Deloitte.

RootedCON

DoS
Un enfoque práctico

Gerardo García Peña

Agenda

- **Introducción**
- Metodología de Análisis
- Herramientas aportadas
- Taxonomía
- Contramedidas
- Clasificación, técnicas, herramientas y mitigación
- Bibliografía

Introducción

¿Qué es?

- Destruir o saturar activos clave de una víctima, privando el acceso a sus recursos.
- Generalmente los ataques más relevantes se ejecutan a través de Internet.
  - Aunque esto es sólo uno de los posibles medios de ataque.
  - Existen muchos tipos de medios y técnicas de ataque, tal como han demostrado los gusanos y troyanos.
  - Por ello es un fenómeno difícil de entender.

Introducción

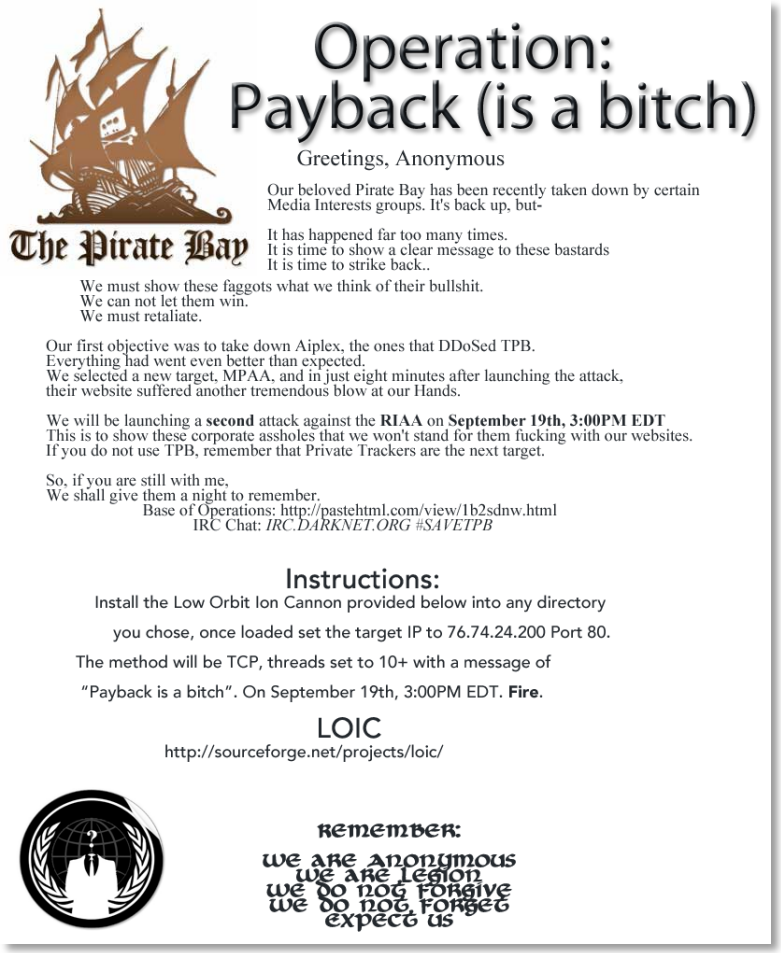
¿Porqué?

- Motivos personales o venganza
- Prestigio (ganar respeto)
- Ganancias materiales (dañar a un competidor)
- Razones políticas
- Ataques indirectos (la víctima es alguien que confía en el servicio atacado)

Introducción

Tipos de riesgos que incluye

- Pérdidas económicas
- Pérdidas de imagen
- Malfuncionamiento de sistemas
- Impacto en la integridad de los datos
- …



The Pirate Bay

Operation:
Payback (is a bitch)

Greetings, Anonymous

Our beloved Pirate Bay has been recently taken down by certain Media Interests groups. It's back up, but-

It has happened far too many times.
It is time to show a clear message to these bastards
It is time to strike back..

We must show these faggots what we think of their bullshit.
We can not let them win.
We must retaliate.

Our first objective was to take down Aiplex, the ones that DDoSed TPB.
Everything had went even better than expected.
We selected a new target, MPAA, and in just eight minutes after launching the attack, their website suffered another tremendous blow at our Hands.

We will be launching a **second** attack against the **RIAA** on **September 19th, 3:00PM EDT**
This is to show these corporate assholes that we won't stand for them fucking with our websites.
If you do not use TPB, remember that Private Trackers are the next target.

So, if you are still with me,
We shall give them a night to remember.
Base of Operations: http://pastehtml.com/view/1b2sdnw.html
IRC Chat: *IRC.DARKNET.ORG #SAVETPB*

Instructions:

Install the Low Orbit Ion Cannon provided below into any directory

you chose, once loaded set the target IP to 76.74.24.200 Port 80.

The method will be TCP, threads set to 10+ with a message of

"Payback is a bitch". On September 19th, 3:00PM EDT. **Fire**.

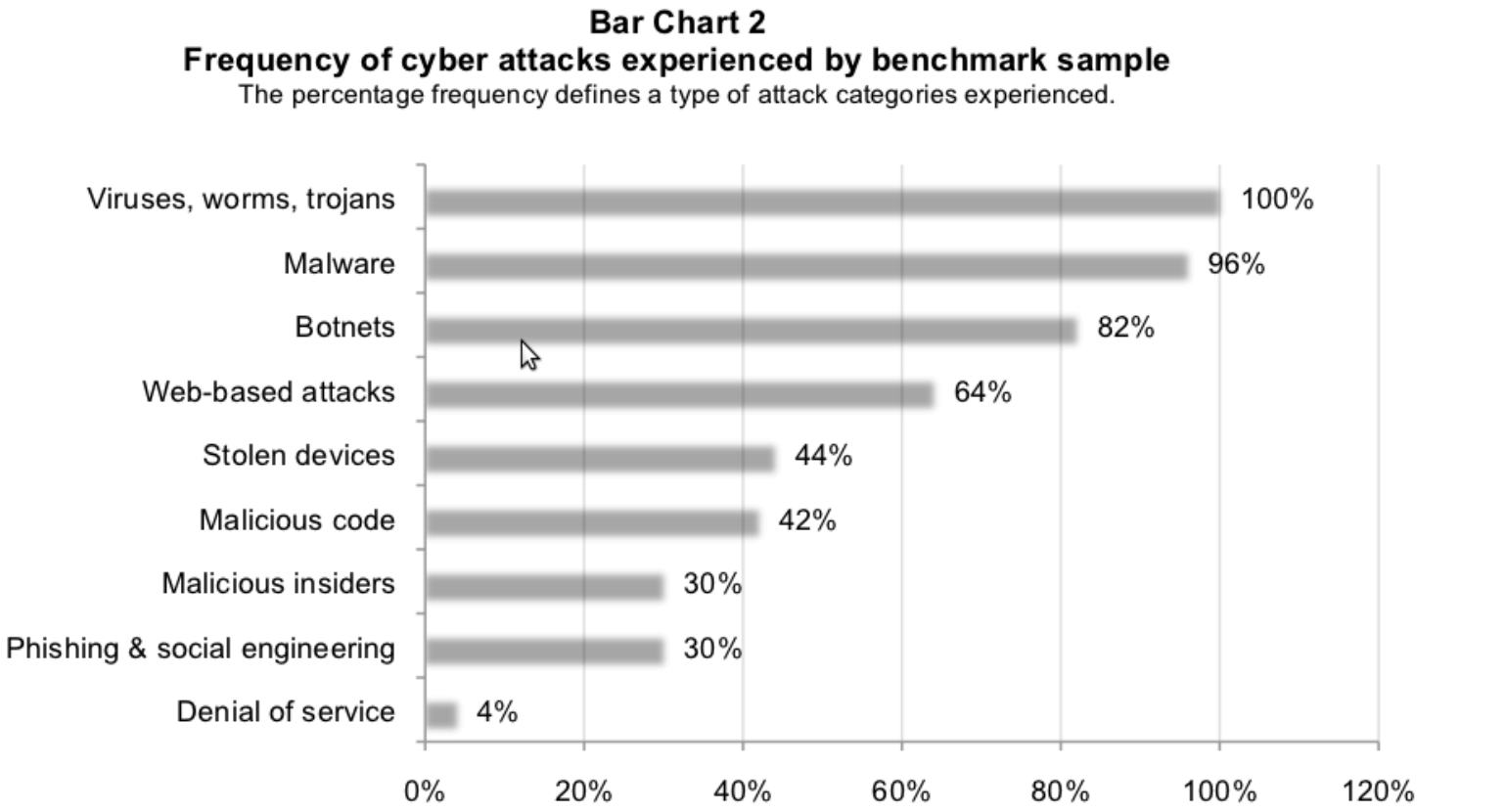
LOIC
http://sourceforge.net/projects/loic/

REMEMBER:
WE ARE ANONYMOUS
WE ARE LEGION
WE DO NOT FORGIVE
WE DO NOT FORGET
EXPECT US

Introducción

¿Qué probabilidad e impacto tiene? (1/3)

- No es el riesgo más común,
- aunque va “in crescendo”

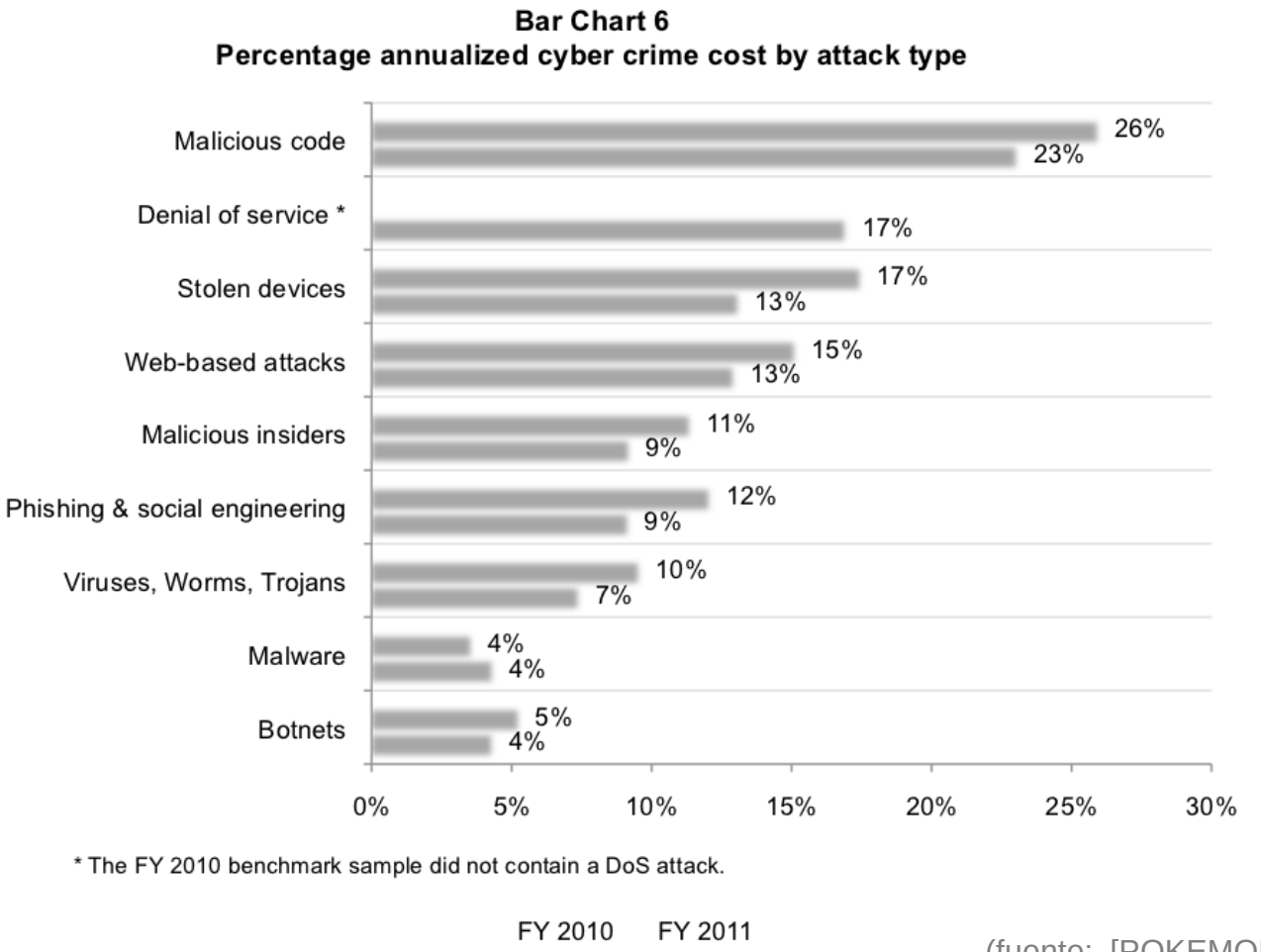


(fuente: [POKEMON])

Introducción

¿Qué probabilidad e impacto tiene? (2/3)

