tardaron 0.519 segundos en ejecución. Así que después de concluir satisfactoriamente esta primera prueba proseguimos a las siguientes.





100 solicitudes en un tiempo de 9.229 segundos.

```
hello-world.js

python dos.py "127.0.0.1" 80 1000
Error al enviar solicitud: [WinError 10054] Se ha
forzado la interrupción de una conexión existente por
el host remoto
Se enviaron 1000 solicitudes
Tiempo transcurrido: 89.82048487663269 segundos
```

Las 100 solicitudes ocurrieron en un tiempo de 89 segundos. Durante su ejecución, pudimos ver un error de conexión.

Los errores que aparecen en la consola del servidor son comunes cuando se realizan pruebas de saturación. Estos errores como ConnectionAbortedError y WinError 10053 indican que el servidor no pudo manejar todas las solicitudes enviadas en un corto período de tiempo y las conexiones se están cerrando. Esto es esperable durante un ataque de DoS.

```
hello-world.js

python dos.py "127.0.0.1" 80 2000
Se enviaron 2000 solicitudes
Tiempo transcurrido: 188.49538779258728 segundos
```





Por último 2,000 solicitudes, las cuales se realizaron en un tiempo de 188 segundos.

```
Símbolo del sistema - python x + v

File "C:\Python312\Lib\http\server.py", line 878, in copyfile
  shutil.copyfileobj(source, outputfile)
File "C:\Python312\Lib\shutil.py", line 204, in copyfileobj
  fdst_write(buf)
File "C:\Python312\Lib\socketserver.py", line 840, in write
  self._sock.sendall(b)

-----
Exception occurred during processing of request from (::ffff:127.0.0.1', 56049, 0, 0)
ConnectionAbortedError: [WinError 10053] Se ha anulado una conexión establecida por el software en su equipo host
-----
Traceback (most recent call last):
  File "C:\Python312\Lib\socketserver.py", line 692, in process_request_thread
    self.finish_request(request, client_address)
  File "C:\Python312\Lib\http\server.py", line 1311, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "C:\Python312\Lib\http\server.py", line 672, in __init__
    super().__init__(*args, **kwargs)
  File "C:\Python312\Lib\socketserver.py", line 761, in __init__
    self.handle()
  File "C:\Python312\Lib\http\server.py", line 436, in handle
    self.handle_one_request()
  File "C:\Python312\Lib\http\server.py", line 424, in handle_one_request
    method()
  File "C:\Python312\Lib\http\server.py", line 679, in do_GET
    self.copyfile(f, self.wfile)
  File "C:\Python312\Lib\http\server.py", line 878, in copyfile
    shutil.copyfileobj(source, outputfile)
  File "C:\Python312\Lib\shutil.py", line 204, in copyfileobj
    fdst_write(buf)
```

Mientras las solicitudes se ejecutaban, en la terminal donde levantamos nuestro servidor web podíamos ver las respuestas a cada una de ellas, incluso los errores de conexión. Esto indicando que la practica se ha realizado con éxito.

3.- Posibles acciones que se pueden realizar para prevenir este tipo de ataques.

Prevención de ataques de fuerza bruta

Bloqueo temporal de cuentas

Una de las estrategias más efectivas para prevenir ataques de fuerza bruta es bloquear temporalmente una cuenta después de un número específico de intentos fallidos de inicio de sesión. Esta técnica puede aplicarse mediante un sistema de temporización progresiva, en el que después de un número de intentos fallidos, el tiempo de espera entre los siguientes intentos aumenta de forma exponencial, dificultando así los ataques automatizados.

Ejemplo: Si un usuario falla cinco intentos de inicio de sesión, se bloquea el acceso por 15 minutos. Si los intentos continúan fallando, el bloqueo puede incrementarse a una hora o incluso a un día.

Autenticación multifactor (MFA)

La autenticación multifactor agrega una segunda capa de seguridad. Incluso si un atacante logra adivinar o descifrar la contraseña, sin el segundo factor de autenticación, no podrá acceder al sistema. Este segundo factor puede ser un código temporal enviado al teléfono móvil del usuario o generado por una aplicación de autenticación.

