GOBIERNO DE ESPAÑA
MINISTERIO DE HACIENDA Y ADMINISTRACIONES PÚBLICAS

CCN
CENTRO CRIPTOLÓGICO NACIONAL

GUÍA DE SEGURIDAD DE LAS TIC (CCN-STIC-820)

GUÍA DE PROTECCIÓN CONTRA DENEGACIÓN DE SERVICIO

ens
Esquema Nacional de Seguridad

JUNIO 2013

Edita:

NIPO 002-13-021-9
Fecha de Edición: junio de 2013
Deloitte ha participado en la elaboración y modificación del presente documento y sus anexos.
El Ministerio de Hacienda y Administraciones Públicas ha financiado el desarrollo del presente documento y sus anexos.

PRÓLOGO

El uso masivo de las tecnologías de la información y las telecomunicaciones (TIC), en todos los ámbitos de la sociedad, ha creado un nuevo espacio, el ciberespacio, donde se producirán conflictos y agresiones, y donde existen ciberamenazas que atentarán contra la seguridad nacional, el estado de derecho, la prosperidad económica, el estado de bienestar y el normal funcionamiento de la sociedad y de las administraciones públicas.

La Ley 11/2002, de 6 de mayo, reguladora del Centro Nacional de Inteligencia, encomienda al Centro Nacional de Inteligencia el ejercicio de las funciones relativas a la seguridad de las tecnologías de la información en su artículo 4.e), y de protección de la información clasificada en su artículo 4.f), a la vez que confiere a su Secretario de Estado Director la responsabilidad de dirigir el Centro Criptológico Nacional en su artículo 9.2.f).

Partiendo del conocimiento y la experiencia del CNI sobre amenazas y vulnerabilidades en materia de riesgos emergentes, el Centro realiza, a través de su Centro Criptológico Nacional, regulado por el Real Decreto 421/2004, de 12 de marzo, diversas actividades directamente relacionadas con la seguridad de las TIC, orientadas a la formación de personal experto, a la aplicación de políticas y procedimientos de seguridad, y al empleo de tecnologías de seguridad adecuadas.

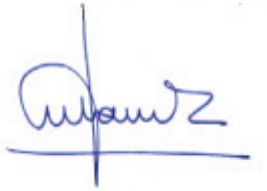
Una de las funciones más destacables del Centro Criptológico Nacional es la de elaborar y difundir normas, instrucciones, guías y recomendaciones para garantizar la seguridad de los sistemas de las tecnologías de la información y las comunicaciones de la Administración, materializada en la existencia de la serie de documentos CCN-STIC.

Disponer de un marco de referencia que establezca las condiciones necesarias de confianza en el uso de los medios electrónicos es, además, uno de los principios que establece la ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los servicios públicos, en su artículo 42.2 sobre el Esquema Nacional de Seguridad (ENS).

Precisamente el Real Decreto 3/2010 de 8 de Enero de desarrollo del Esquema Nacional de Seguridad fija los principios básicos y requisitos mínimos así como las medidas de protección a implantar en los sistemas de la Administración, y promueve la elaboración y difusión de guías de seguridad de las tecnologías de la información y las comunicaciones por parte de CCN para facilitar un mejor cumplimiento de dichos requisitos mínimos.

En definitiva, la serie de documentos CCN-STIC se elabora para dar cumplimiento a los cometidos del Centro Criptológico Nacional y a lo reflejado en el Esquema Nacional de Seguridad, conscientes de la importancia que tiene el establecimiento de un marco de referencia en esta materia que sirva de apoyo para que el personal de la Administración lleve a cabo su difícil, y en ocasiones, ingrata tarea de proporcionar seguridad a los sistemas de las TIC bajo su responsabilidad.

Junio de 2013



Félix Sanz Roldán
Secretario de Estado
Director del Centro Criptológico Nacional

ÍNDICE

1. INTRODUCCIÓN

1. La denegación de servicio (conocida como DoS, por sus siglas en inglés *Denial Of Service*) es un tipo de ataque informático orientado a interrumpir la disponibilidad de un servicio provocando que usuarios legítimos no dispongan de acceso al mismo.
2. El objetivo de este tipo de ataques no es conseguir acceso no autorizado para leer o modificar información, sino provocar la inutilización o destrucción de un activo. Esta particularidad hace este tipo de ataques informáticos distinto al resto por lo que la forma de afrontarlos también es diferente.
3. Dada la complejidad de este tipo de ataques esta guía introduce una clasificación del problema con el objetivo de entender cómo puede afectar un ataque de este tipo a una organización, qué características puede tener y hacia qué activos puede estar dirigido.
4. El Esquema Nacional de Seguridad en el Ámbito de la Administración Electrónica [1], en función de la clasificación del activo, especifica una serie de requisitos que deben ser aplicados a los sistemas de información para protegerlos contra ataques de denegación de servicio. Las contramedidas contempladas en esta guía están centradas en cumplir dichos requisitos.

2. OBJETO

5. Objeto de la presente guía:
   - Definir en qué consiste un ataque de denegación de servicio;
   - Qué tipos de ataques existen y cómo clasificarlos;
   - Definir qué factores se deben tener en cuenta a la hora diseñar protecciones contra ataques de denegación de servicio;
   - Describir qué mitigaciones deben ser aplicadas para cumplir los requisitos del Esquena Nacional de Seguridad en el ámbito de la Administración Electrónica [1] en cuanto a protección contra ataques de denegación de servicio.

3. ALCANCE

6. El Responsable de Seguridad, determinará el alcance de su aplicación, considerando la Política de Seguridad de la Organización y los requisitos de obligado cumplimiento definidos por el Esquema Nacional de Seguridad.
7. Este documento describe las características de seguridad que deben ser contempladas para hacer frente a ataques de denegación de servicio (DoS) dentro del ámbito del Esquema Nacional de Seguridad.

4. CONCEPTOS PREVIOS

4.1 DEFINICIÓN DEL PROBLEMA

8. Una denegación de servicio ataca a la fuente de información o al canal de transmisión impidiendo el acceso a un recurso informático por parte de usuarios con fines legítimos, como puede ser navegar por la web, realizar una transferencia bancaria, enviar un correo electrónico o incluso podría interrumpir el funcionamiento de un sistema SCADA (*Supervisory Control And Data Adquisition*). Este tipo de sistemas

son aplicaciones software que se encargan de controlar y supervisar procesos de producción y suelen verse en fábricas, plantas de tratamiento, etc.

4.2 MOTIVACIÓN Y FUENTE DE LOS ATAQUES

9. En la actualidad los motivos por los cuales se efectúan este tipo de ataques pueden ser muy diversos. En términos generales los motivos suelen ser:
    - Motivos personales o venganza;
    - Extorsión: los atacantes extorsionan al administrador del sistema para conseguir un beneficio económico a cambio de parar el ataque;
    - Sabotaje: se ataca un servicio de la competencia para causarle pérdidas y así conseguir una ventaja en el mercado;
    - Prestigio: El atacante busca prestigio y reconocimiento.
10. Los orígenes de un ataque pueden ser variados:
    - Hacker que actúa por su cuenta: persona con altos conocimientos informáticos que ataca de forma individual un servicio, generalmente por motivos personales, políticos o de prestigio;
    - Hackers que actúan en grupo: grupo de personas con altos conocimientos informáticos que atacan de forma organizada un activo de un servicio, los motivos suelen ser políticos;
    - Script kiddies: personas sin grandes conocimientos informáticos que se limitan a utilizar herramientas desarrolladas por otros;
    - Cibercriminales: buscan el beneficio económico;
    - La competencia: busca conseguir ventaja en el mercado;
    - Uso incorrecto accidental por parte de un empleado.

4.3 IMPACTO DE UN ATAQUE DE DOS

11. Vistas las características de un ataque de DoS, el siguiente paso es visualizar y entender el impacto que tiene este tipo de ataques en las organizaciones.
12. Un ataque de DoS es un fenómeno conocido desde hace muchos años cuya importancia ha ido en aumento con el paso del tiempo hasta considerarse un riesgo importante. Esto es debido a que el éxito de los ataques de DoS está directamente relacionado con la exposición de servicios de las entidades a Internet: a medida que los servicios son más importantes y los usuarios dependen más de ellos, el riesgo de su indisponibilidad es más crítico y los ataques de DoS se vuelven más atractivos para los atacantes.
13. Por ello informes que tratan los principales riesgos de una entidad ya incluyen la denegación de servicio en los rankings de riesgos e impactos. Seguidamente se exponen los resultados del último análisis del instituto Ponemon [2] sobre los principales riesgos que gestionan actualmente diversas entidades.
14. El impacto asociado a una denegación de servicio puede tener una gran repercusión en la Organización:

- Pérdidas económicas: si por ejemplo el servicio atacado es una tienda online o un banco, su actividad económica se verá considerablemente afectada el tiempo que dure la denegación de servicio pudiendo causarles pérdidas millonarias.
- Pérdidas de imagen: la reputación de un servicio atacado puede quedar dañada al parecer que es un sitio inseguro o que tiene un mal funcionamiento.
- Multas por incumplimiento de SLA (*Service Level Agreement*): en caso de servicios externalizados cuya disponibilidad queda fijada por un SLA, un ataque de DoS puede implicar su incumplimiento.

5. CLASIFICACIÓN DEL PROBLEMA

5.1 TAXONOMÍA

15. A la hora de afrontar un ataque de DoS surge un problema básico por la dificultad de entender el objetivo, la fuente origen, el impacto, el método de propagación, etc. Para resolver el problema es necesario entender este tipo de ataques en profundidad ya que una denegación de servicio puede afectar a diferentes niveles y de distintas formas. Es muy común encontrarse con una limitada visión de este tipo de ataques, esta suele enfocarse principalmente a:
    - DoS
    - DDoS
    - Botnets
16. Por estas razones se hace esencial disponer de una taxonomía que permita identificar sobre qué activos puede focalizarse un ataque de DoS, visualizar qué modos y en qué capas se puede mitigar y las diferentes técnicas por las cuales se puede materializar. Una taxonomía puede servir de orientación para definir un plan eficiente que trate cualquier tipo de ataque de DoS de una forma detallada y precisa.
17. A continuación se propone una taxonomía que busca una clasificación lo suficientemente completa que cubre todos los ataques de DoS conocidos hasta la fecha.

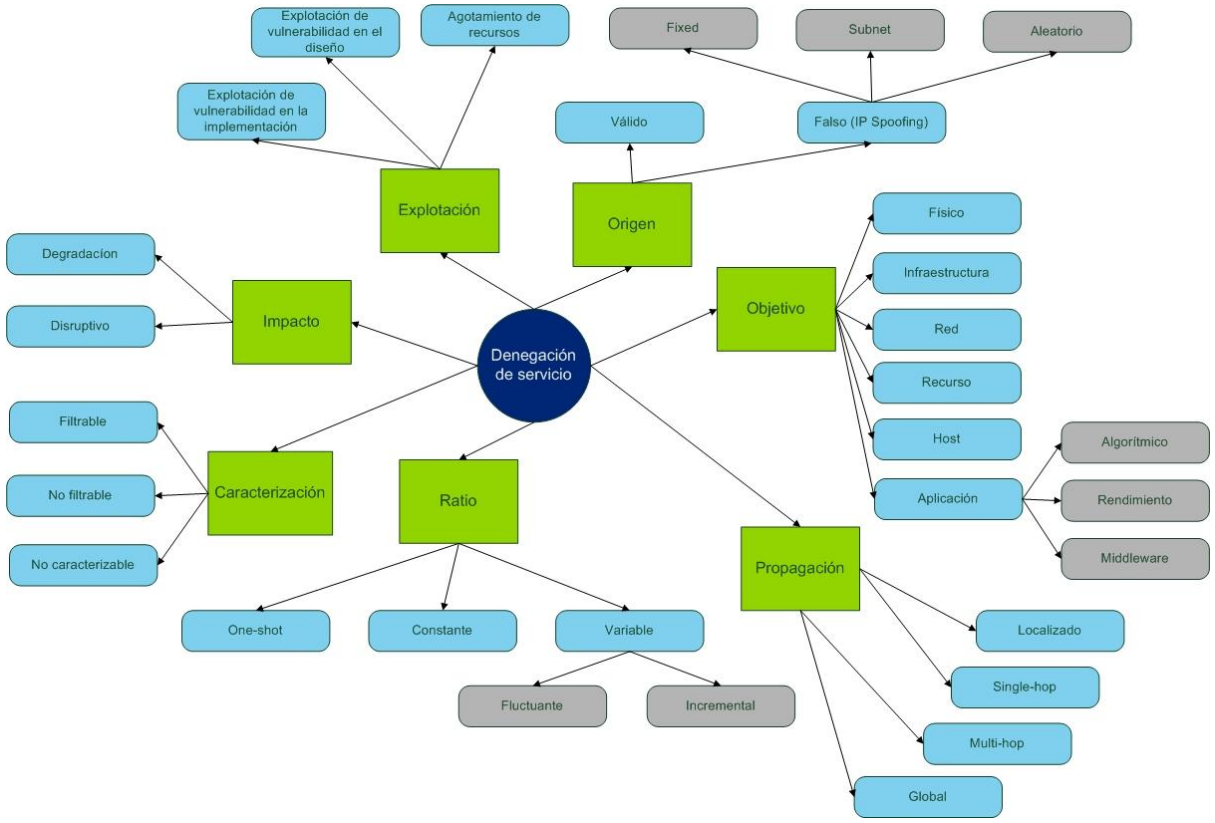


Figura 1 - Taxonomía propuesta

18. Una taxonomía genérica permite:
    - Clasificar los distintos tipos de ataques de DoS.
    - Entender los diferentes aspectos de un ataque de DoS.
    - Desarrollar un plan de reacción y valorar las posibles mitigaciones.
    - Valorar cómo estos activos pueden ser víctimas de un ataque de DoS.
    - Detectar los posibles vectores de ataque y sus mitigaciones.