- No obstante su impacto no es nada desdeñable

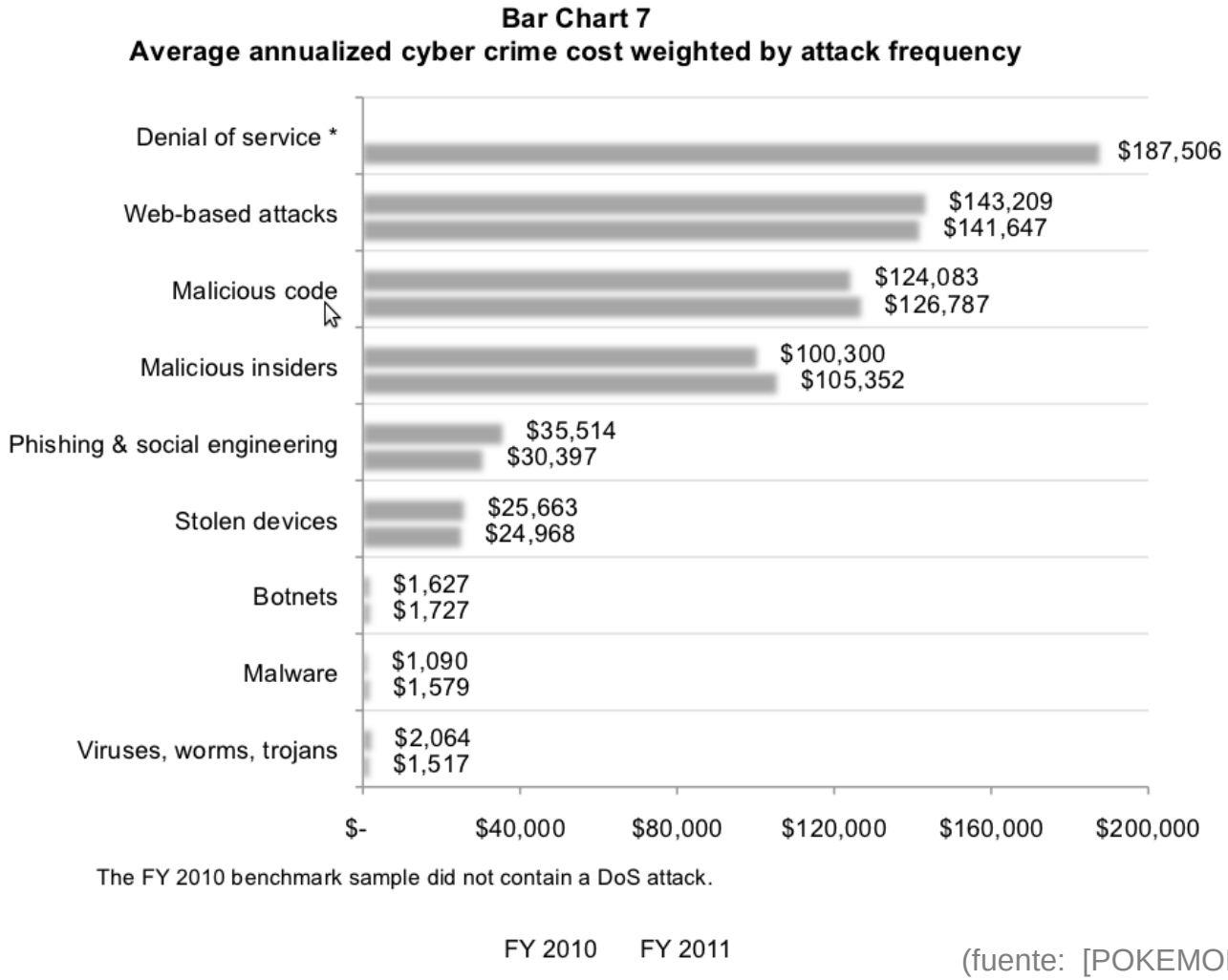


(fuente: [POKEMON])

Introducción

¿Qué probabilidad e impacto tiene? (3/3)

- Si cruzamos la probabilidad de ocurrencia al año con el coste del ataque vemos que el asunto no es de broma, a pesar de ser puntual.



(fuente: [POKEMON])

Introducción

¿Por qué sucede?

- Es un riesgo al cual la gente evita enfrentarse.
- Primero porque probarlo es complicado por sus posibles implicaciones
- Y porque su mitigación depende del trabajo de varios equipos y rara vez puede mitigarse en un solo punto
- Multidisciplinar
- Situaciones extremas
- Debido a ello los diferentes equipos suelen pasarse la patata caliente unos a los otros.

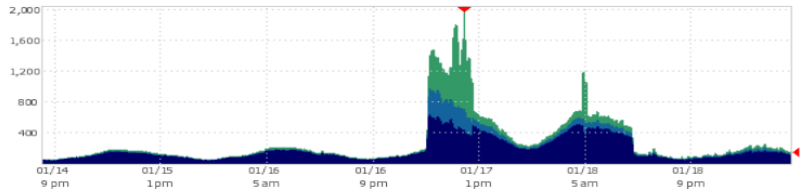
Agenda

- Introducción
- **Metodología de Análisis**
- Herramientas aportadas
- Taxonomía
- Contramedidas
- Clasificación, técnicas, herramientas y mitigación
- Bibliografía

Metodología de análisis

Gestión de una DoS

- Preparación
  - Planes de actuación,
  - preparación de la plataforma, y
  - acuerdos con proveedores e ISP's
- Inspección
  - Captura e identificación del ataque
  - Monitorización del mismo
- Absorber
  - Dimensionamiento del sistema
  - Priorización de servicios
  - Protección de activos
- Detener
  - Bloquear el ataque
  - Priorización de entrega de información
  - Limitar el ataque o carga



Before Attack

Agent	% Visitors
IE 8.0	21%
Firefox 3.6	16%
Feedfetcher	13%
Safari 533.19	9.7%
IE 7.0	8.8%
Other Browsers	32%

Location	Requests	% of Total
USA	66,795,330	36%
United Kingdom	16,879,792	9%
France	11,563,374	6%
Germany	11,344,749	6%
Canada	8,149,259	4%
All Other Countries	66,434,318	40%
Total	185,436,822	100%

During Attack

Agent	% Visitors
Opera	32%
Firefox 3.0	8.3%
Firefox 2.0	7.5%
IE 7.0	6.5%
Firefox 1.5	5.8%
Other Browsers	40%

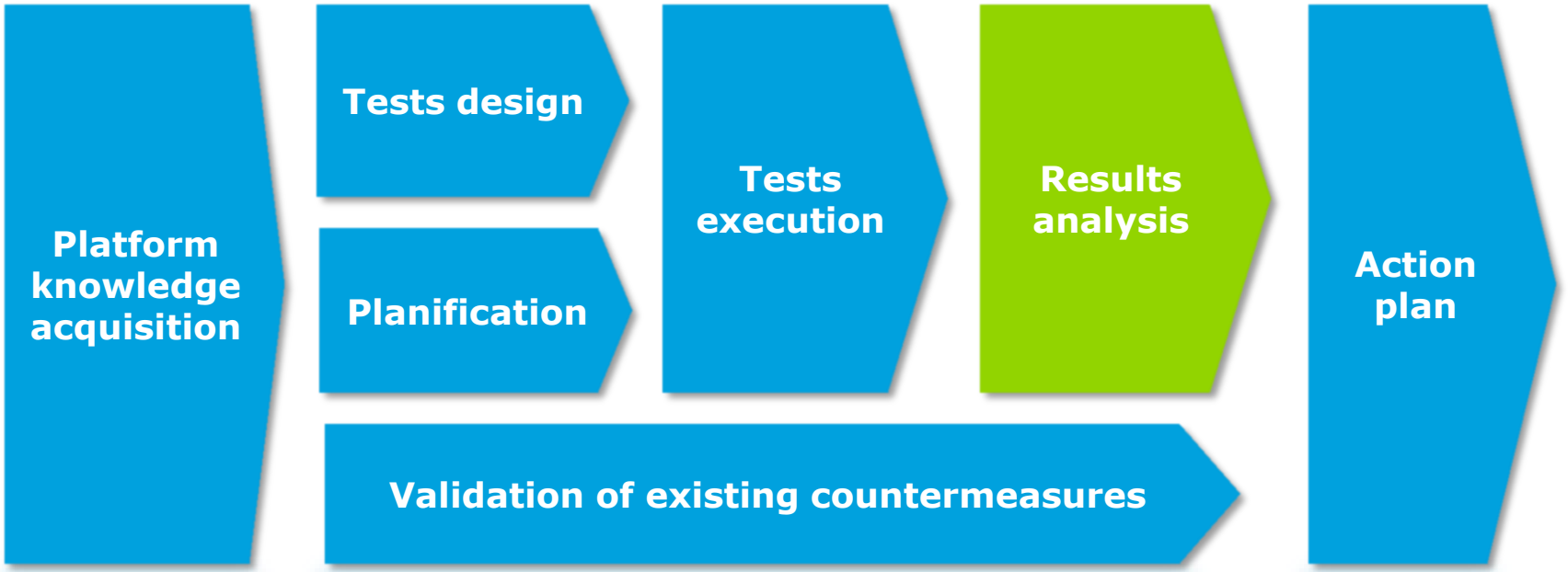
Location	Requests	% of Total
USA	75,068,890	27%
Brazil	26,586,001	10%
United Kingdom	14,759,973	4%
Canada	9.643.411	4%
Mexico	9,234,362	3%
All Other Countries	140,125,542	50%
Total	275,418,179	100%

(fuente: [AKADDOS])

Metodología de análisis

Metodología de auditoria, evaluación y preparación (1/2)

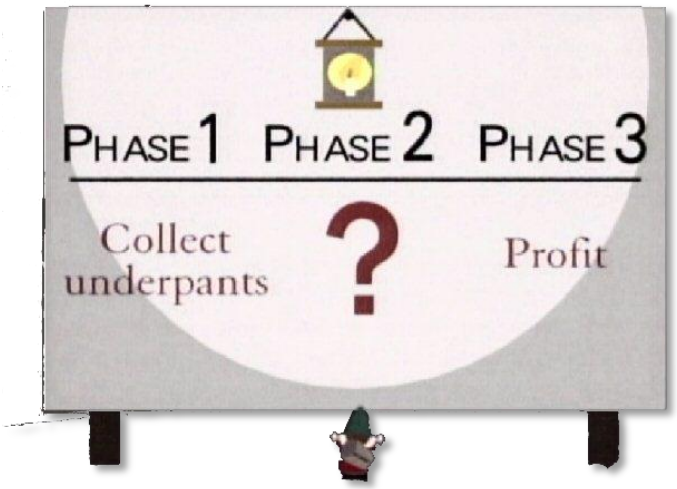
- Un proyecto de
  - Denegación de Servicio
  - Evaluación de rendimiento
  - Pruebas de carga
  - …
- Suele tener siempre el mismo aspecto



Metodología de análisis

Metodología de auditoria, evaluación y preparación (2/2)

- Adquisición de conocimiento
  - Análisis de mapas de red
  - Análisis de comunicaciones y protocolos (pcap analyzer)
  - Identificación de la cadena de servicios y activos
  - Identificación de los sistemas de monitorización
    - ❖ HP OpenView? SNMP? WEBEM/WMI? iftop?
- Identificación y diseño de pruebas
  - Análisis del activo y valoración según taxonomías de DoS
  - Identificar dependencias o afectación del activo sobre el resto del conjunto
  - Diseñar como mediremos rendimiento
- Planificación
  - ¿Cómo conduciremos las pruebas?
  - ¿Qué herramientas usaremos?
  - ¿Hay que programar?
- Ejecución
- Validación contramedidas existentes, y diseño de mecanismos de defensa nuevos



Agenda

- Introducción
- Metodología de Análisis
- **Herramientas aportadas**
- Taxonomía
- Contramedidas
- Clasificación, técnicas, herramientas y mitigación
- Bibliografía

Herramientas aportadas

Herramientas nombradas en esta presentación (1/5)

- Chucuchu
  - Herramienta de DoS sobre aplicaciones web
  - Algo más avanzado que LOIC
  - Diseñado para alto rendimiento
  - “Fácil de configurar”
  - Altamente flexible para poder adaptarse a cualquier web
  - Permite trabajar con transacciones, no sólo con simples peticiones y respuestas.
  - Puede tomar decisiones según las respuestas obtenidas.
  - Es multiproceso y distribuido.
  - Salva conversaciones, tiempos y detalles críticos para un posterior análisis más profundo.