Ejemplo: Un usuario necesita ingresar su contraseña y luego un código de seis dígitos enviado a su dispositivo móvil para completar el inicio de sesión.



Uso de Captchas

Un Captcha puede evitar que los bots automatizados realicen intentos masivos de inicio de sesión. Al requerir que los usuarios completen un desafío de reconocimiento humano (como identificar imágenes o resolver un rompecabezas), se logra interrumpir los ataques de fuerza bruta automatizados.

Ejemplo: Después de tres intentos fallidos de inicio de sesión, el sistema solicita al usuario resolver un Captcha antes de poder intentar nuevamente.

Control de accesos basado en la ubicación y el dispositivo

Implementar restricciones de inicio de sesión basadas en la geolocalización del usuario o el dispositivo desde el cual acceden puede ayudar a bloquear intentos sospechosos. Si un usuario normalmente inicia sesión desde una ubicación geográfica específica y de repente se observa un intento desde una ubicación remota, el sistema puede solicitar verificación adicional o bloquear el intento.

Ejemplo: Un usuario que inicia sesión regularmente desde México recibe una notificación cuando se detecta un intento de inicio de sesión desde un país diferente.

Monitoreo y detección de patrones anómalos

Utilizar herramientas de monitoreo de seguridad que identifiquen patrones anómalos de inicio de sesión puede alertar sobre posibles ataques de fuerza bruta. Las soluciones de detección de intrusiones (IDS) y prevención de intrusiones (IPS) permiten detectar intentos sospechosos antes de que se complete un ataque.

Ejemplo: Un sistema de monitoreo detecta múltiples intentos fallidos desde una misma IP y genera una alerta de seguridad.





Prevención de ataques de denegación de servicio (DoS)

Firewall de aplicaciones web (WAF)

Un WAF es una herramienta que filtra y monitorea el tráfico HTTP hacia y desde una aplicación web. Detecta y bloquea solicitudes maliciosas antes de que lleguen al servidor. Los WAF modernos son efectivos contra ataques DoS al bloquear solicitudes sospechosas, limitando la cantidad de tráfico que un servidor puede recibir de una sola fuente.

Ejemplo: Un WAF bloquea el tráfico de una dirección IP que intenta enviar miles de solicitudes en un corto periodo de tiempo, evitando que el servidor se sature.

Balanceo de carga (Load Balancing)

El balanceo de carga distribuye el tráfico entre varios servidores, evitando que un solo servidor quede sobrecargado por un ataque DoS. Si un servidor comienza a recibir demasiado tráfico, el balanceador puede redirigir las solicitudes a otros servidores, garantizando que el sistema siga funcionando.

Ejemplo: Durante un ataque DoS, un balanceador de carga redirige parte del tráfico a un servidor menos congestionado para evitar la caída del servicio.

Limitación de tasa (Rate Limiting)

Implementar la limitación de tasa significa restringir la cantidad de solicitudes que una dirección IP puede hacer en un periodo específico de tiempo. Esta técnica reduce la posibilidad de que una sola fuente sobrecargue el servidor con un número elevado de solicitudes.





Ejemplo: Un servidor permite que cada dirección IP realice un máximo de 100 solicitudes por minuto. Si se excede este límite, el tráfico adicional se bloquea temporalmente.

Protección contra ataques DDoS

Los ataques DDoS son versiones distribuidas de DoS, donde múltiples sistemas atacan simultáneamente. Las soluciones especializadas de protección contra DDoS, como las ofrecidas por Cloudflare o Akamai, pueden mitigar estos ataques al filtrar el tráfico antes de que llegue al servidor. Estas soluciones analizan el comportamiento del tráfico y pueden identificar y bloquear solicitudes maliciosas.

Ejemplo: Una empresa contrata un servicio de protección DDoS que detecta un aumento anormal en el tráfico y bloquea el tráfico sospechoso antes de que afecte al servidor.

Configuración adecuada del servidor

Ajustar la configuración del servidor web puede prevenir ataques DoS al optimizar el uso de recursos y evitar que las conexiones inactivas sobrecarguen el servidor. Esto incluye cerrar conexiones no utilizadas rápidamente y manejar grandes cantidades de tráfico de manera eficiente.