5.2 VECTORES DE TAXONOMIZACIÓN

5.2.1 ORIGEN DEL ATAQUE

19. El origen de un ataque de DoS se refiere a las direcciones IP que originan el tráfico que tiene como objetivo causar la denegación de servicio. Estas direcciones IP pueden ser válidas y por lo tanto se puede identificar el origen del ataque o pueden haber sido falseadas mediante técnicas *IP Spoofing*.

20. *IP Spoofing* se aprovecha de que el campo de la cabecera IP que hace referencia a la dirección IP origen puede ser alterado. Esto suele realizarse utilizando herramientas que permiten generar paquetes (por ejemplo: scapy, hping3, nmap, etc) consiguiendo así falsear el origen del ataque o ampliar o reflectar el tráfico. Es muy común hacerse la pregunta de ¿por qué existe esta capacidad en la pila TCP/IP? Hay razones legítimas para alterar paquetes TCP/IP y trasmitirlos a la red. Por ejemplo, las redes VPN (redes que proporcionan acceso a una red local a un usuario de una red remota) sustituyen la IP del host remoto por una IP de la red local. También en entornos móviles IP, un host

roaming (capacidad de realizar o recibir llamadas en una red móvil que se encuentra fuera del área de servicio local de la compañía) tiene que usar una dirección IP local en una red extranjera.

21. Hay varios tipos de s*poofing*:

    - **Generación de direcciones IP aleatorias:** A la víctima le llegan paquetes de direcciones aleatorias y aparentemente falsas. El host atacante genera direcciones IP aleatorias. Esta estrategia genera direcciones IP inválidas, como por ejemplo direcciones del rango 192.168.0.0 (reservadas para ser utilizadas en redes locales), direcciones *broadcast*, direcciones no enrutables y direcciones no válidas (0.2.41.3), las cuales pueden causar a los routers problemas significativos. Sin embargo, la mayoría de las direcciones IP generadas serán válidas y enrutables.
    - ***Spoofing* en una subnet:** A la víctima le llegan paquetes de subredes identificables. En una red que utiliza el rango 192.168.1.0/24, es relativamente fácil falsear la dirección IP de un vecino a no ser que el administrador de la red haya tomado las medidas necesarias para evitarlo.
    - ***Fixed***: A la víctima le llegan paquetes de direcciones falsas fácilmente identificables. Un atacante que desee realizar un ataque de reflexión o que quiera echar la culpa del ataque a otras máquinas utilizará este tipo de *IP Spoofing*.

22. La defensa más común contra *IP Spoofing* es el *ingress/egress flitering*. Esto es un conjunto de reglas que se aconseja aplicar en cualquier router que comunique dos o más redes distintas en una organización. Algunos ISP (Proveedor de Servicios de Internet) también aplican estas reglas para evitar la generación de direcciones IP aleatorias por parte de usuarios mal intencionados o para evitar que un atacante pueda sustituir su dirección IP por una dirección interna de la red local.

23. Por ejemplo, en la siguiente figura se muestra cómo el tráfico que genera un atacante cuyo rango de direcciones IP es 204.67.209.0/24 será filtrado por el "Router 2" de su ISP en caso de que cambie la dirección IP de origen por una que no pertenezca a dicho rango. Esta técnica se llama ***Egress filtering*** refiriéndose al tráfico de salida del "Router 1".

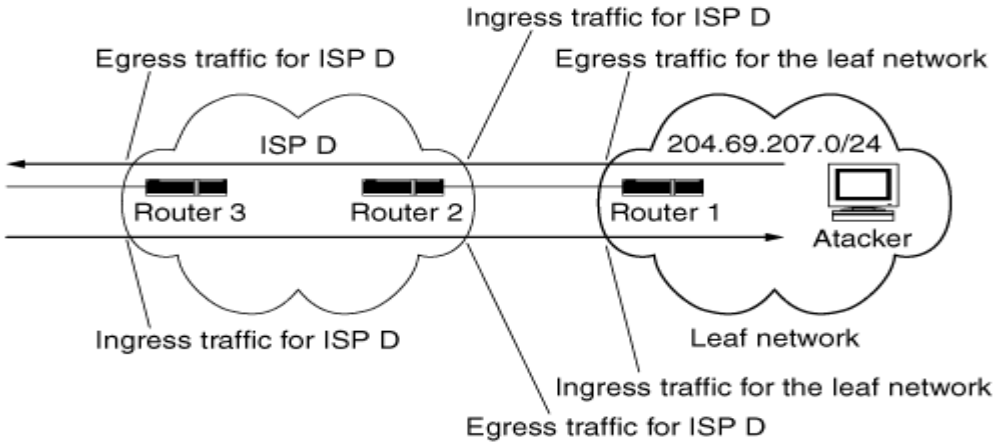


Figura 2 - Ingress/egress filtering example from RFC 2827

24. ***Ingress filtering*** se refiere al tráfico que llega al ISP y posteriormente a la red local del atacante, en este caso el objetivo del filtrado es evitar que otro potencial atacante (con origen en Internet) pueda sustituir su dirección IP por una dirección del rango 204.69.207.0/24.
25. Hay que tener en cuenta que determinados ataques de denegación de servicio distribuidos que son llevados a cabo por un gran número de máquinas. En estos casos el *IP Spoofing* es irrelevante, por lo tanto tomar medidas en estos casos no servirá de nada.

5.2.2 OBJETIVO DEL ATAQUE

26. En un ataque de denegación de servicio los objetivos pueden ser variados. Éstos pueden diferenciarse en aplicación, host, recurso, físico, infraestructura y red. A continuación se procederá a explicar detalladamente cada uno de ellos.

5.2.2.1 FÍSICO

27. El objetivo del atacante es dañar física o permanentemente un activo de forma que el servicio quede caído o dañado. Es posible efectuar un ataque de DoS contra el hardware de una máquina, como por ejemplo la CPU, memoria RAM, etc.
28. Una denegación permanente (PDoS) se debe tener muy en cuenta ya que puede provocar grandes pérdidas a la Organización que lo sufra debido a que su solución no es posible vía software. Además este tipo de ataques no requieren un gran esfuerzo para ser llevados a cabo. Por ejemplo, un atacante, de manera remota, puede forzar la actualización del firmware a una serie de dispositivos conectados en red de tal manera que cuando se produzca la actualización del dispositivo se produzca un daño irreparable que sólo puede ser solucionado si se sustituyen ciertas piezas del hardware.
29. También es posible perpetrar una denegación de servicio explotando una vulnerabilidad en el hardware de un dispositivo de red.
30. Como contramedidas es aconsejable reducir la superficie del ataque a la mínima expresión segregando las comunicaciones mediante un total aislamiento o uso de *proxys* y no exponiendo aquellos servicios que no sean estrictamente necesarios.

5.2.2.2 INFRAESTRUCTURA

31. En este caso se realiza un ataque que actúa negativamente sobre un activo o protocolo de forma que el servicio deja de estar disponible. Es posible hacer una diferenciación dentro de estos ataques:

    - **Ataques contra la infraestructura de Internet:** en este tipo de ataques los objetivos son muy atractivos para los atacantes por el impacto que pueden tener pudiendo afectar a un gran número de usuarios. Normalmente, los ataques que se han producido en la infraestructura de Internet han sido dirigidos hacia servicios críticos, como servidores DNS, routers, protocolos de routing, etc.
    - **Ataques contra la infraestructura de una red local:** el procedimiento en este tipo de ataques es muy parecido al caso anterior, con la diferencia de que puedan ser llevados a cabo con menor dificultad y por un solo atacante. El objetivo de este tipo de ataques suelen ser los switches o hubs, servidores

proxy, etc. Al tratarse de una red local es posible llevar a cabo ataques específicos contra los switches de estas redes como el ARP *Spoofing* o MAC *Flooding*.

5.2.2.3 RED

32. En este tipo de ataques la capacidad de la red se ve afectada debido a un gran uso del ancho de banda.
33. Se logra enviando un gran número de paquetes del tipo que sea, lo más rápido posible y cuyo destino es un servicio de la máquina objetivo, el cual se ve incapaz de procesar los paquetes al consumirse todo el ancho de banda. La víctima puede verse incapaz de defenderse por sí sola ante este tipo de ataques y pedir ayuda a su ISP o a servicios especializados, ya que suelen enviarse paquetes desde multitud de máquinas a la vez (DDoS).
34. El tráfico generado por las máquinas atacantes puede llegar a ser tan grande que como consecuencia el ISP de la víctima también se verá afectado. En ocasiones el tráfico es fácil de identificar y de filtrar (paquetes UDP largos cuyo destino son puertos inusuales), pero otras veces resulta más complicado, como ocurre con algunos ataques de amplificación.

5.2.2.4 RECURSO

35. En esto caso se ataca directa o indirectamente al recurso del que depende un servicio. Un recurso crítico del que dependen muchos servicios es el servidor DNS, lo que le convierte en uno de los objetivos más comunes de este tipo de ataques. También puede ser objetivo de este tipo de ataques la capacidad de enrutamiento de un router, un enlace que soporta mucha carga o un servicio de autenticación
36. Un ejemplo real de un ataque de este tipo ocurrió en 2001, siendo la víctima los servicios online de Microsoft. La causa fue que todos los servidores DNS estaban en el mismo segmento de red y eran servidos por el mismo router. Los atacantes aprovecharon para atacar la infraestructura de este router causando la caída de los servicios. Para evitar esto es recomendable añadir redundancia a la red reforzando la topología para que existan caminos alternativos y en caso de caída de un router no se vean afectados los servicios.

5.2.2.5 HOST

37. Un ataque al host deshabilita al objetivo el acceso a la red sobrecargando o deshabilitando su mecanismo de comunicaciones o haciendo que el host se caiga, se cuelgue o se reinicie.

5.2.2.6 APLICACIÓN

38. En este tipo entran todos los ataques cuyo objetivo es la aplicación que corre en un servicio. Suele ser un objetivo común del atacante ya que no es usual proteger la aplicación contra una denegación de servicio a pesar de que el impacto a este nivel es muy considerable.