Herramientas aportadas

Herramientas nombradas en esta presentación (2/5)

- Dosis (free & open-source)
  - Herramienta de DoS a nivel de red/infraestructura
  - Esta herramienta aporta
    - ✓ mecanismos para trabajar a bajo nivel los distintos protocolos
    - ✓ Simular comportamientos de red mediante un lenguaje que permite forjar comportamientos y flujos de información de forma sencilla
    - ✓ Ataques de DOS contra protocolos, forjar paquetes y simular comunicaciones maliciosas
    - ✓ Altamente modular y expandible.
    - ✓ Expone una API a los módulos que facilita la creación de nuevos ataques
  - Altamente inestable, es un repositorio de código fuente low-level
  - ¿Alguien se apunta a mejorarlo?

https://github.com/killabytenow/dosis

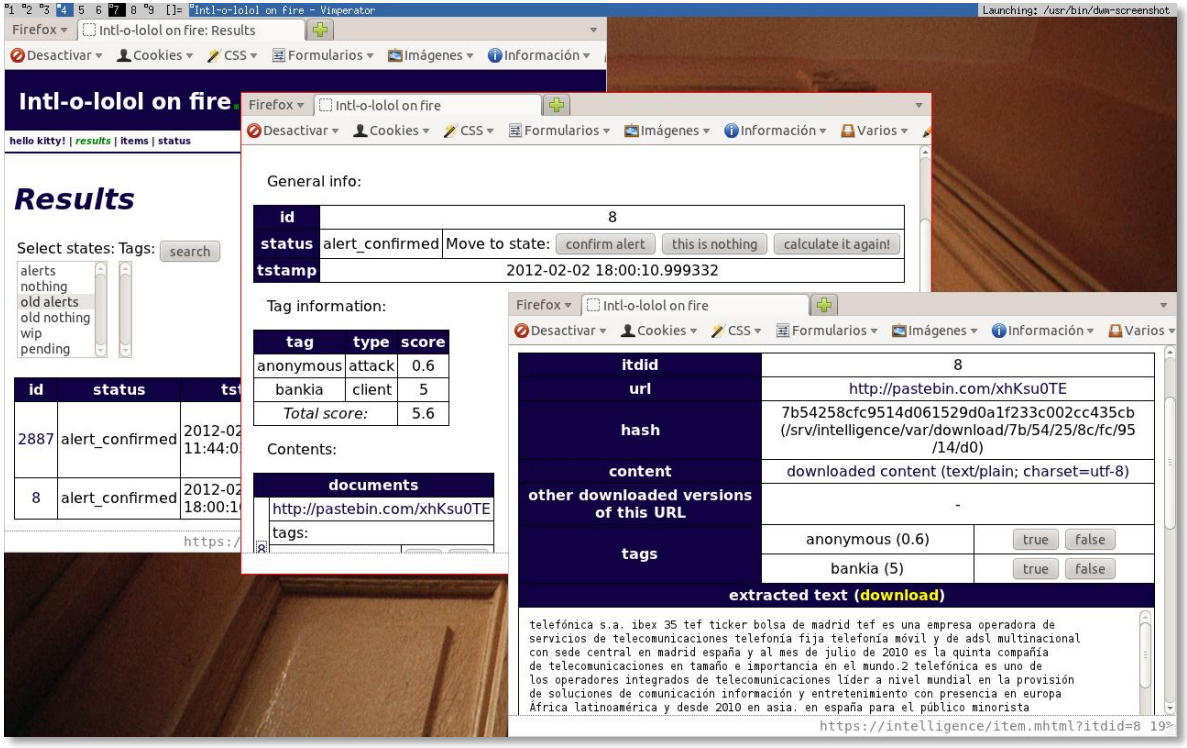
```
# configuration
? THOST="127.0.0.1"
? TPORT="80"
? SRT="5.0"
? RT="30.0"

# script
#0.0 ON 1 LISTEN DEBUG
0.0 ON 1 LISTEN
+.0 ON 2 SEND DEBUG
+.0 ON 3 TCP OPEN DST $THOST $TPORT PAYLOAD FILE("tcpopen.payload") DELAY 100 DEBUG
+.1 ON 4 TCP RAW DST $THOST $TPORT FLAGS S PERIODIC [ 0.2 ]
$SRT OFF 4
$RT OFF *
```

Herramientas aportadas

Herramientas nombradas en esta presentación (3/5)

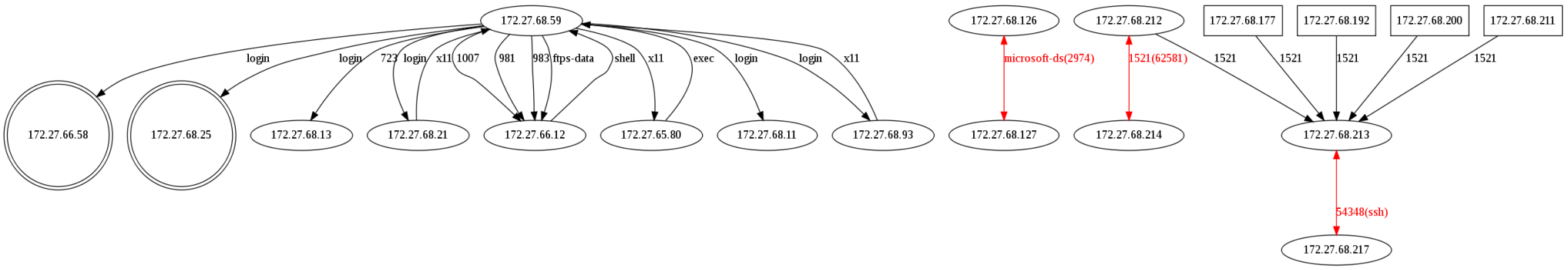
- Intelligence
  - Software de detección temprana de riesgos en Internet
  - Sistema experto que analiza fuentes de información en busca de
    - ✓ Posibles amenazas y ataques
    - ✓ Publicaciones de vulnerabilidades
    - ✓ Information leaks
    - ✓ Noticias de interés



Herramientas aportadas

Herramientas nombradas en esta presentación (4/5)

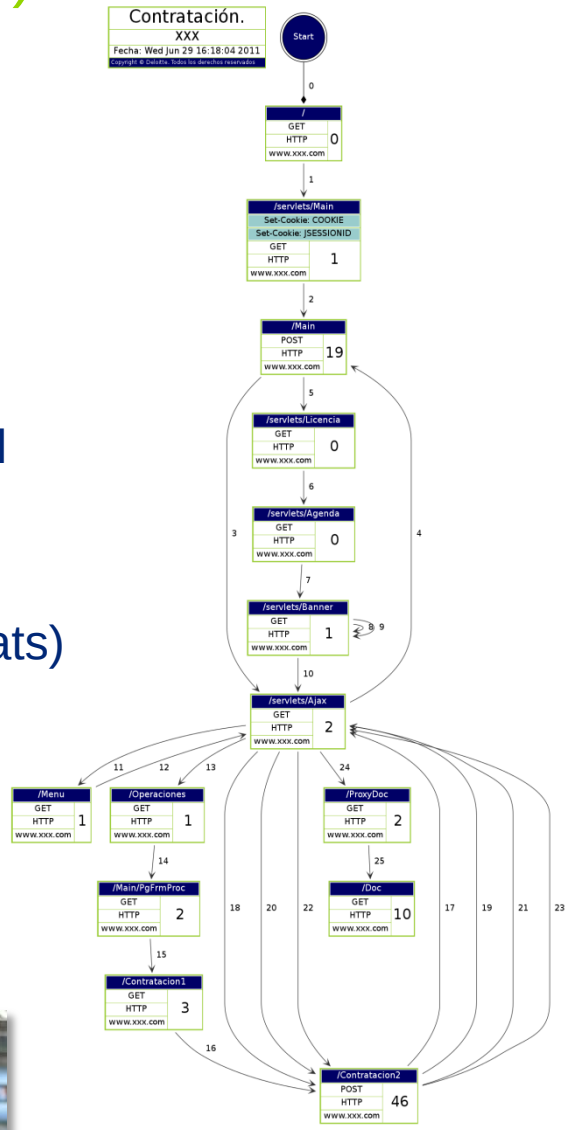
- PCAP analyzer
  - Usado para la monitorización y análisis de sistemas
  - Basado en libpcap, sirve para detectar relaciones entre máquinas de la red.
  - Permite por lo tanto análisis en tiempo de ejecución y post-análisis
  - Genera mapas de red donde se visualizan
    - ✓ Dependencias
    - ✓ Relaciones
    - ✓ Dirección de las comunicaciones y su intensidad
    - ✓ Problemas de red
  - La salida puede ser tanto gráfica, como en report HTML, tablas Excel o incluso volcado a una base de datos SQL.



Herramientas aportadas

Herramientas nombradas en esta presentación (5/5)

- Atackw
  - Sistema para modelización de transacciones web y análisis exhaustivo de las mismas
  - De entre sus muchas funcionalidades usamos su capacidad como herramienta de profiling para
    - ✓ monitorizar transacciones web, o
    - ✓ buscar posibles casos de denegación de servicio a nivel de aplicación
- ANO.LOLCATHOST.ORG (escenario)
  - Es un servidor de imágenes “desenfadadas” en Internet (lolcats)
  - Desarrollado según buenas prácticas SSDLC
  - Sirve como escenario de pruebas y como plataforma donde testear técnicas anti-DOS



Agenda

- Introducción
- Metodología de Análisis
- Herramientas aportadas
- **Taxonomía**
- Contramedidas
- Clasificación, técnicas, herramientas y mitigación
- Bibliografía

Taxonomía

Ese gran desconocido

- Un problema básico a afrontar es la dificultad de entender la DoS.
- Pero para resolver un problema es necesario entenderlo,
- Es un fenómeno complejo que puede afectarnos a diferentes niveles y de muchas formas.
- Pero por desgracia es normal encontrarse con una limitada visión:
  - DoS
  - DoS distribuido
  - Botnets
  - Y poco más
- Por lo que se hace esencial disponer de una taxonomía.
- ¿Que nos permite una taxonomía?
  - Identificar sobre que activos puede cernirse una DoS
  - Visualizar de que modos y en que capas se puede mitigar
  - Las diferentes técnicas por las cuales se puede materializar
- Y en conjunto nos puede orientar para definir un plan para tratarlo de una forma detallada y precisa
  - Mejor que UDP, TCP y ICMP.

Taxonomía

Taxonomías existentes (1/3)

- El artículo "*A Taxonomy of DDoS Attack and DDoS Defense Mechanisms*" [TAXDOS] es un clásico en la materia.
- Realiza una taxonomía detallada y bastante completa de ataques y defensas.
- Taxonomía de ataque:

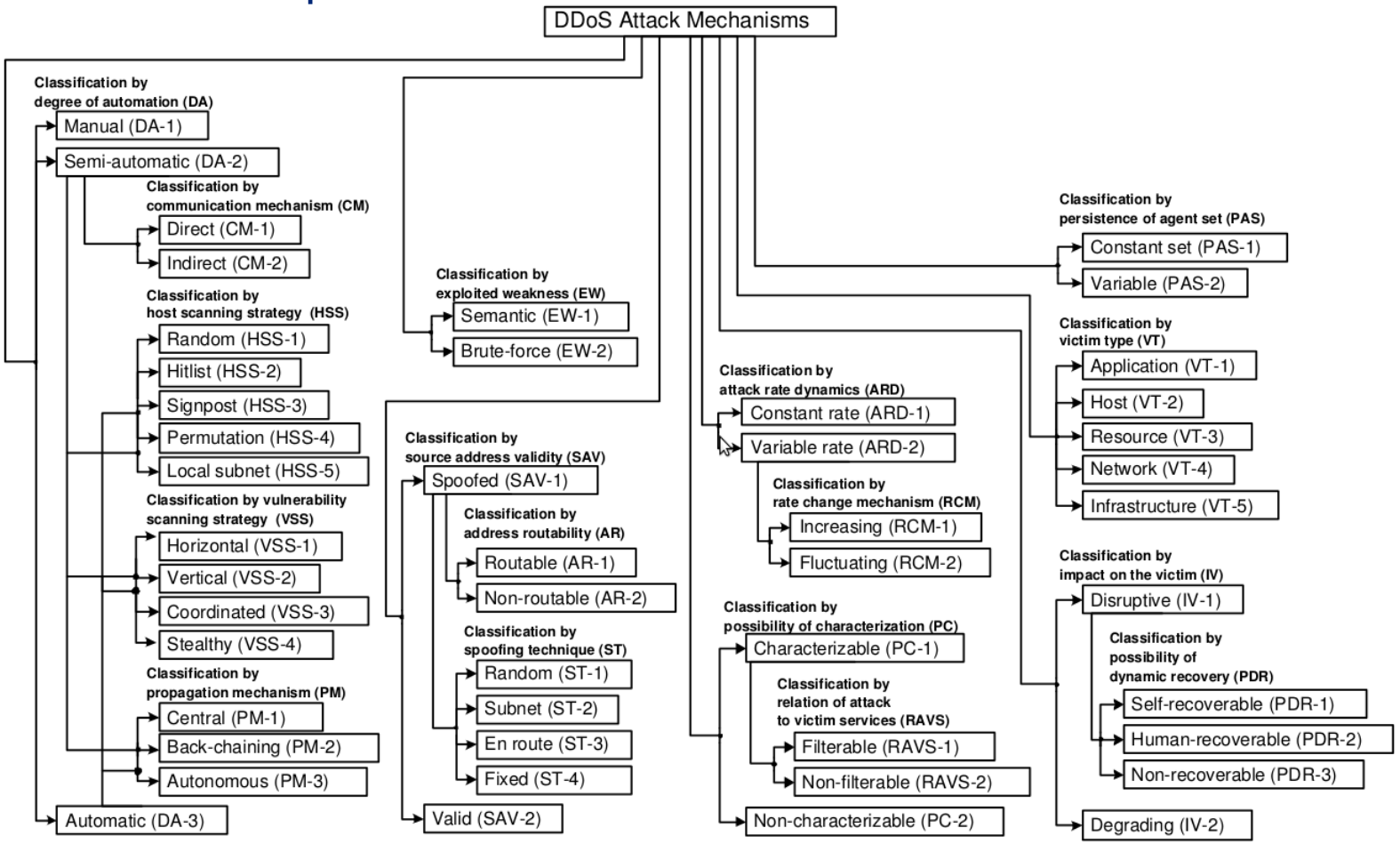
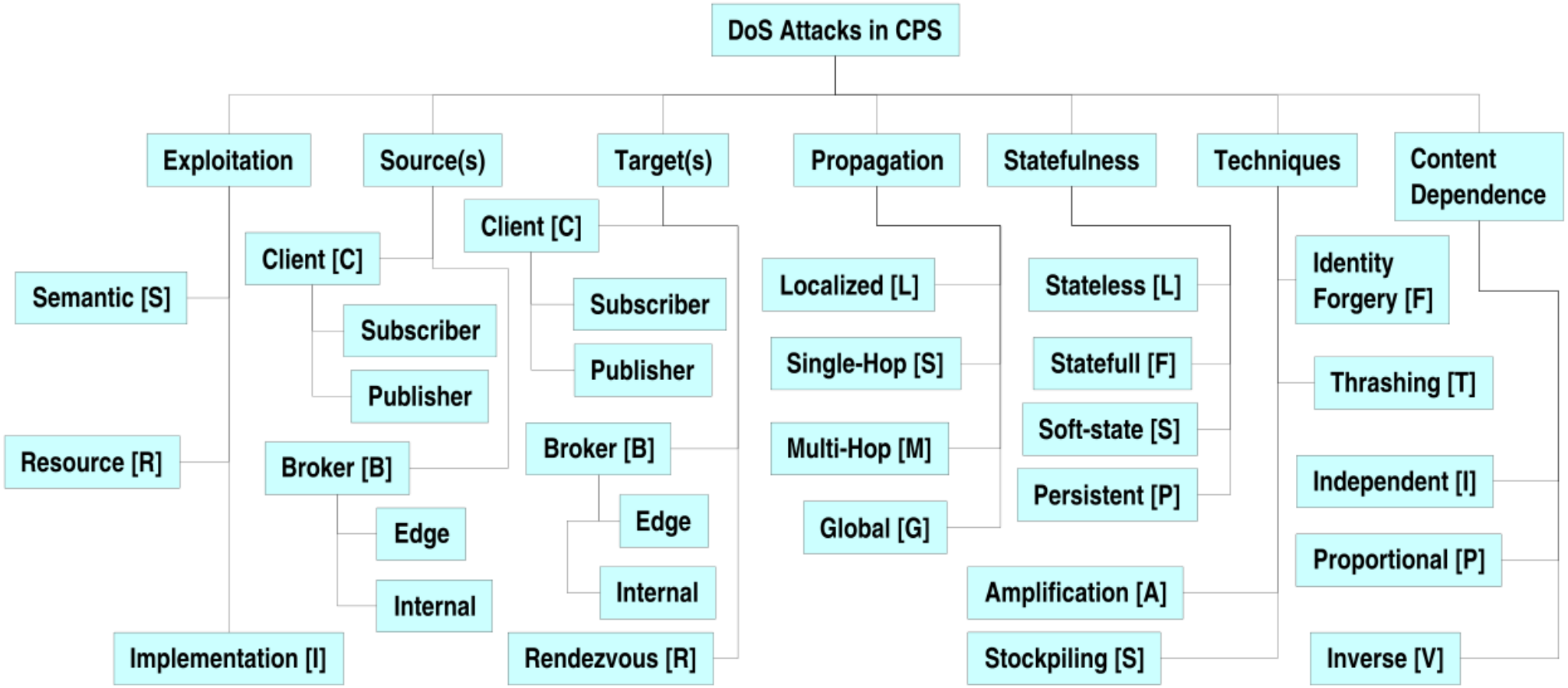