Ejemplo: Un servidor configurado para cerrar conexiones inactivas después de 30 segundos puede liberar recursos y reducir el impacto de un ataque DoS.





Investigación

Investiga y describe los siguientes conceptos:

Ataque de fuerza bruta

Un ataque de fuerza bruta es una de las formas más rudimentarias pero efectivas de atacar un sistema de seguridad. Básicamente, este ataque consiste en probar todas las combinaciones posibles de credenciales, como nombres de usuario y contraseñas, hasta encontrar la correcta. Es como intentar abrir una puerta probando todas las llaves disponibles, una tras otra, hasta que finalmente encuentras la que encaja. El proceso no es muy sofisticado, pero puede ser sorprendentemente efectivo si las medidas de seguridad no son lo suficientemente robustas.

La forma en que funciona un ataque de fuerza bruta se basa en la persistencia y la capacidad computacional. El atacante emplea programas que pueden generar combinaciones de caracteres a gran velocidad, lo que permite probar miles o millones de combinaciones en poco tiempo. Sin embargo, la rapidez con la que este ataque puede tener éxito depende de dos factores principales: la fortaleza de la contraseña y los recursos que tiene el atacante. Por ejemplo, una contraseña corta o común puede ser descubierta en minutos, mientras que una contraseña más larga y compleja podría resistir durante años de intentos continuos.





Ataque de denegación de servicio (DoS)

Un ataque de denegación de servicio es un tipo de ciberataque cuyo objetivo es hacer que un servidor, sitio web o red quede inaccesible para los usuarios legítimos. Esto se logra saturando el sistema objetivo con una cantidad masiva de solicitudes o sobrecargando los recursos del servidor, de modo que no pueda gestionar más peticiones, lo que lleva a su ralentización o caída total.

Los ataques DoS pueden tomar diferentes formas. Uno de los más comunes es el ataque de inundación, donde el atacante envía una cantidad enorme de solicitudes de conexión al servidor objetivo, más de las que puede manejar. Debido a que los servidores tienen una capacidad limitada de procesamiento y memoria, cuando se sobrepasan esos límites, el servidor empieza a fallar o se vuelve incapaz de responder a nuevas peticiones.

Otro método es el ataque de amplificación, en el cual el atacante utiliza recursos mínimos para desencadenar una respuesta masiva del sistema. Esto se logra al explotar protocolos que generan una respuesta mucho mayor en comparación con el tamaño de la solicitud enviada. Como resultado, el sistema queda saturado de respuestas que no puede gestionar, lo que lleva a su colapso.

Ataque económico de denegación de servicio (EDoS)

Un ataque económico de denegación de servicio es una variante más específica y sofisticada del ataque de denegación de servicio, en la que el objetivo principal no es simplemente interrumpir el servicio de un servidor o red, sino causar un





daño financiero directo a la víctima. En lugar de buscar la saturación y colapso inmediato del sistema, un ataque EDoS está diseñado para aprovechar los modelos de facturación de servicios en la nube, que cobran en función de los recursos utilizados, como el ancho de banda, la CPU o el almacenamiento.

Este tipo de ataque se basa en el hecho de que muchos servicios en la nube escalan automáticamente sus recursos para hacer frente a la demanda. Cuando un servidor recibe una gran cantidad de tráfico, el sistema responde aumentando la capacidad para gestionar ese tráfico adicional. Aunque esto permite que el servicio siga funcionando y no se caiga, la consecuencia es que el costo de los recursos utilizados también aumenta. Aquí es donde entra el ataque EDoS: los atacantes envían un volumen continuo y elevado de tráfico falso, no para derribar el sistema, sino para obligar a la víctima a pagar por los recursos adicionales necesarios para gestionar esa carga.

Los ataques EDoS son especialmente problemáticos porque pueden ser difíciles de detectar. Mientras que en un ataque DoS el objetivo es hacer que el servicio sea inaccesible, lo que puede ser detectado más fácilmente mediante herramientas de monitoreo de disponibilidad, en un EDoS el tráfico puede parecer legítimo en la superficie. Los atacantes a menudo distribuyen las solicitudes desde diferentes direcciones IP y regiones geográficas, lo que complica aún más la detección.