39. En este nivel se suele englobar cualquier software por encima del sistema operativo, como por ejemplo una base de datos, servidor web, servidor de aplicaciones o librerías y aplicaciones.
40. El resultado de un ataque de DoS a estos objetivos puede ser muy variado: puede ser temporal y de recuperación automática; puede ser temporal pero ser necesaria la intervención humana o incluso es posible que se produzca la destrucción lógica de un activo.
41. La falta de protección se suele deber a un diseño ineficiente, un error en el código, un exceso en el uso de contenidos dinámicos o un *backend* no preparado para soportar una carga alta. Estos ataques se suelen clasificar dentro de tres categorías:
    - Rendimiento
    - Algorítmico
    - Middleware

5.2.2.6.1 RENDIMIENTO

42. El modo de actuar de un atacante en este tipo de ataques es enviar paquetes hasta que se supere el límite que tiene la aplicación para manejar peticiones o explotar una vulnerabilidad de la aplicación.
43. Por ejemplo, los servidores web tardan una cantidad de tiempo en responder a una petición (GET, POST, etc), por lo tanto, existe un límite en el número de peticiones por segundo que pueden mantener. Asumiendo que un servidor web es capaz de procesar 1000 peticiones por segundo, si un atacante envía 1001 peticiones en un segundo causará una denegación de servicio. Esto es posible si un atacante controla por ejemplo 10000 equipos, en este caso sería capaz de enviar más de 1000 peticiones en un segundo superando así la capacidad del servidor.

5.2.2.6.2 ALGORÍTMICO

44. Se ataca una deficiencia en una estructura de datos o a la incorrecta implementación de un algoritmo. Los árboles binarios y las tablas hash pueden degenerar en su rendimiento si se elige una entrada adecuada:
    - **Árboles binarios no balanceados:** el orden de crecimiento de este algoritmo para insertar "n" elementos será de O(nlogn), pero si los elementos ya han sido ordenados de antemano, entonces el árbol podría degenerar a una "*linked list*" y tener un orden de $O(n^2)$.
    - **Tablas Hash:** en una tabla hash se espera un orden de O(n) para insertar "n" elementos. Sin embargo, si el hash de cada elemento a insertar coincide correspondiéndole una misma "celda" del array (colisión) la tabla hash degenerará en una "*linked list*" de orden $O(n^2)$.

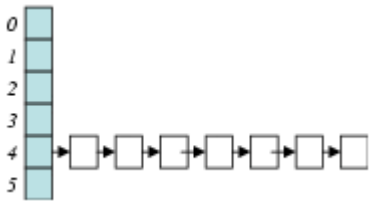


Figura 3 - Colisión en tabla hash

45. En definitiva, un atacante puede provocar una denegación de servicio sin enviar demasiados paquetes a la víctima si puede controlar y predecir las entradas usadas por estos algoritmos.

5.2.2.6.3 MIDDLEWARE

46. En este caso el atacante puede utilizar debilidades de ejecución o programación que puedan ser usados para impactar en la disponibilidad del servicio.

5.2.3 PROPAGACIÓN DEL ATAQUE

47. Forma en la que se propaga el ataque dentro de una red, cuanto más se propague más sistemas se verán afectados y más difícil será su mitigación.
48. Según el alcance de la propagación es posible hacer cuatro diferenciaciones:
    - **Localizado:** el ataque no supera una determinada capa como el servidor web o el firewall.
    - ***Single-hop*:** el ataque afecta a los servidores de aplicaciones.
    - ***Multi-hop*:** el ataque afecta a la base de datos, DNS y otros servicios.
    - **Global:** la red queda inoperativa (por ejemplo, cae el *backend* de autenticación)
49. En este tipo de ataques se puede diferenciar tres modos de propagación:
    - **Propagación manual:** requieren la intervención humana.
    - **Propagación semiautomática:** requieren la intervención humana en alguna de sus fases.
    - **Propagación automática:** no requieren intervención humana.
50. Centrándose en el modo de propagación automática es posible diferenciar tres técnicas de propagación llevadas a cabo por "gusanos" en un ataque DDoS:
    - **Propagación utilizando un servidor central:** este mecanismo se usa para comprometer sistemas de manera que el servidor central transfiere una copia de las herramientas al nuevo sistema comprometido para llevar a cabo el ataque. Si el servidor central deja de funcionar el ataque se detiene. Un ejemplo de gusano que utiliza este tipo de propagación es el "1i0n".

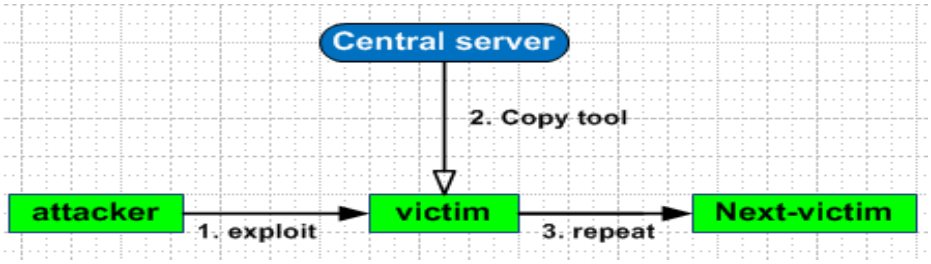


Figura 4 - Propagación utilizando un servidor central

- **Propagación en cadena:** este mecanismo se utiliza para comprometer sistemas, en este caso, el host del atacante transfiere una copia de las herramientas para llevar a cabo el ataque a sistemas comprometidos, éstos a su vez, serán capaces de realizar nuevas transferencias a otros sistemas vulnerables. Para que esta técnica funcione es necesario que las herramientas de ataque contengan algún método para establecer conexiones y enviar ficheros al nuevo sistema comprometido. La ventaja de esta técnica respecto a la anterior es que no existe una dependencia con el servidor central y por lo tanto hay más posibilidades de que el ataque perdure en el tiempo. Un ejemplo de gusano que utiliza esta técnica es el llamado "ramen".

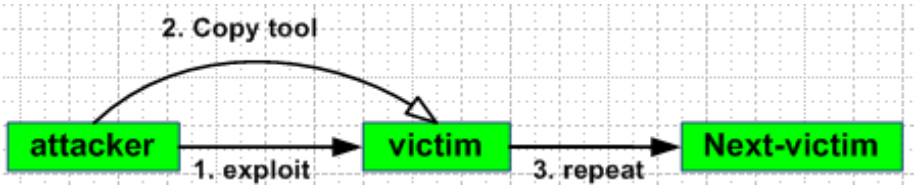


Figura 5 - Propagación en cadena

- **Propagación autónoma:** esta técnica incluye inyecciones de comandos directamente en el host de la víctima, como consecuencia el ciclo del ataque se inicia sin la necesidad de transferir ningún archivo desde una fuente externa. Un ejemplo de gusano es "Code Red".

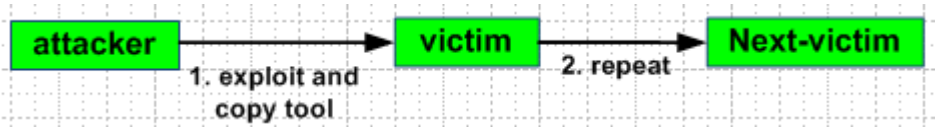


Figura 6 - Propagación autónoma

5.2.4 FRECUENCIA E INTENISDAD DEL ATAQUE (RATIO)

51. La frecuencia de un ataque de DoS puede variar y por lo tanto complicar su detección. Es posible hacer una clasificación de un ataque de DoS basándose es su frecuencia:

    - **One-shot:** con muy pocos bytes se consigue detener el servicio, como por ejemplo un ataque de complejidad algorítmica, un exploit de DoS o un formulario muy mal programado.
    - **Constante:** se observa un flujo constante de interacción, un gran número de los ataques conocidos envían un flujo constante de tráfico, por ejemplo el flujo generado por una red de botnets.

- **Variable fluctuante:** el flujo varía en el tiempo produciéndose picos en la frecuencia de envío, esta sofisticación provoca que este tipo de ataques sean muy difíciles de detectar, por ejemplo un ataque efectuado en horas punta.
- **Variable incremental:** a medida que pasa el tiempo el ataque va en aumento. Como consecuencia el servicio de la víctima se va degradando lentamente lo que provoca que sea también difícil de detectar.

5.2.5 CARACTERIZACIÓN

52. Llevando a cabo un análisis del tráfico es posible hacer la siguiente clasificación:

- **Filtrable:** el ataque se puede identificar y filtrar. Los paquetes pueden ser filtrados fácilmente por un firewall.
- **No filtrable:** el ataque difícilmente se identifica ya que existen pocas diferencias respecto a tráfico legítimo. El tráfico no filtrable utiliza paquetes bien formados que hacen peticiones de servicios legítimos de la víctima. Un filtrado de estos paquetes podría causar un bloqueo de las peticiones hechas por usuarios legítimos al no existir una diferencia clara entre el contenido de los paquetes legítimos y los que no lo son. Un ejemplo es la inundación de peticiones HTTP a un servidor web o la inundación de peticiones DNS a un servidor DNS.
- **No caracterizable:** es imposible distinguir el tráfico ilegítimo del fiable. Generalmente este tipo de ataques intenta consumir todo el ancho de banda de la víctima utilizando una variedad de paquetes que se diferencian a nivel de protocolo o de aplicación. Un ataque que utilice una mezcla de paquetes TCP SYN, TCP ACK, ICMP ECHO, ICMP ECHO REPLY y paquetes UDP probablemente podría ser caracterizable, pero sólo después de un considerable periodo de tiempo y esfuerzo y si se tuviera acceso a una herramienta de caracterización sofisticada.

5.2.6 EXPLOTACIÓN

53. En la actualidad se pueden diferenciar tres métodos de explotación en una denegación de servicio:

- **Ataque por agotamiento de los recursos de la víctima o fuerza bruta:** este tipo de ataques se pueden considerar de fuerza bruta y no tienen gran complejidad. Simplemente se trata de enviar un volumen de tráfico lo suficientemente grande como para que la víctima no lo pueda manejar. Suelen dirigirse al enlace que existe entre el servicio de la víctima y su ISP. Cuando esto ocurre, el enlace se satura o la interfaz de red deja de funcionar correctamente impidiendo así el acceso al servicio del resto de usuarios legítimos.
- **Explotación de una debilidad de diseño de un protocolo o algoritmo:** los ataques de este tipo explotan una determinada debilidad en el diseño de los protocolos. Esta debilidad, puede darse en protocolos tan comúnmente usados en Internet como SSL/TLS o incluso TCP. En estos casos el volumen de tráfico es menor que en el ataque por fuerza bruta. Un ejemplo de este tipo de explotación son los ataques de denegación de servicio que aprovechan el elevado tiempo de negociación entre el cliente y el servidor en el protocolo

SSL/TLS. Cuando un cliente negocia una conexión con un servidor que implementa estos protocolos, el servidor requiere 10 veces más capacidad de procesamiento que el cliente, para ser más exactos, un cliente puede aguantar entre 150 y 300 negociaciones por segundo mientras que un servidor puede aguantar hasta 1000. De este modo, un atacante puede aprovecharlo para renegociar nuevas conexiones con el objetivo de causar progresivamente un mal rendimiento del servidor. Hay que tener en cuenta que este ataque sólo funciona contra servidores que tienen activados la característica de renegociación SSL/TLS. El problema es que existen varias implementaciones de este protocolo que no permiten desactivar esta opción, como es el caso de OpenSSL. También hay casos en los que es necesario utilizar la renegociación, como ocurre con conexiones de larga duración.

- **Explotación de una debilidad en la implementación de un protocolo o tecnología:** a diferencia de los ataques que aprovechan debilidades en el diseño de protocolos, este tipo de ataques suelen aprovechar debilidades en aplicaciones, implementaciones de protocolo, contramedidas, etc. El volumen de tráfico es pequeño, los paquetes que forman este tráfico suelen ser construidos manualmente. Al ser fallos en la implementación generalmente es posible corregirlos o mitigarlos de manera que un ataque de DoS no tenga éxito.