Figure 1: Taxonomy of DDoS Attack Mechanisms

Taxonomía

Taxonomías existentes (2/3)

- Otro artículo, "*A Taxonomy for Denial of Service Attacks in Content-based Publish/Subscribe Systems*" [TAXCPS] presenta una taxonomía contra CPS.



Taxonomía

Taxonomías existentes (3/3)

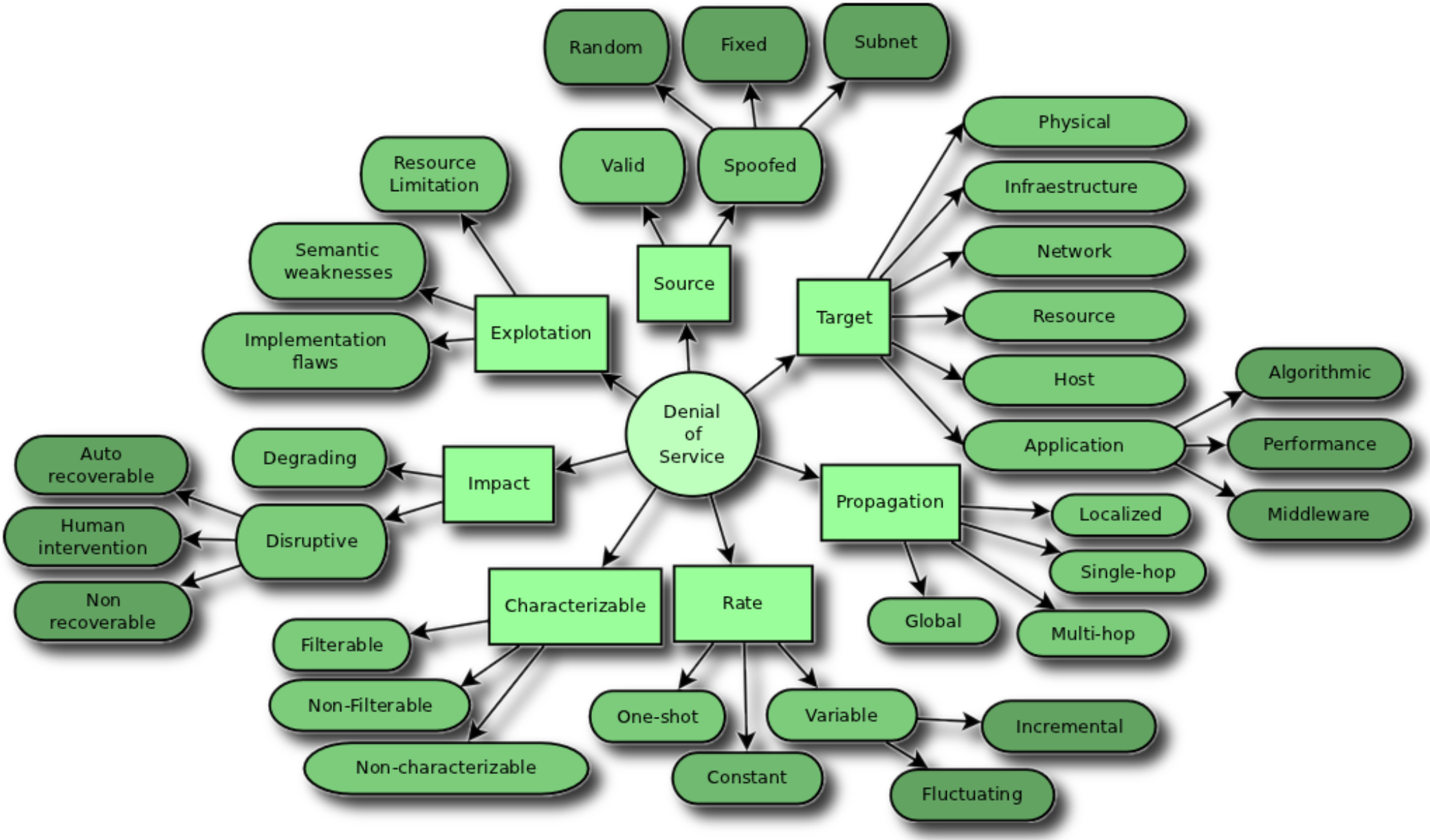
- El problema está en que no existe (o no hemos encontrado) una clasificación taxonómica lo suficientemente completa que cubra todos los ataques DoS conocidos hasta la fecha.

- Por ello nos apoyamos en la clasificaciones taxonómicas actuales para
  - Crear una taxonomía genérica que nos ayude a conducir nuestros proyectos.
  - Sea lo suficientemente simple como para que ayude a
    - ✓ ver a través del bosque
    - ✓ desarrollar un análisis sobre la plataforma existente
    - ✓ valorar las posibles mitigaciones sobre ella

- En nuestra metodología de DoS es esencial una aproximación así ya que:
  - Nos ayuda a enumerar los activos
  - Valorar como estos mismos activos pueden ser usados durante la DoS
  - Detectar los posibles vectores de ataque
  - Y sus mitigaciones

Taxonomía

Taxonomía de ejemplo



Taxonomía

Taxonomía de ejemplo (1/5)

- La siguiente taxonomía es una proposición que podría servir durante la planificación de un análisis:

	Clasificación		Descripción	Ejemplos
Explotación (mecanismo)	Resource limitation (bruteforce)		Ingente cantidad de conexiones/peticiones que agotan el recurso de la víctima	• Ataque deliberado • Referenciados en portal importante • Éxito
	Semantic weaknesses		Explota una característica específica del protocolo o sistema	• El protocolo no es seguro • Una funcionalidad puede usarse para provocar un DoS
	Implementation flaws		Aprovecha debilidades en la implementación de protocolos o tecnologías	• Una mala implementación de software • Un atacante descubre un error que deriva en un excesivo consumo • Una contramedida DoS se convierte en un agravante
Origen	Spoofed	Random	Nos llegan paquetes de direcciones aleatorias y aparentemente falsas	• Ataque de DoS a nivel de red, probablemente del tipo SYN flood, ICMP flood, UDP flood.
		Subnet	Llegan paquetes de subredes identificables	• Posible ataque de tipo reflexion • Puede ser un dispositivo mal configurado que monitorice una dirección de red
		Fixed	Llegan paquetes de direcciones falsas facilmente identificables	• Un ataque limitado por algún tipo de reglas egress/ingress.
	Válidas		Podemos identiicar el origen del ataque	• Puede tratarse de un ataque social (anonymous) • Botnets • Un efecto menéame

Taxonomía

Taxonomía de ejemplo (2/5)

- La siguiente taxonomía es una proposición que podría servir durante la planificación de un análisis:

	Clasificación	Descripción	Ejemplos
Objetivo	Físico	El objetivo es dañar físicamente o permanentemente un activo de forma que el servicio quede caido o destruído	• Puede ser una mala actualización de firmware • Una descarga no verificada con contenido corrupto • Un atacante que ha encontrado un mecanismo remoto de destrucción • Se fuerza un sistema mecánico
	Infraestructura	Se realiza un ataque que actua negativamente sobre un activo o protocolo de forma que el servicio cae	• Un balanceador de carga que se queda sin RAM • Un ataque de ARP poisoning • La red de nuestro ISP cae
	Red	Ataques donde la capacidad de red se ve afectada debido a un gran uso de ancho de banda	• Anonymous
	Recurso	Se ataca,directa o indirectamente un recurso de él que depende el servicio	• Se desconecta accidentalmente un servidor DNS de él que depende la red ESCADA • Un pico de autenticación de usuarios sobrecarga el backend de autenticación afectando a todos los servicios de la empresa.
	Host	Se ataca directamente la plataforma del servidor para deshabilitar el servicio	• Ataques sockstress • Blaster

Taxonomía

Taxonomía de ejemplo (3/5)

- La siguiente taxonomía es una proposición que podría servir durante la planificación de un análisis:

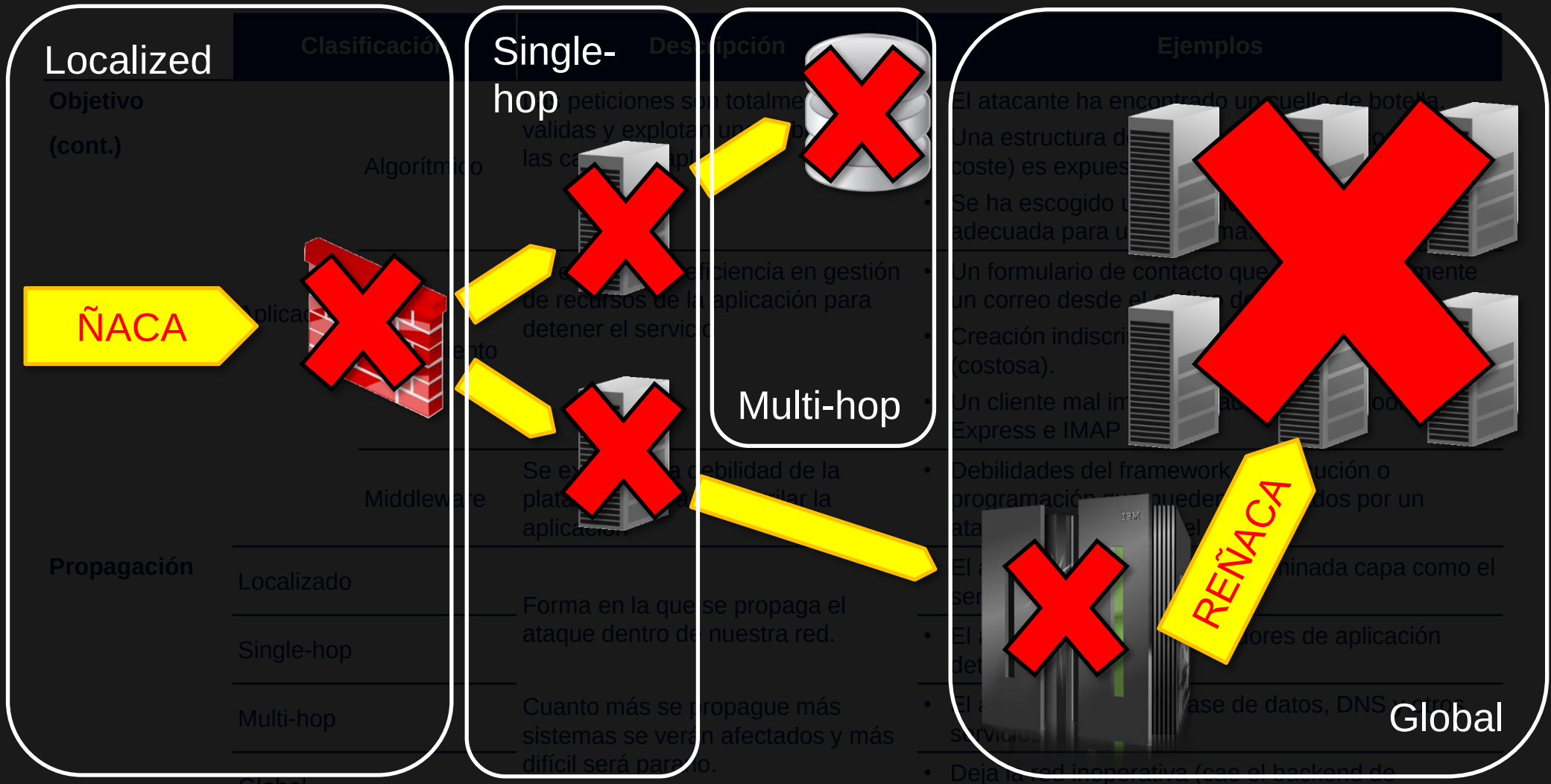
	Clasificación		Descripción	Ejemplos
Objetivo (cont.)	Aplicación	Algorítmico	Las peticiones son totalmente válidas y explotan una debilidad de las capas de aplicación	• El atacante ha encontrado un cuello de botella. • Una estructura de datos vulnerable (casos de peor coste) es expuesta al usuario. • Se ha escogido una solución algorítmica no adecuada para un problema.
		Rendimiento	Se explota la ineficiencia en gestión de recursos de la aplicación para detener el servicio	• Un formulario de contacto que envía directamente un correo desde el código de la interfaz web • Creación indiscriminada de información de sesión (costosa). • Un cliente mal implementado como Outlook Express e IMAP
		Middleware	Se explota una debilidad de la plataforma para descarrilar la aplicación	• Debilidades del framework de ejecución o programación que pueden ser usados por un atacante para tumbar el servicio
Propagación	Localizado		Forma en la que se propaga el ataque dentro de nuestra red. Cuanto más se propague más sistemas se verán afectados y más difícil será pararlo.	• El ataque no supera una determinada capa como el servidor web o firewall
	Single-hop			• El ataque afecta a los servidores de aplicación detrás
	Multi-hop			• El ataque afecta a la base de datos, DNS y otros servicios
	Global			• Deja la red inoperativa (cae el backend de autenticación, p.ej)