Ataque de denegación de servicio distribuido (DDoS)

Un ataque de denegación de servicio distribuido (DDoS) es una versión más avanzada y dañina del ataque DoS, donde múltiples dispositivos comprometidos, a menudo conocidos como "bots" o "zombies", se utilizan de manera coordinada para inundar un servidor, red o servicio con una abrumadora cantidad de tráfico. A diferencia del ataque DoS tradicional, que proviene de una única fuente, el DDoS involucra múltiples dispositivos ubicados en diferentes lugares, lo que hace que el ataque sea mucho más difícil de mitigar o bloquear. La red de dispositivos comprometidos que lanzan el ataque se denomina botnet, y puede estar compuesta por cientos o incluso millones de dispositivos infectados, desde computadoras hasta dispositivos IoT (Internet of Things), como cámaras de seguridad o electrodomésticos conectados a la red.

El propósito de un ataque DDoS es similar al de un DoS: sobrecargar los recursos del servidor o de la red objetivo, haciendo que el sistema se vuelva extremadamente lento o, en muchos casos, inaccesible para los usuarios legítimos. Al recibir una avalancha de solicitudes simultáneas, el servidor se ve incapaz de gestionarlas todas, lo que genera interrupciones o fallos en el servicio. Este tipo de ataque puede afectar desde pequeños sitios web hasta grandes infraestructuras en la nube, y las consecuencias varían dependiendo de la naturaleza del servicio afectado: interrupciones en negocios en línea, caídas de plataformas de comunicación o incluso el colapso de servicios críticos.





Una de las características más complicadas de los ataques DDoS es que el tráfico generado puede parecer legítimo. Debido a que proviene de diferentes direcciones IP y ubicaciones geográficas, es difícil filtrar las solicitudes maliciosas sin afectar también a usuarios reales. Además, los dispositivos utilizados para lanzar el ataque a menudo están comprometidos sin que sus propietarios lo sepan, lo que hace que los atacantes puedan ocultar su verdadera identidad detrás de estos dispositivos infectados.

Los ataques DDoS suelen clasificarse en diferentes tipos, dependiendo de la forma en que se ejecutan. Algunos de los más comunes incluyen ataques de capa de aplicación (que apuntan directamente a aplicaciones web, como el envío masivo de solicitudes HTTP), ataques de amplificación (que utilizan un pequeño paquete de datos para generar una respuesta mucho mayor en el servidor) y ataques de agotamiento de ancho de banda (que inundan la red con un volumen de datos tan grande que satura la conexión).

Los efectos de un ataque DDoS pueden ser devastadores para las empresas y organizaciones. Además de la pérdida temporal de servicios, los costos asociados a un ataque DDoS pueden incluir la pérdida de ingresos, el daño a la reputación de la empresa y los gastos adicionales en personal técnico y medidas de mitigación. En algunos casos, los atacantes DDoS incluso utilizan estos ataques como parte de una estrategia de extorsión, exigiendo pagos para detener el ataque.





Ataque de denegación de servicio por agotamiento de recursos

Un ataque de denegación de servicio por agotamiento de recursos es un tipo de ciberataque que tiene como objetivo agotar los recursos de un servidor, sistema o red para que no pueda procesar las solicitudes legítimas de los usuarios. A diferencia de otros tipos de ataques de denegación de servicio que buscan saturar la conexión a la red o sobrecargar la capacidad de procesamiento del servidor, este tipo de ataque se centra en agotar recursos específicos como la memoria, el CPU, el ancho de banda o incluso los archivos temporales del sistema. En un ataque de agotamiento de recursos, los atacantes suelen aprovechar vulnerabilidades en las aplicaciones o configuraciones de los servidores para enviar una gran cantidad de solicitudes que demandan más recursos de los que el sistema puede manejar. Esto provoca que el servidor se vuelva extremadamente lento o se bloquee por completo, dejando fuera de servicio tanto al sistema como a las aplicaciones que dependen de él. Los efectos pueden ser graves, especialmente para las empresas que dependen de la disponibilidad constante de sus sistemas para mantener operaciones comerciales, lo que puede resultar en pérdida de ingresos y daños a la reputación.