6. BÚSQUEDA DE UNA CONTRAMEDIDA

54. A la hora de implantar una contramedida para proteger un activo se debe realizar una valoración de las siguientes características:

    - **Efectividad:** en primer lugar se debe estudiar cómo de efectiva es una contramedida ante un ataque de denegación de servicio y de cómo su implementación servirá para mitigarlo. Pueden darse casos en los cuales el porcentaje de tráfico malicioso que detiene una contramedida es tan bajo que no se justifica su implantación siendo más conveniente invertir esfuerzos en otras contramedidas más efectivas.
    - **Fiabilidad:** una contramedida debe poder manejar cualquier tipo de ataque por la cual se ha diseñado además de evitar falsos positivos ya que no convendría que una contramedida afectase o dañase el tráfico legítimo de la red. Por ejemplo, puede darse el caso de que se detecte un ataque de DoS cuando en realidad no se está produciendo. Esto puede ser muy perjudicial para la organización no sólo porque se bloqueará tráfico legítimo, sino también porque si este hecho ocurre con frecuencia un ataque real podría no ser tomado en serio.
    - **Daño colateral:** es importante analizar los efectos que puede tener la implantación de una contramedida sobre el tráfico legítimo. Se entiende como daño colateral cuando se bloquea el tráfico legítimo entrante de un cliente o cuando los efectos del ataque lo sufren servicios o máquinas que no eran el objetivo principal del ataque. Distinguir el tráfico legitimo del tráfico malicioso puede ser una tarea muy complicada por varias razones, ya que los atacantes suelen camuflar el tráfico malicioso entre el tráfico legítimo utilizando ambos tipos de tráfico un mismo enlace, un mismo protocolo o un mismo puerto destino. Por ejemplo, podría darse el caso de que como contramedida se instalase un balanceador de carga que balancease el tráfico

entrante entre dos o más servidores. Si un atacante consiguiese explotar una vulnerabilidad en este balanceador se produciría una denegación de servicio ya que el tráfico no llegaría a los servidores.

- **Completitud:** cuando se implementa una contramedida se debe tener en cuenta la posibilidad de que por sí sola no sea capaz de resolver el problema y que sea necesario mecanismos de defensa complementarios. Por ejemplo, si se implanta una contramedida de detección será necesario implementar también una medida de reacción, ya que la primera no servirá para detener el ataque.
- **Tiempo de respuesta:** en el caso de que una contramedida sea un mecanismo de reacción, la respuesta deberá ser lo suficientemente rápida como para asegurar que el objetivo del ataque no sufra una interrupción en el servicio.
- **Facilidad de implementación:** se debe tener en cuenta que puede darse el caso en el que una contramedida no aporte el suficiente beneficio como para ser llevada a cabo. Es significativo que si un mecanismo de defensa supone un coste económico mayor que la pérdida económica o de imagen producida por un potencial ataque, no es recomendable implantarlo. También hay que apreciar el coste de administración, ya que un mecanismo de defensa que se base en la detección de firmas o patrones debe ser actualizado periódicamente.
- **Lugar de instalación:** es necesario entender a qué nivel se está implementando la contramedida, por ejemplo, un firewall puede ayudar a filtrar el tráfico a nivel de red e incluso a nivel de transporte, pudiendo evitar ataques de tipo ICMP Flood, pero no servirá para filtrar el tráfico a nivel de aplicación. Por otro lado, el lugar de instalación también se refiere a la proximidad del mecanismo de seguridad con respecto al objetivo:
    - **Instalación próxima al objetivo:** esta es la localización más común. Se puede pensar que la mejor protección de cara a este tipo de ataques se debe focalizar en el entorno de red próximo al objetivo y en parte es cierto ya que es el objetivo el que recibe todo el tráfico malicioso. Por lo tanto, tras un análisis del ataque será más fácil obtener las características del mismo para poder reaccionar en su contra de una manera efectiva. Estas defensas podrían estar en la propia máquina objetivo, o pueden estar relativamente próximas, como es el caso de routers, firewalls o dispositivos IDS. Otra ventaja es que en el caso de que se trate de un falso positivo será más sencillo desactivar el mecanismo de seguridad.

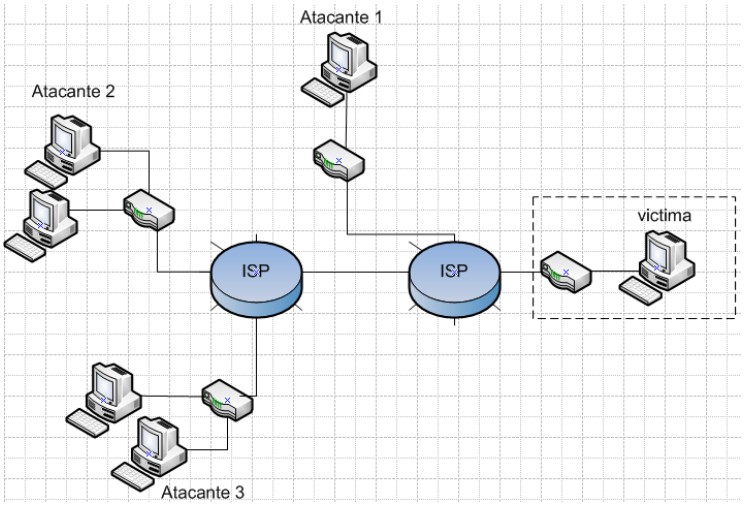


Figura 7 - Instalación próxima al objetivo

Existen desventajas respecto a la instalación próxima al objetivo, sobre todo en lo que a ataques de DDoS se refiere. El volumen de estos ataques puede llegar a ser tan grande que los mecanismos de defesa pueden verse incapaces de manejar el tráfico.

- **Instalación próxima al atacante:** implementar mecanismos en la red del atacante puede ser una buena opción, esto es lo que hacen determinadas herramientas Anti-DoS como las REDES DE ENTREGA DE CONTENIDO que lo que intentan es bloquear el origen del ataque aproximándose a él. Hay que tener en cuenta que puede ser difícil localizar el origen, sobretodo en ataques de DDoS en los cuales el tráfico malicioso puede tener direcciones IP orígenes distintas o direcciones IP falseadas.

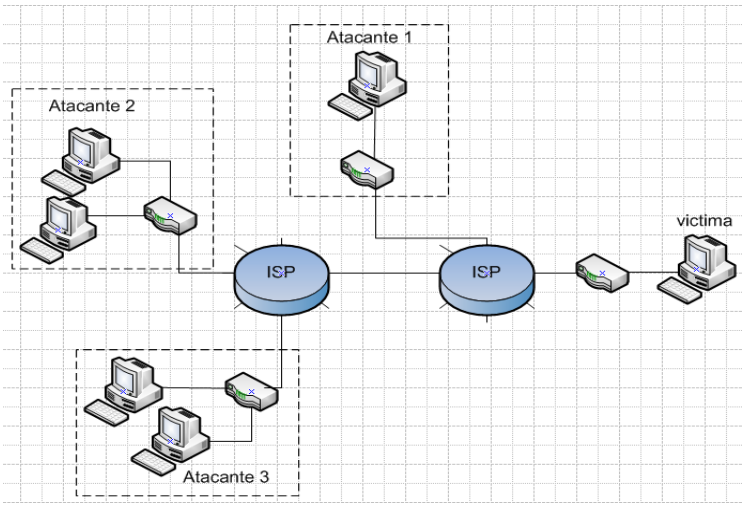


Figura 8 - Instalación próxima al atacante

- **Instalación en el ISP:** en este caso se requiere llegar a un acuerdo con el proveedor de servicio de Internet. Muchos proveedores ofrecen a sus clientes la posibilidad de contratar un segundo enlace de

comunicación o más ancho de banda. Esto puede ayudar a mitigar algunos ataques de denegación de servicio cuyo mecanismo de explotación es la inundación. El problema de esta localización es que los routers de los ISP suelen manejar una carga de tráfico muy elevada y se pueden llegan a saturar durante un ataque con gran volumen de tráfico. Además, al igual que las contramedidas próximas al atacante no sirven de nada contra ataques de DoS originados en la red local de la organización.

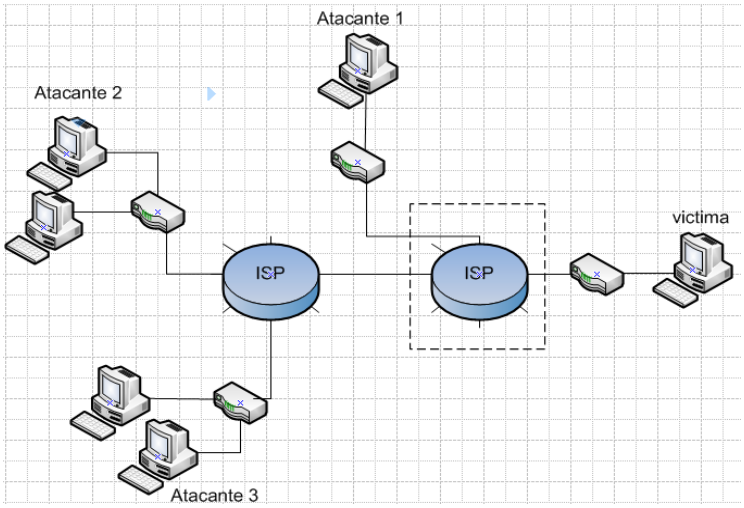


Figura 9 - Instalación en el ISP

7. CONTRAMEDIDAS

55. Según el Real Decreto 3/2010 - Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica [1] se deben aplicar medidas frente a ataques de denegación de servicio en los sistemas de información categorizados con nivel MEDIO o ALTO.

DIMENSIONES		
Bajo	Medio	Alto
No aplica	Aplica	Aplica+

Tabla 1 – Aplicación de medidas

56. El Real Decreto 3/2010 también especifica qué tipos de medidas deben aplicarse según la categoría del activo:

- **Nivel MEDIO**: Se establecerán medidas preventivas y reactivas frente a ataques de denegación de servicio (DoS, Denial of Service). Para ello:
  - Se planificará y dotará al sistema de capacidad suficiente para atender a la carga prevista con holgura.
  - Se desplegarán tecnologías para prevenir los ataques conocidos.
- **Nivel ALTO:**

- Se establecerá un sistema de detección de ataques de denegación de servicio.
- Se establecerán procedimientos de reacción a los ataques, incluyendo la comunicación con el proveedor de comunicaciones.
- Se impedirá el lanzamiento de ataques desde las propias instalaciones perjudicando a terceros.

57. A su vez la guía del Esquema Nacional de Seguridad (ESN) CCN-STIC-803 – Valoración de Sistemas [3] a la hora de evaluar la disponibilidad de un servicio pone como referencias de Requisitos de disponibilidad de un servicio (RTO, *Recovery Time Objective*) los siguientes tiempos:

RTO	< 4horas	4horas – 1día	1día – 5días	> 5días
Nivel	ALTO	MEDIO	BAJO	Sin valorar

Tabla 2 – Requisitos de disponibilidad

58. Teniendo en cuenta estas medidas y tiempos de referencia se han agrupado las contramedidas para mitigar ataques de denegación de servicio entre las que se han de aplicar a sistemas de información clasificados de nivel MEDIO y ALTO. Para los clasificados como nivel BAJO no se incluyen contramedidas ya que en el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica no las contempla.

7.1 CONTRAMEDIDAS PARA SISTEMAS DE INFORMACIÓN CATEGORIZADOS CON NIVEL MEDIO

7.1.1 CONFIGURACIONES ESPECÍFICAS PARA PREVENIR ATAQUES CONOCIDOS

59. A continuación se describe una serie de ataques y contramedidas específicas contra los principales ataques de denegación de servicio.

7.1.1.1 SLOWHTTP

60. Los ataques de peticiones HTTP lentas (*SlowHTTP*) buscan consumir los recursos de un servidor web aprovechándose de debilidades en el diseño del protocolo HTTP. Tienen como objetivo mantener ocupado un servidor empleando los recursos mínimos por parte del atacante. En vez de inundar el servidor con peticiones se intenta mantener las conexiones abiertas el mayor tiempo posible. Para lograrlo el atacante envía peticiones HTTP parciales y continúa lentamente enviando subsecuencias de la petición para evitar que se cierre el socket. El ataque intenta ocupar todos los sockets del servidor impidiendo que usuarios legítimos puedan realizar peticiones.
61. Existen variantes del ataque ***SlowHTTP*** que no se centran en enviar muy lentamente la cabecera de la petición, dichas variantes son:

- ***Slow HTTP Post*:** las cabeceras de la petición son enviadas correctamente pero el cuerpo del mensaje es enviado muy lentamente.
- ***Slow Read*:** en vez de enviar lentamente datos al servidor, el cliente fuerza al servidor a enviar una gran cantidad de datos lo más lentamente posible declarando un ancho de ventana muy pequeño al negociar la conexión.