Taxonomía

Taxonomía de ataques (6/7)

(Definición gráfica de una propagación)

- La siguiente taxonomía es una proposición que podría servir durante la planificación de un análisis:



Clasificación	Descripción	Ejemplos
Objetivo		
(cont.)		
Algorítmico	peticiones s totalme válidas y explota un las c pl	El atacante ha encontrado un cuello de botella Una estructura d o coste) es expues Se ha escogido adecuada para u ma
Aplicac nto	iciencia en gestión de recursos de la aplicación para detener el servic	Un formulario de contacto que mente un correo desde el d Creación indiscri (costosa). Un cliente mal in oc Express e IMAP
Middlewa e	Se e a debilidad de la plata lar la aplicación	Debilidades del framework ución o programación pueden dos por un ata el
Propagación		El minada capa como el ser
Localizado	Forma en la que se propaga el ataque dentro de nuestra red.	El ores de aplicación det
Single-hop		El ase de datos, DNS, otros servicios
Multi-hop	Cuanto más se propague más sistemas se verán afectados y más difícil será para o.	Deja la red inoperativa (cae el backend de autenticación, p.ej)
Global		

Taxonomía

Taxonomía de ejemplo (4/5)

- La siguiente taxonomía es una proposición que podría servir durante la planificación de un análisis:

	Clasificación		Descripción	Ejemplos
Rate	One-shot		Con muy pocos bytes se detiene el servicio	• Un ataque de complejidad algorítmica • Un exploit de DoS • Un formulario muy mal programado
	Constante		Se observa un flujo constante de interacción	• Botnet
	Variable	Fluctuante	Varía en el tiempo, por lo que en ocasiones es muy difícil de detectar	• Anonymous • Un ataque perpetrado aprovechando horas punta
		Incremental	A medida que pasa el tiempo va en aumento	• Anonymous • Un gusano
Caracterizable	Filtrable		El ataque se puede identificar y filtrar	• Quick Win!
	No filtrable		El ataque se identifica (más o menos), pero en el fondo es tráfico legítimo	• LOIC
	No caracterizable		Es imposible distinguir el tráfico legítimo del fiable	• Una conexión de red saturada debido a que los usuarios usan SW que camufla las conexiones como SSL.

Taxonomía

Taxonomía de ejemplo (5/5)

- La siguiente taxonomía es una proposición que podría servir durante la planificación de un análisis:

	Clasificación		Descripción	Ejemplos
Impacto	Disruptivo	Auto recuperable	El sistema es capaz de recuperarse sólo una vez finalizado el ataque	• Tenemos watchdog's o sistemas capaces de reconocer el fallo de los servicios y recuperarlos
		Intervención humana	Hay que darle al botón	• El sistema se queda en estado inestable y requiere ser reiniciado por un humano
		No recuperable	El sistema queda gravemente dañado y debe reconstruirse	• Se ha hecho un Phlash Dance
	Degradación del servicio		El servicio se ve degradado aunque aguanta estoicamente el ataque	• Una vez pasa el mal tiempo todo vuelve a la normalidad sin haber fallado nada realmente

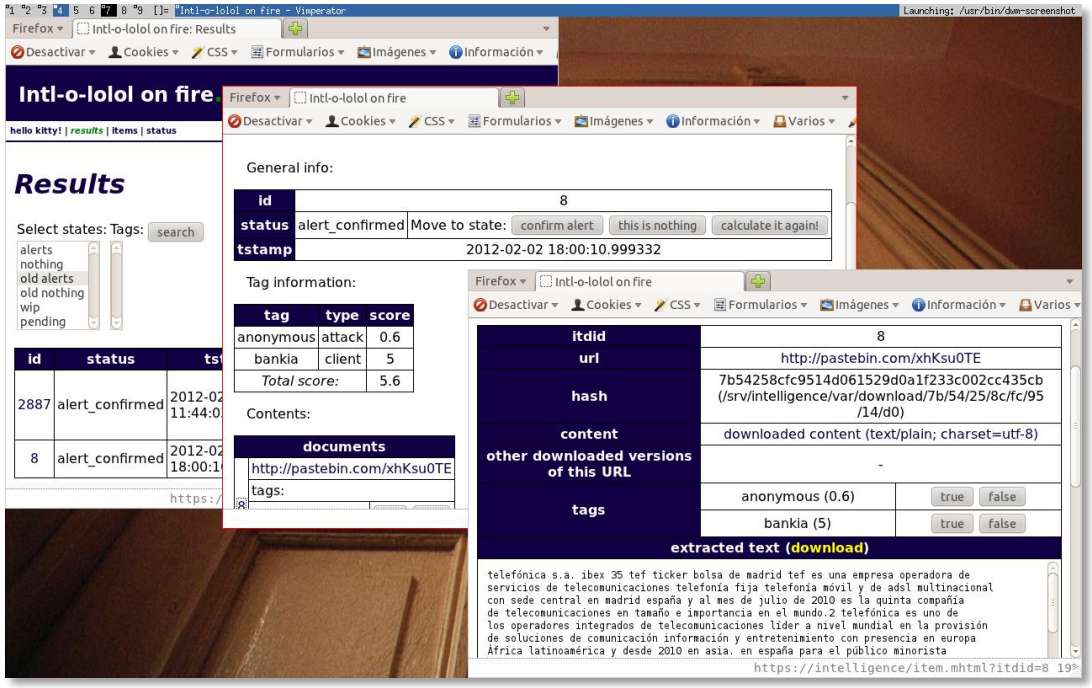
Agenda

- Introducción
- Metodología de Análisis
- Herramientas aportadas
- Taxonomía
- **Contramedidas**
- Clasificación, técnicas, herramientas y mitigación
- Bibliografía

Contramedidas

Detección del ataque, detección temprana

- La detección temprana nos puede permitir
  - Avisar a nuestros clientes
  - Desactivar partes del servicio
  - Activar mecanismos de anti DoS
  - Contratar especialistas y/o servicios (como Akamai o Telefónica)
  - Proteger ciertos activos (en especial en casos de malware)
  - En definitiva, **tomar medidas antes de un potencial desastre**.
- Ejemplo: Deloitte Intelligence
  - Recoge información de diferentes fuentes
  - Tagging mediante AI de documentos



Contramedidas

Detección del ataque, detección de patrones

- Anomalías o inferencia
  - Mediante un IPS/IDS
  - Light-weight detection [LWDETECT] basado en BLINC [BLINC]
  - Aporta al sistema la capacidad de
    - ❖ Redirigir el tráfico a una red de escape (RTBH routing)
    - ❖ Suavizar la carga
      - ➢ Cambio a contenido estático
      - ➢ Activar producto de tercero
      - ➢ DOYS, desafíos o incluso Overlay Networks
- Tomar medidas a nivel de programación en los servicios
- Ejemplo en ano.lolcathost.org:
  - Con mod_evasive se hace saltar un trigger que indica que estamos bajo ataque
  - Automáticamente la interfaz AJAX exige, si el cliente no está verificado, que resuelva un problema de hashes (que le llevará unos segundos).

    Las escrituras en ano.lolcathost.org están centralizadas a través de Ajax
  - Una vez entrega el resultado se valida durante un tiempo al cliente para que opere de forma normal.

Contramedidas

Detección del ataque, third party detection

- Externalizamos la gestión o nos enteramos por terceros
  - Hay ejemplos académicos
    - Backscatter analysis [CAIDA]
      - Útil en casos de IP-Spoofing: nodos externos a nuestra red se ven “salpicados” por respuestas nuestras evidenciando un posible ataque de DoS.
  - Servicios de terceros como Akamai
    - Detectan el ataque
    - Lo caracterizan
    - Y usan sus servicios para pararlo/gestionarlo.

Contramedidas

Planteándose una contramedida

	Preguntas
Efectividad	¿Como es capaz el mecanismo de defensa de mitigar un ataque DoS?
Fiabilidad	¿Es siempre efectivo mitigando, o a veces no tanto?
	¿Hay posibilidad de falsos positivos?
Subvertibilidad	¿Un atacante podría aprovechar este mecanismo como una herramienta inesperada para perpetrar una DoS?
Daño colateral	¿Puede este mecanismo de defensa causar efectos negativos, como penalización del rendimiento o necesidad de excesiva intervención humana para solventar sus limitaciones?
Proactividad	¿Puede prevenir ataques nuevos o des conocidos o sólo es capaz de reaccionar a patrones conocidos?
Completitud	¿Qué otros mecanismos de defensa pueden ser necesarios? (ej:un mecanismo detectivo requiere otro reactivo)
Tiempo de respuesta	¿Con que velocidad actúa?
Facilidad de implementación	¿Es factible o posible implementarlo?
	¿Afecta a más organizaciones?
	¿Justifica el esfuerzo?
Facilidad de uso	¿Tiene una interfaz de usuario fácil de usar?
	¿Encaja el mecanismo dentro de nuestra infraestructura?
Lugar de instalación	¿Cuál es el lugar óptimo para su instalación?

(fuente [MITDOS])

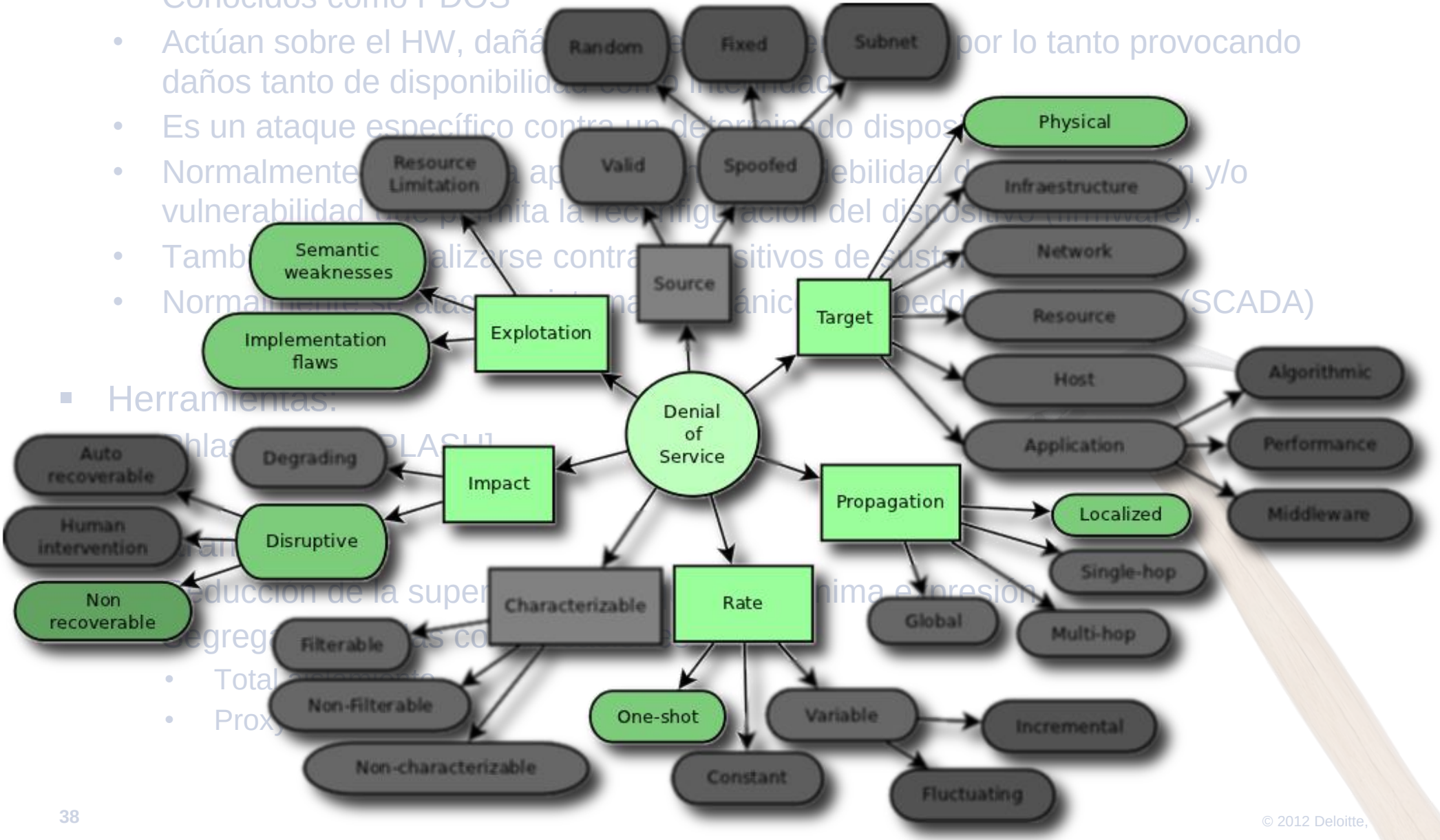
Agenda

- Introducción
- Metodología de Análisis
- Herramientas aportadas
- Taxonomía
- Contramedidas
- **Clasificación, técnicas, herramientas y mitigación**
- Bibliografía