Ataque de denegación de servicio por saturación de ancho de banda

Un ataque de denegación de servicio por saturación de ancho de banda es una técnica utilizada por los ciberatacantes para sobrecargar la capacidad de transmisión de datos de una red, lo que impide que el tráfico legítimo acceda a los servicios en línea. Este tipo de ataque se basa en enviar una cantidad masiva de datos o peticiones a un servidor o a una red, con la intención de consumir todo el ancho de banda disponible. Como resultado, los usuarios genuinos no pueden acceder a los recursos porque la red o el servidor están ocupados procesando el flujo excesivo de datos generados por el atacante.

En estos ataques, los hackers suelen utilizar herramientas automatizadas o botnets, una red de dispositivos comprometidos, para lanzar el ataque desde múltiples fuentes. Esto aumenta la intensidad del ataque y lo hace más difícil de contrarrestar, ya que el tráfico malicioso no proviene de una única ubicación, sino de diversas direcciones IP, lo que puede dificultar la identificación y bloqueo de los atacantes. El objetivo es consumir todo el ancho de banda de la red hasta que ésta se vuelva inoperativa o extremadamente lenta.

El efecto principal de un ataque de saturación de ancho de banda es que los paquetes de datos legítimos, como solicitudes de usuarios reales, no pueden llegar a su destino. Esto es especialmente devastador para empresas que dependen de servicios en línea, como páginas web de comercio electrónico, servicios en la nube o sistemas financieros, ya que el tiempo de inactividad puede provocar pérdidas financieras considerables y afectar la confianza de los usuarios.





Existen varias formas de este tipo de ataque, siendo una de las más comunes el flooding (inundación) de paquetes. Esto ocurre cuando se envía una gran cantidad de paquetes de datos, a menudo utilizando técnicas como la saturación de paquetes ICMP (Internet Control Message Protocol), también conocido como Ping Flooding, o la sobrecarga de peticiones HTTP.

Conclusión

Esta práctica nos permitió comprender y analizar en profundidad los ataques cibernéticos, así como reflexionar sobre las medidas de protección que pueden implementarse, cómo funcionan y con qué fines son creadas. A través de la simulación, fue posible visualizar estos ataques desde una perspectiva práctica, replicando sus efectos en nuestro servidor local y, a su vez, observando las consecuencias de una configuración incorrecta en términos de seguridad.

La investigación complementó esta experiencia práctica al ofrecernos un panorama más detallado, con clasificaciones claras y características particulares para cada tipo de ataque. Pudimos notar que no todos los ataques persiguen los mismos fines, como se evidenció en nuestra revisión de la literatura. En conjunto, tanto la simulación como el análisis teórico subrayan la relevancia de una buena configuración del servidor y la importancia de entender las motivaciones detrás de los ataques para estar mejor preparados y proteger nuestros sistemas.





Bibliografía:

- Lopez, V. (2013). Papel de la explosión combinacional en ataques de fuerza bruta. Investigación e Innovación en Ingenierías, 1(1).
- Osorio, E. F. R., Zea, M. P. C., & Casanova, W. A. C. (2020). Evaluación de ataques DDoS y fuerza bruta utilizando entorno virtual Kali Linux como plataforma experimental. Dilemas contemporáneos: Educación, Política y Valores.
- Avalos, H., & Gómez, E. (2015). Seguridad de la información, Generación y Mitigación de un Ataque de Denegación de Servicios. Revista Tecnológica-ESPOL, 28(5).
- Narváez, D., Romero, C., & Núñez, M. (2010). Evaluación de ataques de Denegación de servicio DoS y DDoS, y mecanismos de protección. GEEKS DECC-REPORTS, 2(1).
- Macía Fernández, G. (2007). Ataques de denegación de servicio a baja tasa contra servidores.
- Fuertes, W., Rodas, F., & Toscano, D. (2011). Evaluación de ataques UDP Flood utilizando escenarios virtuales como plataforma experimental. Facultad de Ingeniería, 20(31), 37-53.
- Pacheco Manotas, M. (2022). Ataques de denegación de servicio distribuido, Cómo evitarlos y cómo enfrentarlos.