62. El propio ataque *SlowHTTP* y sus variantes afectan en mayor o menor medida los distintos servidores web. Sergey Shekyan, empleado de Qualys, publicó en el blog de Qualys Security Labs una entrada tratando cómo afectan los ataques *SlowHTTP* y *Slow HTTP Post* a los distintos servidores web [4]. En resumen la relación de qué servidores se ven afectador por qué ataques se puede observar en la siguiente tabla:

Servidor Web	SlowHTTP	SlowHTTP Post
Apache 2.2.19 MPM prefork	**X**	**X**
nginx 1.0.0	**X**	**X**
Lighttpd 1.4.28	**X**	**X**
IIS 7		**X**

Tabla 3 - Relación de servidores afectados por ataques *SlowHTTP*

63. Se aprecia que el servidor ISS 7 no es vulnerable a ataques *SlowHTTP*, esto es debido a que hasta que no se recibe por completo la cabecera sólo reserva un mínimo de recursos (no pone la conexión en modo escritura) permitiendo soportar un amplio número de "conexiones pendientes". En cambio sí es vulnerable a ataques *SlowHTTP* Post ya que estos ataques se basan en enviar lentamente el cuerpo del mensaje, no la cabecera.

7.1.1.1.1 MITIGACIÓN

7.1.1.1.1.1 USO DE BALANCEADORES DE CARGA

64. Mediante el uso de balanceadores de carga configurados en modo *delayed binding* es posible impedir que las peticiones HTTP se entreguen a los servidores web hasta que los balanceadores hayan recibido la cabecera por completo evitando que los ataques *SlowHTTP* lleguen a los servidores web.
65. Se debe tener en cuenta que los ataques *Slow HTTP Post* sí envían la cabecera correctamente de modo que el uso de balanceadores no mitiga este tipo de ataques. Del mismo modo, esta medida tampoco es efectiva cuando se hace uso de HTTPS ya que en estos casos el balanceador entrega al servidor web directamente la petición.
66. A la hora de implantar esta medida de mitigación debe tenerse en cuenta que un fallo en el balanceador de carga puede impedir que tanto el tráfico legítimo como el

perteneciente al ataque dejen de llegar al servidor causando por lo tanto una denegación de servicio absoluta como daño colateral del fallo en el balanceador.

7.1.1.1.1.2 USO DE CORTAFUEGOS

67. Mediante el uso de cortafuegos, como iptables, es posible limitar el ratio de conexiones entrantes desde un mismo cliente logrando disminuir el impacto del ataque. En caso de especificar un límite de conexiones entrante demasiado bajo se corre el riesgo de impedir conexiones legítimas disminuyendo así la fiabilidad de la contramedida.
68. En los casos en los cuales los ataques provienen de varios clientes, como por ejemplo en un ataque de denegación de servicio distribuido, esta medida pierde eficacia.

7.1.1.1.1.3 MODIFICACIÓN DE LA CONFIGURACIÓN DE LOS SERVIDORES WEB

69. Es posible modificar la configuración por defecto de los servidores para que se vean afectados en menor medida por este tipo de ataques. Estos cambios en la configuración no proporcionan una efectividad total contra ataques de denegación de servicio pero sí aportan una protección suficiente como para justificar su implantación.
70. Se recomienda para mitigar ataques *SlowHTTP* y *Slow HTTP POST*:
    - Aumentar el número máximo de conexiones soportadas;
    - Limitar el tiempo de vida de las conexiones a un valor razonable;
    - Limitar el tamaño de las cabeceras y cuerpo de los mensajes a un tamaño razonable;
    - Denegar peticiones que tengan métodos HTTP no soportados (en principio métodos distintos a GET y POST).
71. Además, como protección ante ataques *Slow Read* se recomienda:
    - No aceptar conexiones con un tamaño de ventana extremadamente pequeño.
    - No habilitar conexiones persistentes y *HTTP pipelining* a no ser que exista un beneficio real a raíz de su activación.
72. También existen modificaciones específicas para cada uno de los servidores pero escapan al alcance de esta guía. Se recomienda la lectura del artículo "*How to Protect Against Slow HTTP Attacks*" [5] en el cual se incluyen recomendaciones de configuración para los servidores más populares.
73. Para comprobar la efectividad de la configuración del servidor web se puede hacer uso de la aplicación slowhttptest [6] que permite simular un ataque y medir el número de conexiones abiertas, cerradas y pendientes a lo largo del tiempo:

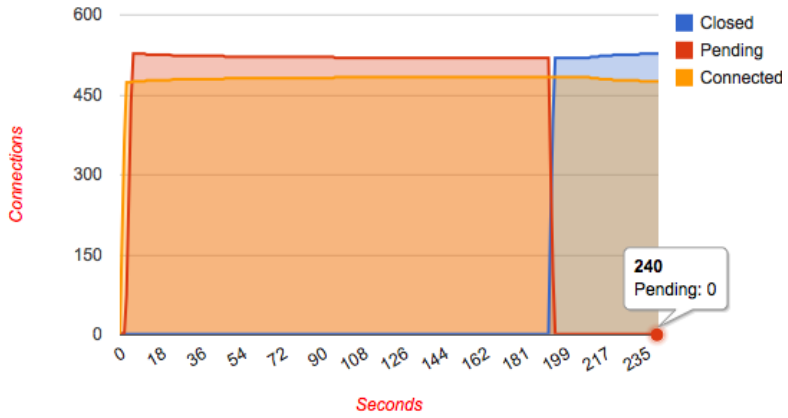


Figura 10 – Ejemplo de salida del slowhttptest

7.1.1.1.1.4 SCORING

74. Además de los métodos anteriormente descritos existen dispositivos que realizan un scoring de los clientes de manera que en caso de realizar determinadas acciones (por exceso o por defecto) bloquean las conexiones de los mismos. Estos parámetros se pueden ajustar en función de:

    - El número de peticiones que realizan los clientes, tanto por exceso como por defecto;

    - Las URLs a las que están tratando de acceder;

    - La cantidad de información que envían en dichas peticiones (tanto por exceso como por defecto).

75. Mediante el scoring es posible determinar cuando un cliente está realizando acciones dentro de los parámetros establecidos y bloquear aquellos clientes que se salgan de los patrones definidos.

7.1.1.2 TCP SYN FLOOD

76. Los ataques *TCP SYN Flood* se basan de una debilidad en el diseño del protocolo TCP para consumir los recursos de un servidor.

77. Para comprender este tipo de ataque se debe conocer cómo es el proceso de inicio de conexión del protocolo TCP: cuando se crea una conexión TCP, el cliente envía al servidor la solicitud de conexión (paquete SYN). En este momento el servidor reserva recursos para almacenar el estado de la conexión y responde al cliente con un paquete SYN+ACK.

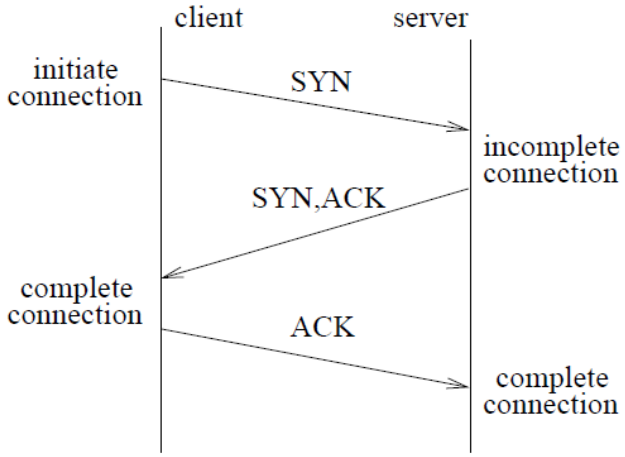


Figura 11 - Diagrama de inicio de una conexión TCP

78. El ataque *TCP SYN Flood* se basa en inundar al servidor de peticiones de inicio de conexión, no enviando los ACK correspondientes a los SYN+ACK enviados por el servidor, con el fin de agotar sus recursos.

7.1.1.2.1 MITIGACIÓN

79. Existen varias medidas para mitigar este tipo de ataques. En el RFC4987, *TCP SYN Flooding Attacks and Common Mitigations* [7] del IETF se recogen varias técnicas como son:

7.1.1.2.1.1 FILTRADO

80. Es posible filtrar, mediante el uso de cortafuegos, un número excesivo de intentos de inicio de conexión desde una misma dirección IP logrando así bloquear este ataque. Esta medida no es completamente efectiva ya que el atacante puede ocultar su dirección IP mediante técnicas de *IP Spoofing* haciendo parecer, de cara al cortafuegos, que cada paquete de inicio de conexión proviene de un origen distinto.

7.1.1.2.1.2 REDUCIR EL TIEMPO SYN-RECEIVED

81. Reduciendo el tiempo de espera desde que se envía el paquete SYN+ACK hasta que se recibe el ACK se logra reducir el impacto de este ataque pero se corre el riesgo de impedir conexiones de usuarios legítimos que tengan una conexión lenta por lo que es un valor que se debe adaptar a la latencia de los clientes del servidor.

7.1.1.2.1.3 PERMITIR SOBRESCRIBIR LAS CONEXIONES HALF-OPEN

82. Se considera una conexión en estado *HALF-OPEN* cuando el servidor ha respondido al SYN inicial pero no ha recibido el ACK confirmando la recepción del paquete SYN+ACK.

83. Permitiendo que los nuevos intentos de conexión sobrescriban las conexiones en estado *HALF-OPEN* se logra aceptar nuevas conexiones disminuyendo el impacto del ataque.

7.1.1.2.1.4 HABILITAR SYN CACHE

84. Por defecto, cuando un sistema recibe un paquete SYN reserva un espacio de memoria para almacenar el estado de la conexión. La estructura de datos usada para almacenar el estado se denomina *transmission control block* (TCB).

85. La técnica *SYN Cache* se basa en reducir la cantidad de información almacenada cuando se recibe el paquete SYN en vez de crear el TCB completo, permitiendo así soportar un mayor número de conexiones en estado *HALF-OPEN*.

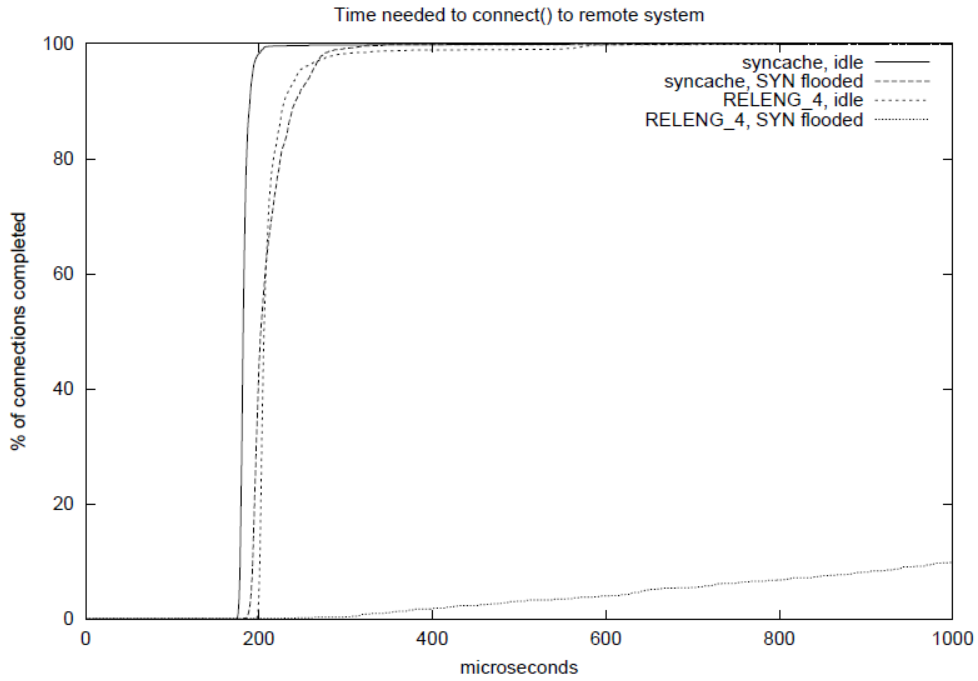


Figura 12 – Rendimiento de SYN Cache

86. Jonathan Lemon, desarrollador de FreeBSD, refleja en su estudio de la técnica "*SYN Cache Resisting SYN flood DoS attacks with a SYN cache"* [8] una comparativa de rendimiento con y sin *SYN Cache* activado. Como se aprecia en la Figura 12 el porcentaje de conexiones aceptadas bajo un ataque *SYN Flood* con *SYN Cache* activado es muy superior a cuando no está activado.