Clasificación, técnicas, herramientas y mitigación

Nivel físico

- Descripción
  - Conocidos como PDOS
  - Actúan sobre el HW, dañá[illegible] por lo tanto provocando daños tanto de disponibilidad como de integridad.
  - Es un ataque específico contra un determinado dispos[illegible]
  - Normalmente [illegible] ap[illegible]ebilidad d[illegible]n y/o vulnerabilidad que permita la reconfiguración del dispositivo (firmware).
  - Tamb[illegible]alizarse contra [illegible]sitivos de sust[illegible]
  - Norma[illegible]ataq[illegible]na[illegible]ánic[illegible]bedd[illegible](SCADA)
- Herramientas:
  - [illegible]hlas[illegible]PLASH]
  - [illegible]
  - Reducción de la super[illegible]nima expresión
  - Segrega[illegible]s co[illegible]
    - Total[illegible]
    - Proxy[illegible]



Clasificación, técnicas, herramientas y mitigación

Nivel físico

- Descripción
  - Conocidos como PDOS
  - Actúan sobre el HW, dañándolo irremediablemente, y por lo tanto provocando daños tanto de disponibilidad como integridad
  - Es un ataque específico contra un determinado dispositivo.
  - Normalmente se ejecuta aprovechando una debilidad de configuración y/o vulnerabilidad que permita la reconfiguración del dispositivo (firmware).
  - También puede realizarse contra dispositivos de sustentación.
  - Normalmente se ataca a sistemas mecánicos, embedded o similares (SCADA)

- Herramientas:
  - PhlashDance [PLASH]

- Contramedidas:
  - Reducción de la superficie de ataque a la mínima expresión
  - Segregación de las comunicaciones
    - Total aislamiento
    - Proxys

Clasificación, técnicas, herramientas y mitigación

Infraestructura: semantic weaknesses

- Un protocolo o algoritmo puede sufrir debilidades de diseño, difíciles de resolver sin romper la compatibilidad
- Ejemplos:
  - SSLv2
  - TCP/IP
  - ISAKMP aggr de [illegible]
- Se ha co una moda e eguridad ya qu
  - no suele ser una s entie
  - es or imagen y es
  - el impacto es altísimo por la imp d de cambiar el prot
  - por contra s suele apar vos:
    - ikescan
    - naphta
    - thc-ssl-do

Clasificación, técnicas, herramientas y mitigación

Infraestructura: semantic weaknesses

- Un protocolo o algoritmo puede sufrir debilidades de diseño, difíciles de resolver sin romper la compatibilidad
- Ejemplos:
  - SSLv2
  - TCP/IP
  - ISAKMP aggressive mode [DOSPK], …
- Se ha convertido en una moda en la seguridad ya que
  - no suele ser útil a *script-kiddies* (no lo entienden),
  - es investigación, por lo que da imagen y es cool,
  - el impacto es altísimo por la imposibilidad de cambiar el protocolo,
  - y por ello las contramedidas suelen ser apaños disruptivos.
- Herramientas:
  - dosis
  - sockstress
  - ikescan
  - naphta
  - thc-ssl-dos

  Connection flood
  Zero window connection
  Small window
  Segment hole
  Req fin pause
  Activate reno pressure
  Stacheldraht

Clasificación, técnicas, herramientas y mitigación

Infraestructura: semantic weaknesses

- Mitigación:
  - Canales secundarios para misiones críticas
  - Desactivar funcionalidades
    - ✓ Desactivar SSLv2
    - ✓ Aggressive mode (ISAKMP)
  - Plan de trabajo que, una vez entendido el ataque, permita activar RTBH [RFC5635]
  - Tunning de TCP/IP
    - ✓ Syncookies
    - ✓ Timeout 3way hs, timeout fin states, timeout unused connections, …
      - Ejemplo en Linux:

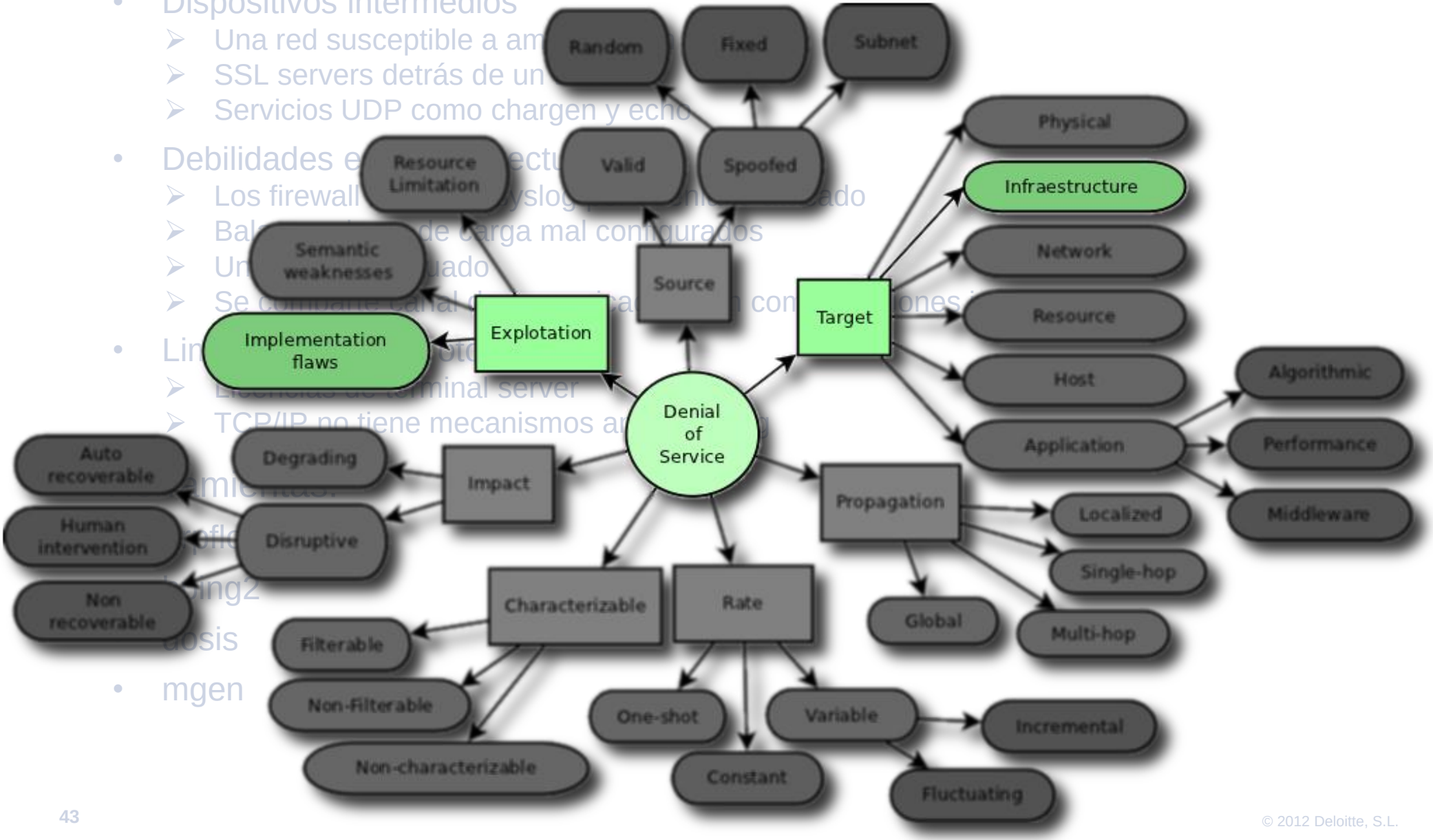
Linux TCP/IP stack parameter	Default	Example
/proc/sys/net/ipv4/tcp_keepalive_time	7200	30
/proc/sys/net/ipv4/tcp_keepalive_probes	9	2
/proc/sys/net/ipv4/tcp_max_ka_probes	5	100
/proc/sys/net/ipv4/tcp_syncookies	0	1

  - Actualizar a una nueva versión del protocolo (en cuanto exista)

Clasificación, técnicas, herramientas y mitigación

Infraestructura: implementation flaws

- Se puede sufrir de distintas formas:
  - Dispositivos intermedios
    - Una red susceptible a am
    - SSL servers detrás de un
    - Servicios UDP como chargen y echo
  - Debilidades e ectu
    - Los firewall yslog
    - Bal de carga mal configurados
    - Un uado
    - Se con ones
  - Lim
    - terminal server
    - TCP/IP no tiene mecanismos a
  - mi
  - fl
  - ing2
  - dosis
  - mgen



Clasificación, técnicas, herramientas y mitigación

Infraestructura: implementation flaws

- Se puede sufrir de distintas formas:
  - Dispositivos intermedios
    - Una red susceptible a amplificación
    - SSL servers detrás de un DNS RR
    - Servicios UDP como chargen y echo
  - Debilidades en la arquitectura
    - Los firewall lanzan el syslog por el enlace atacado
    - Balanceadores de carga mal configurados
    - Un proxy mal situado
    - Se comparte canal de comunicación con comunicaciones internas
  - Limitaciones del protocolo:
    - Licencias de terminal server
    - TCP/IP no tiene mecanismos anti-spoofing
- Herramientas:
  - arpflood
  - hping2
  - dosis
  - mgen

Clasificación, técnicas, herramientas y mitigación

Infraestructura: implementation flaws

- Contramedidas:
  - Disponer de control de acceso si es posible.
  - Desactivar funcionalidades que permitan alcanzar dicha limitación.
  - Instalar appliances de seguridad que limiten el uso o explotación indiscriminada.
  - Rediseño de red.
  - Optimización de los mecanismos:
    - ✓ Elegir parámetros agresivos en el firewall
    - ✓ Desactivar algoritmos costosos (SSL)
    - ✓ Trabajar la configuración de HTTP
    - ✓ Introducir límites que neutralicen el efecto nocivo de alcanzar otros límites

Clasificación, técnicas, herramientas y mitigación

Infraestructura: bruteforce

- Descripción:
  - Se genera un importante tráfico que afecta a la estabilidad del servicio al sobrecargar un activo dete
  - Esta categoría incluye Do S al uso o forzado del protocolo.
  - Familias: SYN, UDP, ICMP flood
- Herramientas:
  - dosis
  - mgen
  - hping2
- Contra
  - El objetivo aquí es poder manej ico de masa cr , la peor carga que e uede llegar a so stra infraestru ata balancear el rendimiento Mejo ión geográfica, replicar activ no de Minim ndencias y efectos ping-pong. egregar la gestión de en separadas. Applian guridad Preparar s superiores para aguanta estos ataques.
  - Labrea

Clasificación, técnicas, herramientas y mitigación

Infraestructura: bruteforce

- Descripción:
  - Se genera un importante tráfico que afecta a la estabilidad del servicio al sobrecargar un activo determinado
  - Esta categoría incluye DoS y DDoS basados en un mal uso o forzado del protocolo.
  - Familias: SYN, UDP, ICMP flood
- Herramientas:
  - dosis
  - mgen
  - hping2
- Contramedidas:
  - El objetivo aquí es poder manejar un tráfico de masa crítica, es decir, la peor carga que en teoría puede llegar a soporta nuestra infraestructura..
  - Tratar la cadena de comunicaciones como un todo para balancear el rendimiento.
  - Mejorar disposición geográfica, replicar activos y rediseño de red.
  - Minimizar dependencias y efectos ping-pong.
  - Segregar la gestión del protocolo en capas bien separadas.
  - Appliances de seguridad (Checkpoint)
  - Preparar las capas superiores para aguantar estos ataques.
  - Labrea

Clasificación, técnicas, herramientas y mitigación

Red: bruteforce

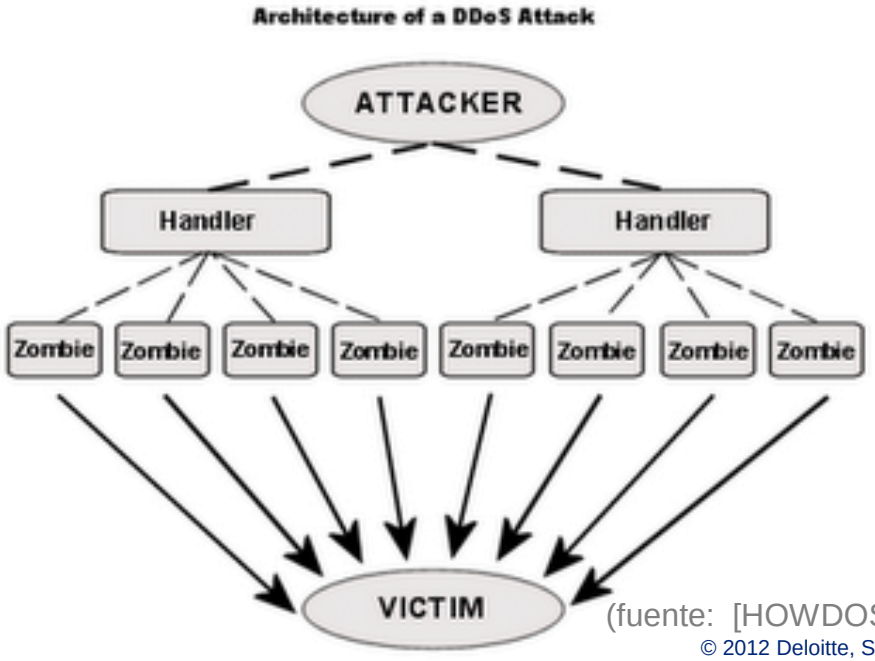
- Para llegar a este nivel debemos asumir que nuestro sistema funciona bien en la capa de infraestructura con una carga del 100% y la capas superiores son capaces de no fallar.
- En los ataques de red asumimos solamente ataques [illegible] nuestra capacidad de ab[illegible] y re[illegible] el ancho de banda.
- De estas situaciones normalmente no podemos defendernos solos:
  - ISP o [illegible]pecializados (A[illegible], Amazon)
  - Si no p[illegible]obtener ayuda n[illegible]objetivo puede ser [illegible]taque sin efe[illegible]degradación [illegible]vicio.
- Ataques clasicos (DDOS):
  - Ampli[illegible]ttacks
  - Botne[illegible]
  - Vorm[illegible]
  - Hack[illegible]nonymous)
  - Herramientas:
    - hping2
    - chucuc[illegible]
    - LOIC
    - Trinoo / Tr[illegible]twork)

(fuente: [HOWDOS])

Clasificación, técnicas, herramientas y mitigación

Red: bruteforce

- Para llegar a este nivel debemos asumir que nuestro sistema funciona bien en la capa de infraestructura con una carga del 100% y la capas superiores son capaces de no fallar.
- En los ataques de red asumimos solamente ataques que exceden nuestra capacidad de absorción y respuesta: agotan el ancho de banda.
- De estas situaciones normalmente no podemos defendernos solos:
  - ISP o servicios especializados (Akamai, Amazon)
  - Si no podemos obtener ayuda nuestro objetivo debe ser aguantar el ataque sin efectos disruptivos – sólo una degradación del servicio.
- Ataques clásicos (DDOS):
  - Amplification attacks
  - Botnets/Zombies
  - Worms
  - Hacktivismo (Anonymous)
- Herramientas
  - hping2
  - chucuchu
  - LOIC
  - Trinoo / TFN (Tribe Flood Network)



(fuente: [HOWDOS])

Clasificación, técnicas, herramientas y mitigación

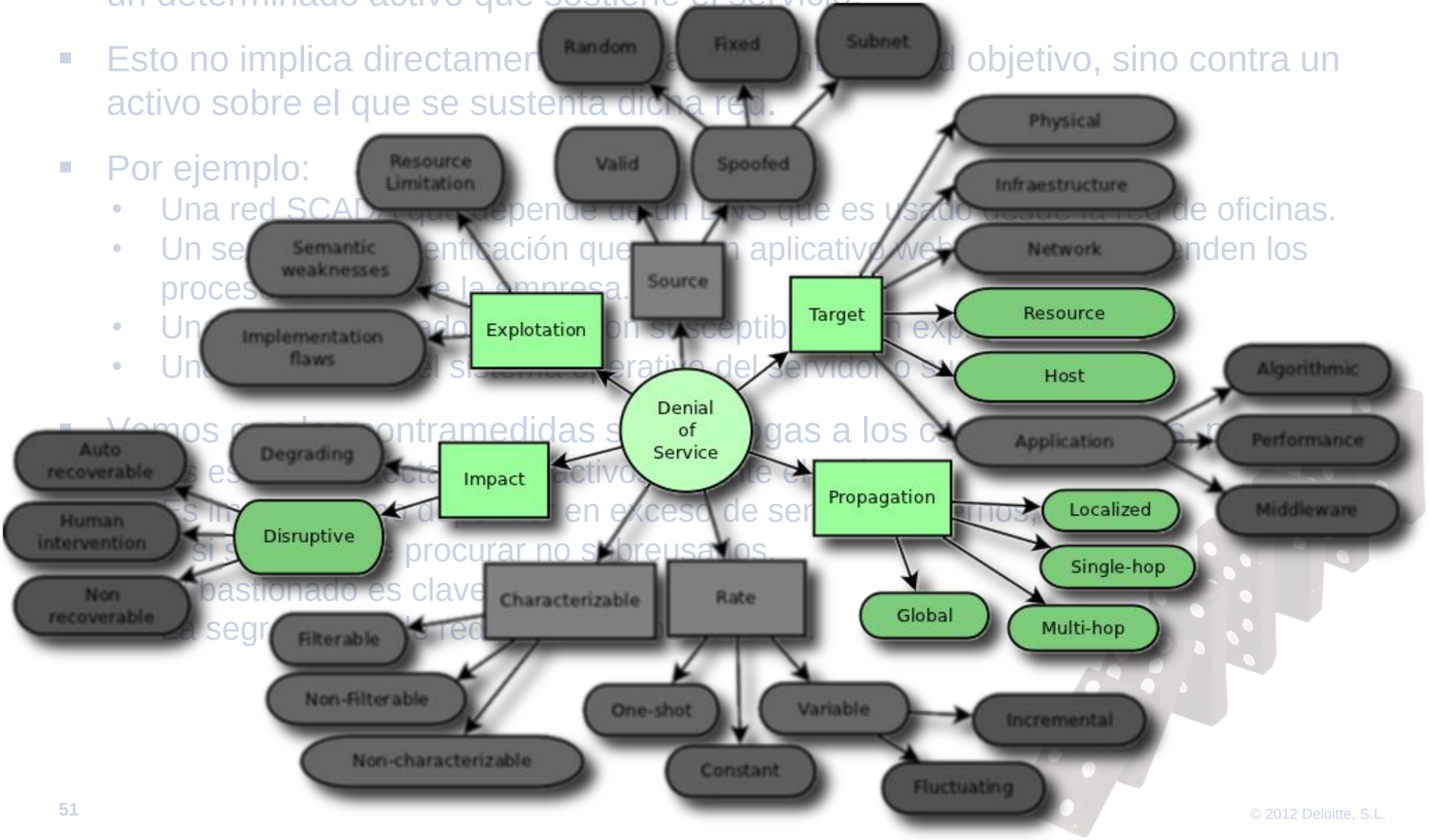
Red: bruteforce

- Contramedidas:
  - Es esencial disponer de un **plan**, claro y suficientemente flexible para poder reaccionar y **personas** capaces de ejecutarlo.
  - Es indispensable implementar una configuración estricta y preparada en el firewall:
    - ✓ Smurf Amplifier Registry (http://smurf.powertech.no/)
    - ✓ Reglas ingress, egress
    - ✓ Limitar bandwidth por cliente
    - ✓ Limitar conexiones por cliente
    - ✓ Políticas restrictivas (DROP)
    - ✓ Técnicas que dificulten el escaneo y propagación (labrea)
  - Tratar de atajar el problema en el punto más cercano al ataque.
  - Uso de overlay networks o redes distribuidas (Akamai)
    - ✓ P.ej. Akamai provee de servicios de aceleración y anti-DOS a nivel de red (Akamai DDoS Defender)
    - ✓ Cloud Computing
  - Dar servicio por diferentes canales según origen geográfico (GeoDNS)
  - Implementar RTBH Routing

Clasificación, técnicas, herramientas y mitigación

Recursos y Host

- Los ataques de DoS contra un recurso o host focalizan sus esfuerzos contra un determinado activo que sostiene el servicio.
- Esto no implica directamente a [illegible] objetivo, sino contra un activo sobre el que se sustenta dicha red.
- Por ejemplo:
  - Una red SCADA que depende de un DNS que es usado desde la red de oficinas.
  - Un se[illegible]enticación que [illegible] aplicativo web [illegible]enden los proces[illegible]e la empresa.
  - Un[illegible]ado [illegible]on susceptib[illegible]n exp[illegible]
  - Un[illegible]el sistema operativo del servidor o su[illegible]
- Vemos [illegible]ontramedidas s[illegible]ogas a los c[illegible]s, p[illegible]s es[illegible]ecta[illegible]activos[illegible]te e[illegible]
  Es im[illegible]d [illegible]en exceso de ser[illegible]ernos, si s[illegible]e procurar no s[illegible]brepasados.
  [illegible]bastionado es clave[illegible]
  [illegible]a segr[illegible]de red[illegible]



Clasificación, técnicas, herramientas y mitigación

Recursos y Host

- Los ataques de DoS contra un recurso o host focalizan sus esfuerzos contra un determinado activo que sostiene el servicio.
- Esto no implica directamente un ataque contra la red objetivo, sino contra un activo sobre el que se sustenta dicha red.
- Por ejemplo:
  - Una red SCADA que depende de un DNS que es usado desde la red de oficinas.
  - Un servicio de autenticación que usa un aplicativo web y del cual dependen los procesos críticos de la empresa.
  - Una red de ordenadores que son susceptibles a un exploit remoto.
  - Una debilidad en el sistema operativo del servidor o sus protocolos.
- Vemos que las contramedidas son análogas a los casos anteriores, pero:
  - Es esencial detectar estos activos durante el análisis.
  - Es importante no depender en exceso de servicios externos,
  - Y si se depende procurar no sobreusarlos.
  - El bastionado es clave.
  - La segregación de redes y servicios también.

Clasificación, técnicas, herramientas y mitigación

Aplicación: introducción

- Este nivel suele dar más juego ya que normalmente no se protege contra DOS
  - A pesar de que el impacto en este nivel suele ser más crítico
  - Y las condiciones de DoS más fáciles de conseguir
- Normalmente es debido a:
  - Un diseño ineficiente o pobre
  - Una codificación pobre
  - Un backend no preparado o mal usado
  - Un exceso en el uso de contenidos dinámicos

- Generalmente en este nivel los resultados son espectaculares.
  - En varios proyectos se ha demostrado que se podía tumbar Application Servers con una conexión ADSL de 1999.
  - En uno de ellos el diseño de red aguantaba más de 2 Gbps.
- Normalmente englobamos en este nivel toda el software por encima del SO:
  - base de datos
  - apache
  - servidor de aplicaciones
  - librerías y aplicación

Clasificación, técnicas, herramientas y mitigación

Aplicación: vulnerabilidades DoS

- Es el tipo de DoS más efectivo y devastador.
- Especialmente porque cualquier vulnerabilidad que subvierta el funcionamiento en lado servidor, puede ser explotada para DoS.
- El resultado puede variar desde temporal, pasando por denegación humana, hasta la destrucción física de equipos e incluso [illegible]
- Existen [illegible] herramientas para conseguirlo, automatizadas como metasploit, [illegible] facilidad de uso.
- Las contramedidas [illegible]
  - Minimizar la superficie de ataque.
  - Implementar control de acceso [illegible] todo lo que no [illegible]
  - Proteger protocolos [illegible] lejos mediante cortafuegos de seguridad como proxies o [illegible] seguridad.
  - Realizar pentests y [illegible] de vulnerabilidades.
  - Proteger [illegible] mediante políticas de seguridad.
  - Tener un [illegible] ágil y constante de parcheo y actuación.
  - Prestar [illegible] noticias [illegible]