7.1.1.2.1.5 HABILITAR SYN COOKIES

87. Esta técnica lleva *SYN Cache* un paso más allá no creando el TCB hasta que ha finalizado el inicio de la conexión TCP.

88. La información que se guardaría en el TCB se codifica y se envía al cliente como número de secuencia en el SYN+ACK. De este modo cuando el servidor recibe el ACK que confirma el inicio de conexión es capaz de crear el TCB a partir del número de secuencia contenido en el paquete y continuar con la conexión.

89. Esta técnica logra soportar un mayor número de intentos de conexión que *SYN Cache* pero posee algunos inconvenientes derivados de que no se almacena el estado de la conexión al recibir el SYN inicial. Por ejemplo. en caso de perderse el paquete

SYN+ACK éste no podrá ser reenviado ya que el servidor no dispone de información necesaria para reconstruirlo y enviarlo nuevamente.

90. Una descripción más detallada de esta técnica se puede encontrar en el artículo de Dan Bernstein: *SYN Cookies* [9].

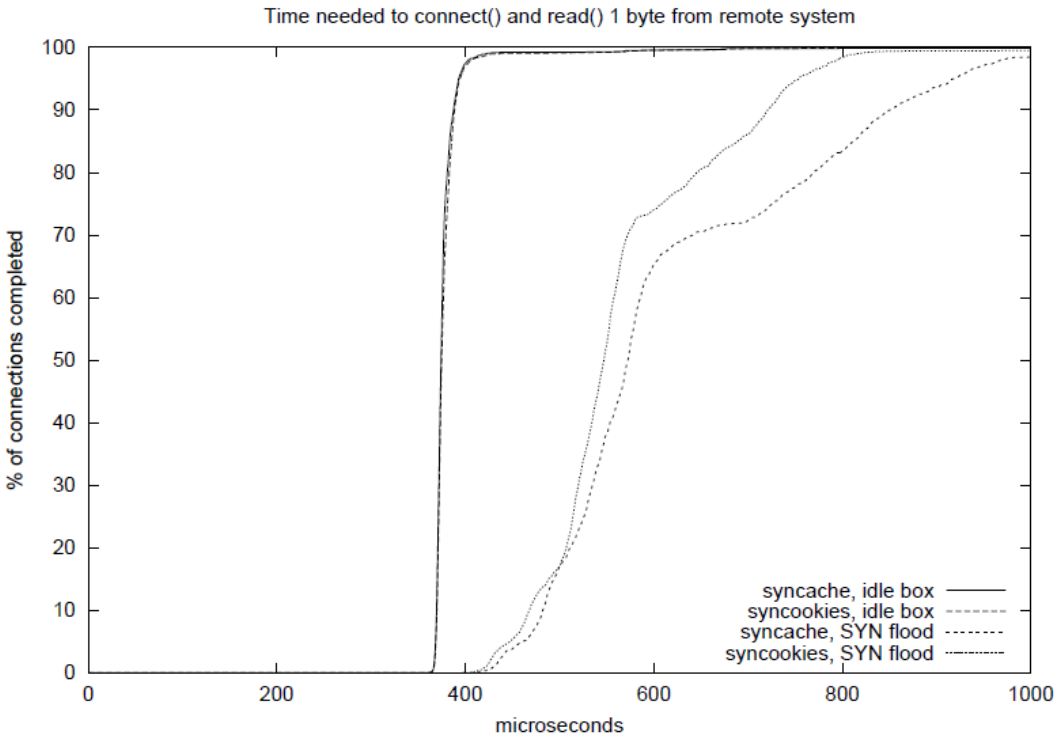


Figura 13 - Comparativa de rendimiento de SYN Cache + SYN Cookies

91. En la Figura 13 se aprecia cómo *SYN Cookies* mantiene un porcentaje mayor de conexiones completadas frente a *SYN Cache*.

7.1.1.2.1.6 SOLUCIONES HÍBRIDAS

92. Debido a que ninguna de las mitigaciones anteriormente comentadas contra este tipo de ataques tiene una completitud total, es posible y recomendable combinar las técnicas *SYN Cache* y *SYN Cookies* aprovechando las virtudes de ambas. Inicialmente se habilita *SYN Cache* hasta que se supera cierto número de conexiones; una vez superado dicho número se cambia a *SYN Cookies* logrando no saturar el servidor permitiendo que siga aceptando nuevas conexiones.

7.1.1.2.1.7 CATEGORIZACIÓN

93. Además de los métodos anteriormente descritos existen dispositivos que categorizan las direcciones IP que establecen conexiones a los servidores de manera que:

    - Se identifican aquellas direcciones IPs que realizan conexiones completas (cierran el *handshake*)

    - Se comportan de manera distinta en función del comportamiento de las direcciones cliente, realizando *SYN proxy* en función del número de conexiones cerradas / no cerradas.

- En caso de ataques masivos mantienen tablas paralelas de intentos de conexión de manera que no se llenen las tablas de estados.

7.1.1.3 UDP FLOOD

94. Para comprender este ataque se debe entender qué pasos sigue un sistema para procesar un paquete UDP entrante:
    - Recibe los datos que contienen el paquete UDP y lo desencapsula hasta llegar a la capa de transporte;
    - Obtiene el puerto destino del paquete UDP y comprueba si existe una aplicación escuchando en dicho puerto;
    - En caso de existir una aplicación escuchando en dicho puerto se lo entrega;
    - En caso de no existir una aplicación escuchando en dicho puerto el sistema responde con un paquete *ICMP Destination Unreachable.*

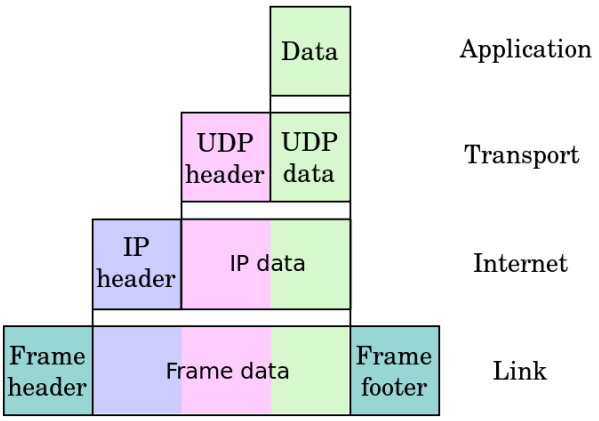


Figura 14 - Encapsulación UDP

95. El ataque *UDP Flood* se basa en inundar un equipo de paquetes UDP con un puerto destino aleatorio haciendo que por cada paquete, el equipo se sobrecargue al tener que comprobar si existe una aplicación escuchando en dicho puerto. Además el atacante puede falsear la dirección IP de origen del paquete UDP logrando ocultarse además de poder redirigir el paquete *ICMP Destination Unreachable* a otro equipo arbitrariamente.

7.1.1.3.1 MITIGACIÓN

96. Se recomienda:
    - Deshabilitar la respuesta de paquetes *ICMP Destination Unreachable*.
    - Limitar la visibilidad de los equipos mediante un firewall impidiendo que les lleguen paquetes UDP a excepción de los que vayan destinados a un servicio voluntariamente publicado.

7.1.1.4 SOCKSTRESS

97. *Sockstress* es un marco de trabajo compuesto por una serie de ataques orientados a agotar los recursos de los sistemas que aceptan conexiones TCP. Fue descubierto

inicialmente por Jack C. Louis, empleado de Outpost24, y presentado posteriormente junto a Robert E. Lee en la SEC-T Security Conference [10].

98. Los ataques que componen Sockstress tienen en común con el ataque *TCP SYN FLOOD* que saturan el objetivo iniciando conexiones TCP pero van un paso más allá: son capaces de establecer la conexión TCP implementando una técnica similar a *SYN Cookies* con el objetivo de reducir la carga que supone mantener el estado de las conexiones por parte del atacante y además son capaces de enviar *payloads* y modificar arbitrariamente los campos de los paquetes TCP con la intención de aumentar el consumo de recursos por parte de la víctima.

99. Algunos de los ataques que componen el marco de trabajo son:

    - **Conexiones con tamaño de ventana cero**: el atacante crea conexiones TCP especificando en el tercer paso del establecimiento de la conexión, junto al paquete ACK, un tamaño de ventana igual a 0 haciendo que la víctima tenga que mantener la conexión abierta enviando paquetes para comprobar el tamaño de ventana del atacante.
    - **Conexiones con tamaño de ventana excesivamente pequeño**: el atacante crea una conexión TCP especificando en el tercer paso del establecimiento de la conexión, junto al paquete ACK, un tamaño de ventana igual a 4 bytes y seguidamente envía un *payload* con los *flags* ACK/PSH. Si la víctima acepta el paquete con el *payload* responderá con trozos de 4 bytes consumiendo memoria y una conexión en el servidor. Este ataque está orientado a servicios que aceptan una petición inicial (contenida en el *payload*) a la cual contestan con una respuesta de gran tamaño, como puede ser un GET a una página web, o descargar un archivo.

7.1.1.4.1 MITIGACIÓN

100. Los ataques que componen el marco de trabajo de *Sockstress* requieren que se establezca la conexión TCP, por ello, a diferencia del ataque *TCP SYN FLOOD*, no se puede suplantar la dirección IP de origen del atacante mediante técnicas de *IP Spoofing*. Gracias a este requisito se puede mitigar este ataque limitando el número de conexiones aceptadas desde un mismo host.

101. Debido a las similitudes con el ataque de *TCP SYN FLOOD* las medidas para mitigar este ataque también son aplicables a los ataques *Sockstress*.

7.1.1.5 DNS AMPLIFICATION

102. Una petición DNS puede ocupar 64 bytes y generar una respuesta de tamaño hasta cincuenta veces superior. Esto, junto al falseo de la dirección IP de origen, es en lo que se basa el ataque DNS Amplification.

103. Los pasos del ataque son:

    - El atacante crea paquetes que contienen una consulta DNS especificando como dirección IP de origen la dirección de la víctima;
    - Envía las consultas DNS a servidores que aceptan consultas desde cualquier dirección IP;
    - Los servidores DNS envían la respuesta a la víctima saturando su conexión.

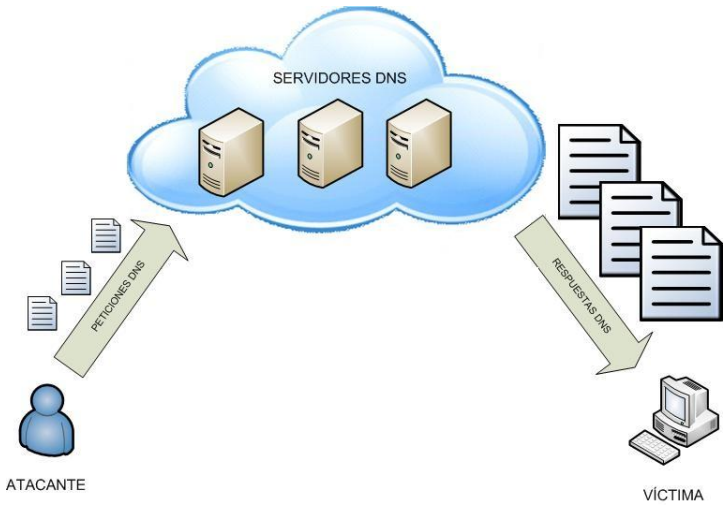


Figura 15 - Diagrama de ataque "DNS Amplification"

7.1.1.5.1 MITIGACIÓN

104. Debido a que este ataque consume el ancho de banda de la conexión de la víctima no existe una contramedida con una efectividad absoluta, sólo es posible mitigarlo aumentando el ancho de banda o distribuyendo el contenido de la víctima a través de conexiones con distintas direcciones IP de modo que ante un ataque el servicio sólo se vea degradado y no afecte por completo a la disponibilidad.

105. Estos ataques suelen buscar como objetivo adicional al de la pérdida del servicio un intento de inyección de *dns poison*. Por ello además de intentar mitigar en la medida de lo posible la saturación del ancho de banda disponible es recomendable bloquear las respuestas no deseadas hacia los servidores de DNS. Para ello es necesario implantar soluciones que permitan el tracking de las peticiones y respuestas de los servidores DNS, bloqueando aquellas que no se hayan originado en la red del cliente.

7.1.2 PROTECCIONES DE RED PARA MITIGAR ATAQUES DE DENEGACIÓN DE SERVICIO

7.1.2.1 CACHÉS

106. Cuando un servicio es víctima de un ataque de DoS orientado a agotar los recursos del sistema que lo hospeda una posible técnica de mitigación es el uso de servidores caché. Gracias a su uso se logra disminuir la carga de los sistemas soportando así un mayor número de consultas concurrentes además de disminuir el tiempo de respuesta.

107. En el caso de servicios web, dependiendo de la arquitectura, colocando servidores cachés frente a los servidores web se puede lograr disminuir el tiempo de respuesta un factor entre 300 y 1000x.

108. En general los servidores caché disponen de varias funcionalidades además de cachear contenido:

- **Balanceo de carga**: en función de la carga que están soportando los servidores web el servidor caché es capaz de entregar las peticiones al más liberado, optimizando así los recursos.
- **Edge Side Includes (ESI** [11]**)**: permite cachear de forma individual ciertas partes de una página web, algo especialmente útil cuando se está cacheando contenido dinámico.
- **Servicio auxiliar**: son capaces de seguir respondiendo peticiones web incluso cuando los servidores están caídos de modo que es posible entregar un mensaje informativo a los usuarios comunicando la indisponibilidad del servicio.
- **Estadísticas y registro:** muestran estadísticas en tiempo real y registran el estado de modo que son útiles para detectar anomalías en el ratio de visitas web.

7.1.2.2 HERRAMIENTAS ESPECIALIZADAS EN MITIGACIÓN DE ATAQUES DOS

109. Existen herramientas especializadas en la detección y mitigación de ataques clásicos como los comentados en el punto "CONFIGURACIONES ESPECÍFICAS PARA PREVENIR ATAQUES CONOCIDOS" además de otros basados en firmas y comportamientos anómalos. Más concretamente estas herramientas son capaces de mitigar:

- Ataques provenientes de herramientas conocidas (Slowloris, Sockstress, LOIC, etc)
- Botnets
- Ataques dirigidos hacia aplicaciones (exploits)
- Ataques conocidos (*SYN Flood, UDP Flood, ICMP Flood, DNS Flood*, etc)
- Ataques HTTP (Peticiones lentas, cabeceras mal formadas, *HTTP Flood*)

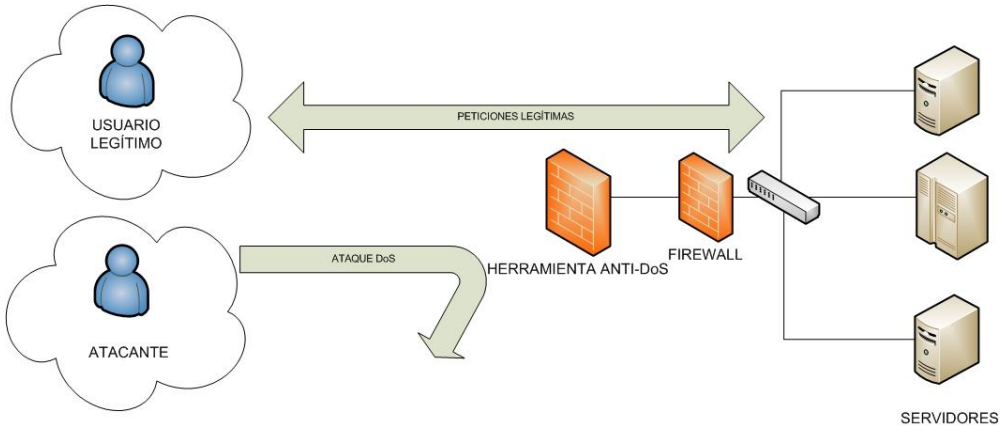


Figura 16 – Herramienta Anti-DoS

110. Estas herramientas especializadas se sitúan frente a la salida a Internet de la Organización consiguiendo filtrar la mayor parte del tráfico malicioso dejando pasar el tráfico legítimo lo que conlleva una serie de beneficios:

- De cara a la protección contra ataques de DoS provenientes de Internet: simplifica la configuración del resto de dispositivos de la red de la organización. Esto es debido a que las medidas de mitigación que se deberían implementar en el resto de firewalls y servidores de la red dejan de ser tan necesarias ya que a priori el tráfico malicioso no debería alcanzarles.
- Optimización de recursos: al detener el tráfico malicioso evitando que llegue a los servidores de la organización se logra que los recursos de éstos se empleen en resolver peticiones legítimas.

111. Esta contramedida tiene una gran facilidad de implantación y una fiabilidad alta ya que integra todas las contramedidas contra ataques conocidos y además puede proteger de otros ataques.

112. Dentro de este tipo de *appliances* se debe buscar la granularidad de las políticas de manera que permitan el establecimiento de distintas protecciones en función de la infraestructura a proteger. Así mismo estas soluciones pueden tener entre otras capacidades:

- Geolocalización: permiten establecer políticas de acceso diferenciadas en función de la localización geográfica de los clientes de manera que se pueda minimizar la exposición de las infraestructuras desde ciertos países.
- Listas de reputación: permiten descartar las conexiones desde aquellas direcciones IPs que se hayan identificado como atacantes, así como direcciones de *botnets* conocidas o por ejemplo servicios de anonimización.

7.2 CONTRAMEDIDAS PARA SISTEMAS INFORMACIÓN CATEGORIZADOS CON NIVEL ALTO

7.2.1 DETECCIÓN DE ATQUES DE DENEGACIÓN DE SERVICIO

113. La detección temprana de ataques de DoS permite tomar medidas de mitigación a tiempo reduciendo el impacto del ataque. Dichas medidas excepcionales pueden ser acciones tanto a nivel de intranet como a nivel de ISP (*Internet Service Provider*, proveedor del servicio de Internet). Del mismo modo, la detección también es posible realizarla en la intranet o en el ISP [12].

114. Existen varias aproximaciones para la detección de ataques de denegación de servicio en la intranet y así poder discernir entre el tráfico perteneciente a un ataque y el tráfico legítimo:

- Comparar el contenido del paquete con firmas de paquetes pertenecientes a ataques: los paquetes provenientes de algunas herramientas destinadas a realizar ataques de DoS tienen características singulares que permiten su identificación. Del mismo modo, paquetes destinados a explotar vulnerabilidades en los servicios pueden ser identificados mediante firmas.
- En función del número de paquetes recibidos por dirección IP: si el número es anormalmente alto es probable que el tráfico sea perteneciente a un ataque.
- Aumento abrupto del número de peticiones desde distintas direcciones IPs: este hecho puede ser un indicador de la sucesión de un ataque de DoS distribuido.

115. La monitorización no es una contramedida completa y debe combinarse con otras para que sea efectiva, por lo tanto una vez detectado el tráfico malicioso en el perímetro de la red mediante cortafuegos, IPS (Intrusion Prevention System) o *herramientas especializadas* de detección de ataques de DoS algunas medidas a tomar son:

- Reducir la cantidad de paquetes pertenecientes al ataque que alcanzan la organización informando a los routers del ISP mediante *pushback* [13].

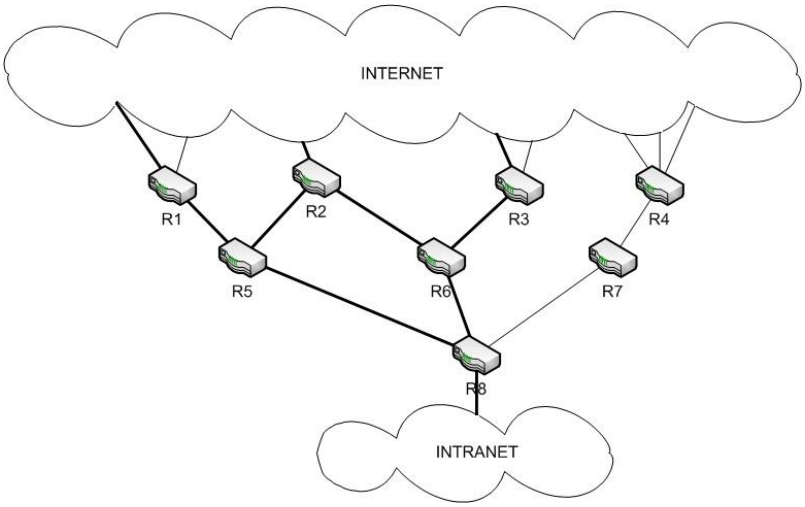


Figura 17 - Ataque de DDoS

116. Por ejemplo, al detectarse un ataque en el perímetro de la intranet, se procedería a definir un patrón que identificase el tráfico malicioso y éste sería entregado a los routers superiores con el objetivo de detener parte del ataque antes de que alcance la red de la organización.

- Tras la identificación del tráfico malicioso añadir reglas a los cortafuegos o IPS para impedir que los paquetes accedan a la red de la entidad.
- Monitorizar la carga de los servidores para identificar posibles caídas.

7.2.2 REDES DE ENTREGA DE CONTENIDO (CDN)

117. Las Redes de Entrega de Contenido distribuyen el almacenamiento de los datos a lo largo del planeta logrando que, en caso de ataque, el impacto no sea global quedando limitado a una región concreta.

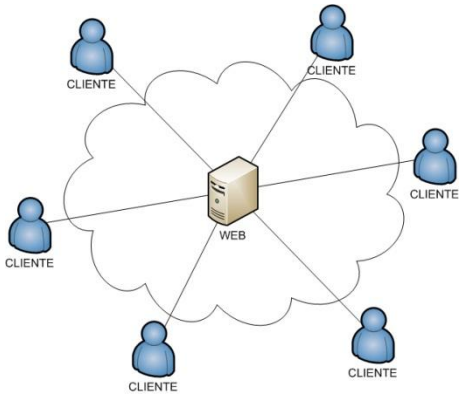


Figura 18 – Web sin CDN

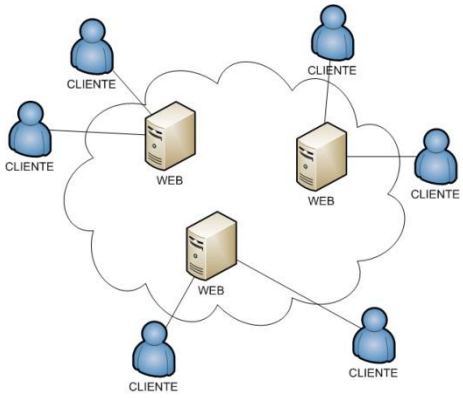


Figura 19 – Web con CDN

118. Con el diseño clásico las peticiones de los clientes, indistintamente de su ubicación física, se concentran en un mismo punto. En cambio con el uso de las Redes de Entrega de Contenido las peticiones se entregan al servidor más cercano logrando así que en caso de ataque sólo se vea afectada la disponibilidad del servicio en el área del cual provienen las peticiones malintencionadas.

119. Este tipo de redes de puede utilizar para distribuir objetos web, archivos descargables, aplicaciones, servicio DNS, consultas a bases de datos, etc.

120. La efectividad de esta contramedida es muy alta contra ataques de denegación de servicios distribuidos ya que logra absorber el tráfico malicioso sin riesgo de sufrir daños colaterales, aunque supone un gran coste económico.

7.2.3 MEDIDAS EN LOS PROTOCOLOS DE ENRUTADO (REMOTELY-TRIGGERED BLACK HOLE)

121. En los casos de ser víctimas de un ataque de denegación de servicio distribuido (DDoS), una vez identificadas las redes de origen del tráfico malicioso, es posible evitar que éste llegue al perímetro de la red a proteger redirigiendo el tráfico a una red de escape.

122. La técnica RTBH (Remotely-Triggered Black Hole) permite desechar el tráfico de un ataque antes de que alance su destino. Para lograrlo se distribuye entre los distintos routers previos a la red a proteger una ruta que dirige el tráfico malicioso a un "agujero negro", es decir, se entrega el tráfico a una red que no existe.

123. RTBH permite:

- Mitigar ataques de DDoS, gusanos, etc.
- Poner en cuarentena todo el tráfico destinado al objetivo de un ataque.
- Filtrar en base a listas negras.

124. Una vez detectado el ataque mediante técnicas de monitorización, el filtrado de tráfico puede hacerse en función del origen o del destino del mismo [14]:

- En función del destino: en este caso entre los routers cercanos al origen del ataque se distribuye una ruta que deseche el tráfico que tenga como destino la dirección IP de la víctima.

- En función del origen: en este caso entre los routers cercanos a la víctima del ataque, se distribuye una ruta que deseche el tráfico que tenga como origen las direcciones IP de los atacantes.

125. Una vez finalizan los ataques se eliminan las rutas extras que se han creado para mitigar el impacto de los ataques volviendo las tablas de rutas a la normalidad.

126. Para aumentar la efectividad de esta contramedida es necesaria una correcta monitorización para disminuir el tiempo de respuesta: cuando antes se distribuyan las rutas creadas para eliminar el tráfico malicioso menor será el impacto del ataque. Del mismo modo, debe detectarse cuanto antes la finalización del ataque para restaurar las rutas originales disminuyendo el daño colateral que supondría desechar tráfico legítimo.

7.2.4 INTEGRAR LA SEGURIDAD COMO UN REQUISITO DEL CICLO DE DESARROLLO DEL SOFTWARE

127. Además de las medidas para evitar que el tráfico perteneciente a ataques de denegación alcance su objetivo, en el diseño de las aplicaciones y servicios se debe tener en cuenta una serie de medidas para evitar vulnerabilidades que su explotación afecte a la disponibilidad.

128. En el diseño de las aplicaciones se debe tener especial cuidado con las funciones pesadas poniendo especial hincapié en su implementación y en el control de su ejecución. Se debe analizar la eficiencia de los algoritmos implementados y se debe controlar el número de ejecuciones y los parámetros que se pasan a este tipo de funciones.

129. MITRE [15] dispone de una base de datos de vulnerabilidades de código, a continuación se describen las vulnerabilidades más comunes que impactan en la disponibilidad:

- **Complejidad algorítmica (CWE-407):** una implementación ineficiente de un algoritmo impacta directamente en la cantidad de recursos requeridos para su ejecución. Es recomendable invertir esfuerzo en optimizar implementaciones y en controlar la ejecución de este tipo de funciones.
- **Bloqueo incorrecto (CWE-667, CWE-412, CWE-764):** sucede cuando el programa no bloquea un recurso correctamente o no lo libera una vez finaliza su uso. Esto ocasiona estados no previstos en los recursos pudiendo provocar comportamientos inesperados que afecten a la disponibilidad.
- **Manejo inadecuado de archivo con un alto ratio de compresión (CWE-409):** en los programas en los cuales aceptan como entrada archivos comprimidos hay que tener especial cuidado en su tratamiento. Puede darse el caso que el archivo tenga un ratio de compresión muy alto y al descomprimirse sature los recursos del equipo provocando una denegación de servicio.
- **Consumo de los descriptores de fichero (CWE-769):** se debe controlar el número de ficheros máximo que se pueden abrir ya que un usuario malintencionado podría forzar la apertura de más archivos de los que soporta el sistema operativo quedando bloqueada la aplicación y el sistema operativo.

- **Número de iteraciones excesivas (CWE-834, CWE-674):** el número de iteraciones máximas que pueden suceder debe estar controlado, sobre todo en los casos en los cuales el número de iteraciones puede ser modificado por la entrada de un usuario. Esto se aplica también al nivel de recursividad máximo que puede alcanzar una función.
- **Consumo de recursos asimétrico (ataques de amplificación) (CWE-405):** la cantidad de recursos que puede consumir un usuario debe ser controlada, sobre todo en los casos en los cuales un usuario malicioso puede consumir más recursos de los que su nivel de acceso permite. Explotando esta debilidad puede provocar un consumo asimétrico implicando un elevado consumo de recursos del sistema o la red.
- **Reserva de recursos sin control (CWE-770):** el software reserva un recurso reusable o un grupo de ellos, sin especificar restricciones de cuántos puede reservar. Se debe controlar el número de recursos que puede reservar un usuario, sobre todo en los casos en los que no se requiere autenticación.
- **Reserva de recursos insuficientes (CWE-410):** a la hora de diseñar el software y calcular la cantidad de recursos que necesitará, se debe tener en cuenta la posibilidad de que existan picos de carga que supongan un consumo de recursos superior a lo habitual.
- **No liberación de descriptores de archivos tras su uso (CWE-775):** una vez se finaliza el uso de un recurso éste debe ser liberado. En caso contrario podría ocurrir que se consumieran todos los descriptores afectando a la disponibilidad.
- **Registro excesivo de datos (CWE-779):** la cantidad y el tamaño de los logs de una aplicación debe estar controlada, por un lado por la carga extra que supone y por otro porque podría darse el caso de saturar toda la memoria del sistema provocando una denegación de servicio.

7.3 MEDIDAS PARA EVITAR FORMAR PARTE DE BOTNETS

130. El Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica (Real Decreto 3/2010 [1]) especifica que para activos de nivel Alto se deben aplicar medidas para impedir el lanzamiento de ataques desde las propias instalaciones perjudicando a terceros.

131. Mediante monitorización del tráfico saliente y el uso de cortafuegos se puede evitar formar parte de *botnets*.

7.3.1 ANÁLISIS DE TRÁFICO

132. Con herramientas IDS/IPS se puede analizar el tráfico saliente de la organización con el objetivo de identificar el tráfico destinado a servidores de Mando y Control (C2, *command and control*). Una vez identificado el tráfico perteneciente a *botnets* es posible bloquear la conexión e identificar los equipos infectados y proceder a su desinfección.

7.3.2 ANTIVIRUS

133. Para el correcto uso de antivirus se recomienda tomar una serie de medidas mínimas:

- Uso de una herramienta que permita una gestión centralizada: gracias a una consola de administración centralizada es posible monitorizar el estado en el cual se encuentran los distintos equipos de la organización, comprobar la fecha de las firmas instaladas, infecciones, etc.;
- Asegurar la actualización automática de firmas de antivirus y el registro de eventos como son momento de actualización, arranque y paradas de servicios y detección de virus;
- En caso de detección de infección, desconectar el equipo afectado de la red de la entidad;
- Ejecutar análisis periódicos en busca de virus.

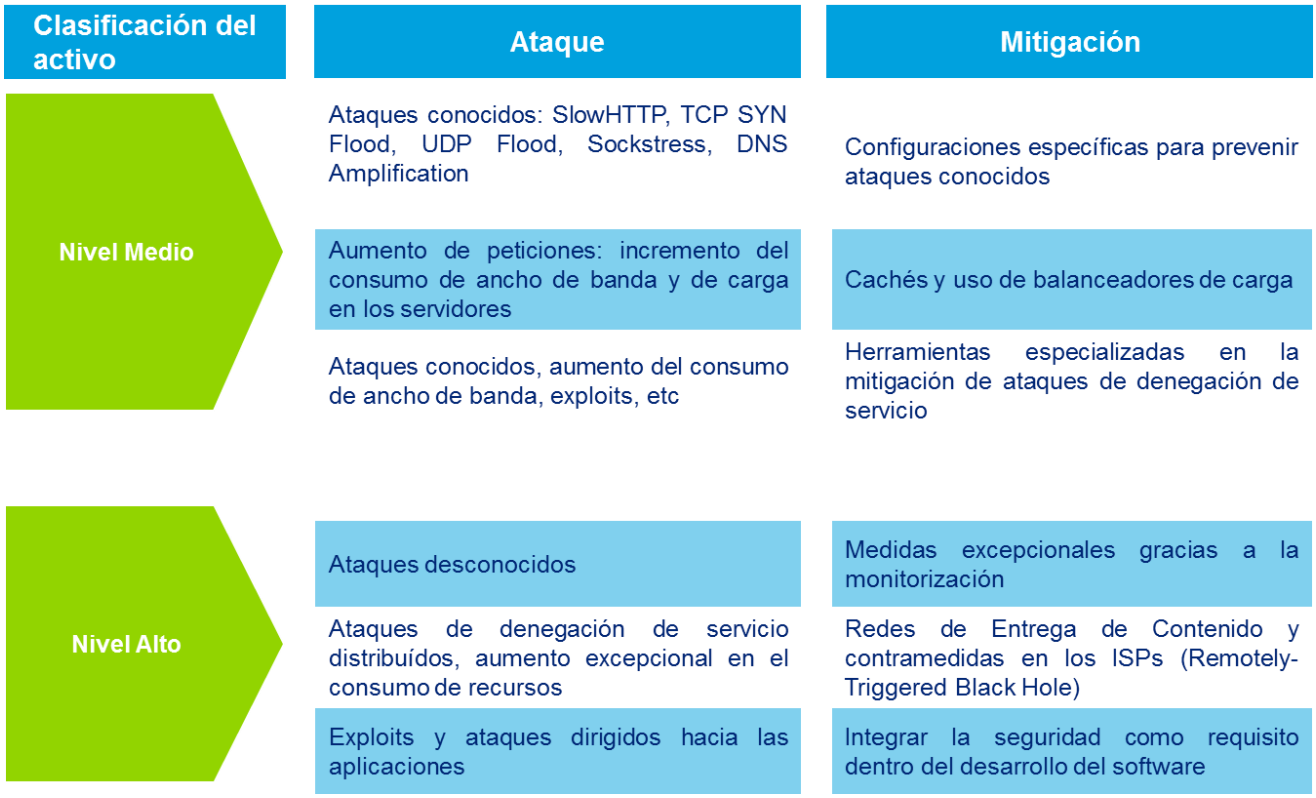


Figura 20 - Resumen de contramedidas contra ataques de DoS

ANEXO A. REFERENCIAS

[1] Gobierno de España, «Real Decreto 3/2010, de 8 de enero, Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica,» 2010.

[2] Ponemon Institute, «Second Annual Cost of Cyber Crime Study - Benchmark Study of U.S. Companies,» 2011.

[3] CCN, «CCN-STIC-803 - Esquema Nacional de Seguridad: Valoración de los Sistemas».

[4] S. Shekyan, «Testing Web Servers for Slow HTTP Attacks,» Septiembre 2011. [En línea]. Available: https://community.qualys.com/blogs/securitylabs/2011/09/19/testing-web-servers-for-slow-http-attacks.

[5] S. Shekyan, «How to Protect Against Slow HTTP Attacks,» Noviembre 2011. [En línea]. Available: https://community.qualys.com/blogs/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks.

[6] S. Shekyan, «slowhttptest - Application Layer DoS attack simulator,» [En línea]. Available: http://code.google.com/p/slowhttptest/.

[7] IETF, «RFC4987: TCP SYN Flooding Attacks and Common Mitigations,» 2007. [En línea]. Available: http://www.ietf.org/rfc/rfc4987.txt.

[8] J. Lemon, «Resisting SYN flood DoS attacks with a SYN cache».

[9] D. J. Bernstein, «SYN Cookies,» [En línea]. Available: http://cr.yp.to/syncookies.html.

[10] J. C. Louis, «Introduction to Sockstress - A TCP Socket Stress Testing Framework».

[11] Akamai, Art Technology Group, BEA Systems, Circadence Corporation, Digital Island, Interwoven, Oracle Corporation, Vignette Corporation, «ESI Language Specification,» 2001. [En línea]. Available: http://www.w3.org/TR/esi-lang.

[12] A. Akella, A. Bharambe, M. Reiter y S. Seshan, «Detecting DDoS Attacks on ISP Networks,» Carnegie Mellon University.

[13] S. M. Bellovin y J. Ioannidis, «Implementing Pushback: Router-Based Defense Against DDoS Attacks,» AT&T Labs Research.

[14] Cisco, «Remotely Triggered Black Hole Filtering - Destination Based and Source Based».

[15] MITRE, «Common Weakness Enumeration,» [En línea]. Available: http://cwe.mitre.org.

[16] Imperva, «Denial of Service Attacks: A Comprehensive Guide to Trends, Techniques, and Technologies,» 2012.

[17] Software Engineering Institute: Carnegie Mellon, «CERT® Advisory CA-1996-01 UDP Port Denial-of-Service Attack,» 1997. [En línea]. Available: http://www.cert.org/advisories/CA-1996-01.html.