Clasificación, técnicas, herramientas y mitigación

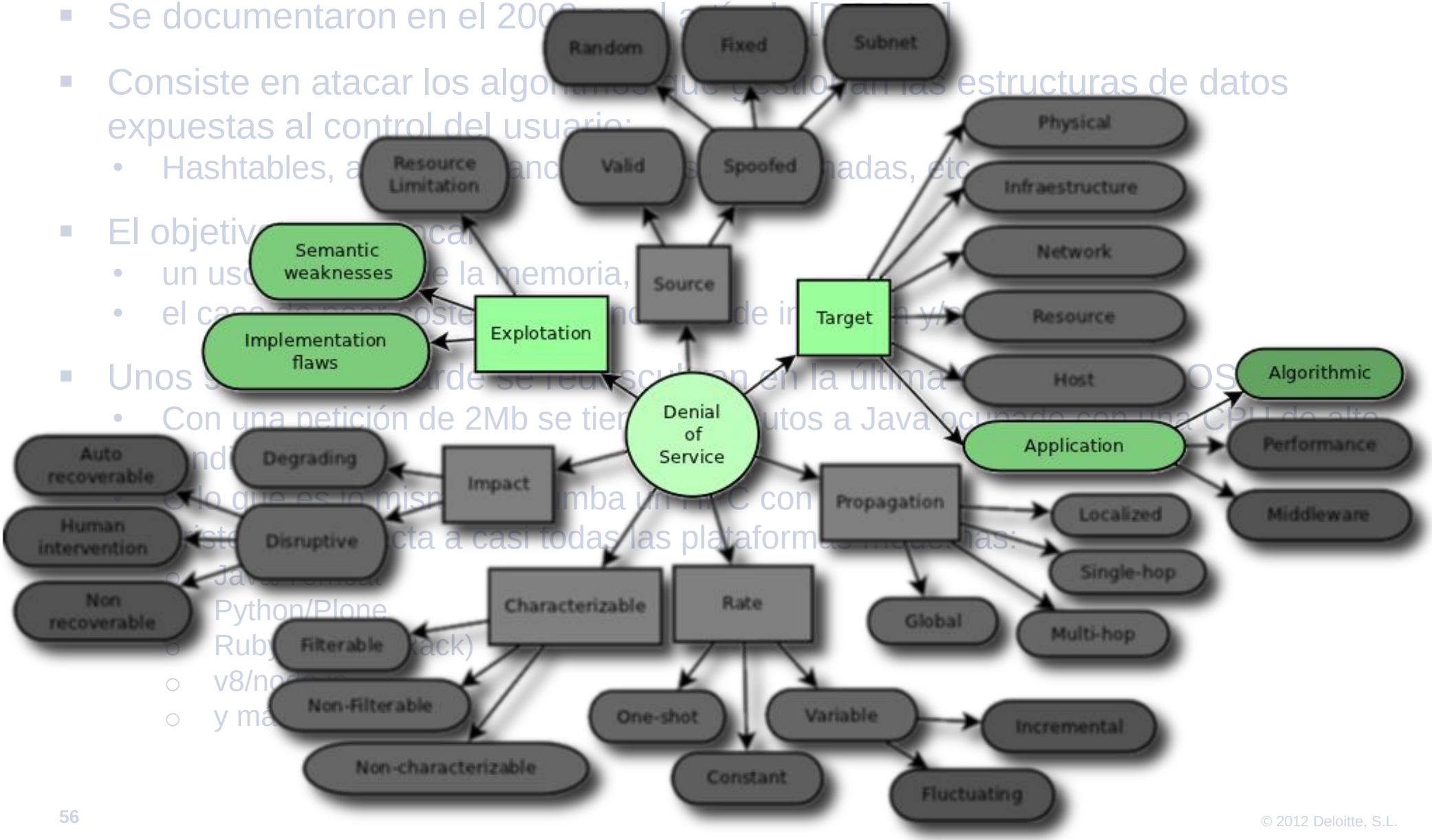
Aplicación: vulnerabilidades DoS

- Es el tipo de DoS más efectivo y devastador.
- Especialmente porque cualquier vulnerabilidad que subvierta el funcionamiento en lado servidor puede implicar una DoS.
- El resultado puede variar desde temporal, pasando por intervención humana, hasta la destrucción lógica del activo (a veces incluso física).
- Existen cientos de herramientas para conseguirlo, aunque destacamos metasploit por su facilidad de uso.
- Las contramedidas claves son:
  - Minimizar la superficie de ataque.
  - Implementar control de acceso sobre todo lo que no sea de uso público.
  - Proteger protocolos complejos mediante capas de seguridad como proxies o appliances de seguridad.
  - Realizar pentests y escaneos periódicos de vulnerabilidades.
  - Proteger los activos más importantes detrás de diversas capas de seguridad.
  - Tener un protocolo ágil y constante de parcheo y actuación.
  - Prestar atención a noticias relacionadas (cyber-intelligence)

Clasificación, técnicas, herramientas y mitigación

Aplicación: DoS algorítmico (1/2)

- Son ataques basados en explotar el coste computacional de los algoritmos.
- Se documentaron en el 200[illegible] [Crosby]
- Consiste en atacar los algoritmos que gestionan las estructuras de datos expuestas al control del usuario:
  - Hashtables, [illegible]nadas, etc.
- El objetivo [illegible]
  - un uso [illegible]e la memoria,
  - el caso de peor coste [illegible]de ir [illegible]a y/
- Unos [illegible]rde se reducen en la última [illegible]OS
  - Con una petición de 2Mb se tie[illegible]utos a Java ocupado con una CPU de alto
  - Solo que es lo mismo [illegible]mba un HTTP con
  - Esto afecta a casi todas las plataformas modernas:
    - Java (Tomcat)
    - Python/Plone
    - Ruby (Rack)
    - v8/node
    - y ma[illegible]



Clasificación, técnicas, herramientas y mitigación

Aplicación: DoS algorítmico (1/2)

- Son ataques basados en explotar el coste computacional de los algoritmos.
- Se documentaron en el 2003 en el artículo [DOSAL]
- Consiste en atacar los algoritmos que gestionan las estructuras de datos expuestas al control del usuario:
  - Hashtables, arboles balanceados, listas ordenadas, etc
- El objetivo es provocar
  - un uso irracional de la memoria, o
  - el caso de peor coste en las funciones de inserción y/o búsqueda.
- Unos 9 años más tarde se redescubren en la última CCC [HASHDOS]:
  - Con una petición de 2Mb se tiene 44 minutos a Java ocupado con una CPU de alto rendimiento.
  - O lo que es lo mismo, se tumba un HPC con 6kbit/s
  - Tristemente afecta a casi todas las plataformas modernas:
    - Java/Tomcat
    - Python/Plone
    - Ruby/CRuby (Rack)
    - v8/node.js
    - y más

Clasificación, técnicas, herramientas y mitigación

Aplicación: DoS algorítmico (2/2)

- La mitigación de esta amenaza se consigue:
  - Toda entrada de usuario es maligna, por ello este no debería poder escogerla libremente más que en contadas excepciones y con LIMITES.
  - Siempre se recomienda o bien limitar la entrada, o bien detectar si vamos de cabeza a una situación límite, abortando la petición (timeouts, preanálisis, etc).
  - Estos límites se pueden establecer fácilmente en el caso de [HASHDOS] con ayuda del módulo mod_security [MSHASHDOS]:
    - ✓ Limitar los POST que no incluyan ficheros a un tamaño máximo
    - ✓ Limitar el número de parámetros de entrada
  - Si no se pude establecer límites si podemos:
    - ✓ Introducir controles que impidan una ejecución indiscriminada de las operaciones intensivas
    - ✓ Cachear resultados
    - ✓ Trabajar con colas de ejecución sobre las que es más sencillo establecer controles de inanición y hacer que la capa de presentación trabaje de forma asíncrona.
      - Esta técnica se aplica en Ano para el pre-análisis de seguridad, procesado y encolamiento de imágenes

Clasificación, técnicas, herramientas y mitigación

Aplicación: localidad e información de estado

- Debido a que HTTP es un protocolo *stateless* se ha escrito y se escribirá lo infinito para resolver el problema de conservar el estado en las transacciones.
- Pero el verdadero problem[illegible] valora atentamente que es necesario mantener y como mantener la información [illegible] base de datos.
- El estado puede [illegible]sla de diversas formas:
  - Es co[illegible]eral
  - Si se centraliza en u[illegible]ocar [illegible]rhead
  - Si [illegible] demos tener [illegible]mas de sincronismo entre servidores (ping pong) o acceso[illegible]tos.
  - Si represento un *working-set* gr[illegible] puede matar[illegible]
  - Contramedidas:
    - Usar un *working-set* que mantenga la informa[illegible]cial.
    - Que no forme parte de[illegible]ral ni que requi[illegible]nua actualización.
    - Potencia[illegible] de sistemas como VIEWSTATE o COOKIES clásicas (pero firmado y/o cifra[illegible]or de Di[illegible]
    - No iniciar [illegible]cambio una sesión o estado.
    - Dificultar la [illegible] estados me[illegible]safío[illegible]etscaler).

Clasificación, técnicas, herramientas y mitigación

Aplicación: localidad e información de estado

- Debido a que HTTP es un protocolo *stateless* se ha escrito y se escribirá lo infinito para resolver el problema de conservar el estado en las transacciones.
- Pero el verdadero problema resulta cuando no se valora atentamente que es necesario mantener y como mantener la información de sesión y base de datos.
- El estado puede jugárnosla de diversas formas:
  - Es costoso de generar.
  - Si se centraliza en una BD puede provocar un overhead global.
  - Si se localiza en el servidor podemos tener problemas de seguimiento entre servidores (ping-pong) o accesos remotos.
  - Si representa un *working-set* grande nos puede matar la aplicación.
- Contramedidas:
  - Usar un *working-set* que mantenga la información esencial.
  - Que no forme parte de la base de datos central ni que requiera su continua actualización.
  - Potenciar el uso de sistemas como VIEWSTATE o COOKIES clásicas (pero firmado y/o cifrado, por el amor de Dios).
  - No iniciar a la primera de cambio una sesión o estado.
  - Dificultar la creación de estados mediante **desafíos** (Citrix Netscaler).

Clasificación, técnicas, herramientas y mitigación

Aplicación: exceso de contenido dinámico

- En la actualidad parece que es un tabú plantearse páginas web estáticas.
- Pero son las que mejor resisten un elevado número de visitas: sólo hay que servirlas.
- Adicionalmente se tiende a delegar en un solo servidor la generación de una página, por lo que detrás de un balanceador tendrá toda la entrega de páginas.
- Una estrategia sencilla, hecha a medias: antes dividíamos los recursos en diferentes categorías:
  - Co
  - Propaganda
- Siempre es recomendable que la entrada sea estática.
- Truco: si sirves incrementos establecidos contra una página, servir una versión estática.

Clasificación, técnicas, herramientas y mitigación

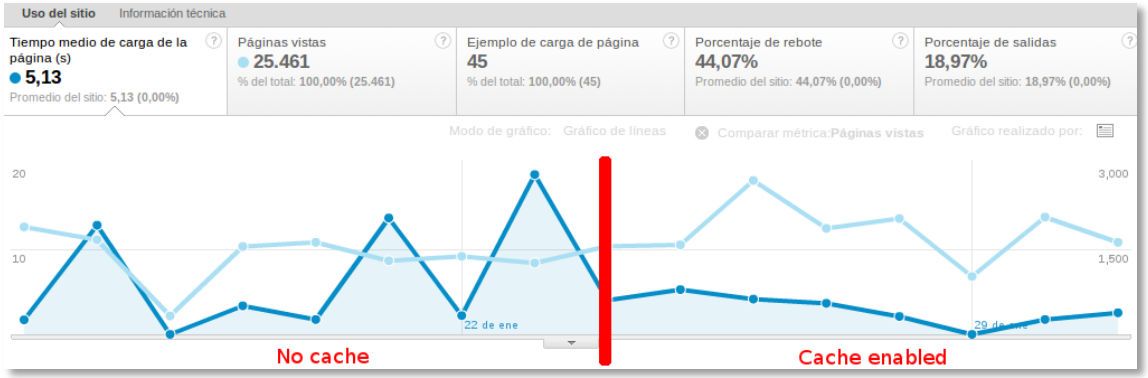
Aplicación: exceso de contenido dinámico

- En la actualidad parece que es un tabú plantearse páginas web estáticas.
- Pero son las que mejor resisten un pico importante de visitas: sólo hay que servirlas.
- Adicionalmente se tiende a delegar en un solo servidor la generación de una página, por lo que si uno de los servidores de detrás se satura, detendrá toda la entrega de páginas.
- Una estrategia y sencilla, hecha desde antaño, es dividir los recursos en diferentes servidores:
  - Contenido estático
  - Propaganda
  - Página principal
  - Otros elementos se pueden cargar bajo demanda a través de Ajax
- Siempre es recomendado que el portal de entrada sea estático.
  - La mayor parte de ataques actuales se dirigen sólo contra la página principal.
- Truco: si se detecta un incremento notable de peticiones contra una página, servir una versión estática.

Clasificación, técnicas, herramientas y mitigación

Aplicación: siguiente paso, el caching

- Una página normalmente en su 90% es estática.
- Podemos implementar o usar un sistema de caching que reduzca en su mayor parte el tiempo de servir una página:
  - Cache-Cache (mod_perl/Mason HQ)
  - Varnish (web accelerator, genérico)
- Herramientas como Varnish Cache son uno de los pilares básicos de entornos de acelaración web (Akamai implementa algo parecido).
- Pueden programarse para
  - Cachear contenidos dinámicos
  - Precachear contenidos dinámicos que suelen ser pedidos.
  - Decorar las páginas servidas para que el servidor no tenga que preocuparse de renderizar la página final y así descargarle trabajo.
  - O incluso programar políticas sobre como deben ser servidos y procesados determinadas peticiones bajo determinada carga.



Clasificación, técnicas, herramientas y mitigación

Aplicación: no poner todos los huevos en la misma cesta

- En Internet es fácil terminar teniendo visitas de **todo el mundo**.
- Pero no tenemos porque dar el mismo servicio a todo el mundo.
- Es interesante, siempre que
  - Dispersar geográficamente el servicio
    (OJO: sin
  - Atender a los usuarios lo más cerca posible
    - ✓ Pa
    - ✓
    - ✓

Esto nos
áfico en una zon
al
clientes/países/localida

Clasificación, técnicas, herramientas y mitigación

Aplicación: no poner todos los huevos en la misma cesta

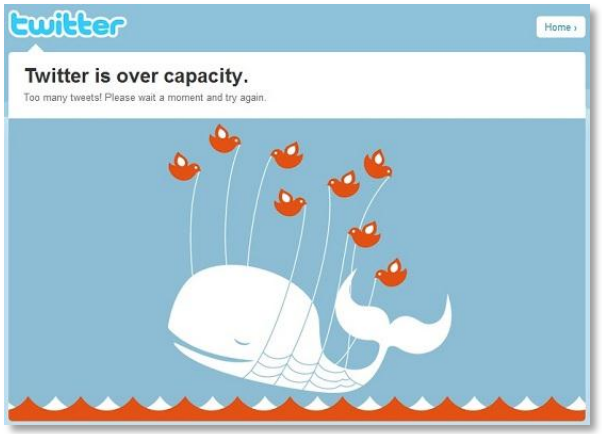
- En Internet es fácil terminar teniendo visitas de **todo el mundo**.
- Pero no tenemos porque dar el mismo servicio a todo el mundo.
- Es interesante, siempre que se pueda,
  - Dispersar geográficamente el servicio

    (OJO: sin que por ello centralizamos todo en un mismo subsistema!)
  - Atender a los usuarios lo más cerca posible
    - ✓ Para disminuir la latencia
    - ✓ Servir más rápido
    - ✓ Y sólo caer donde tengamos un ataque
- Esto nos brinda la posibilidad de adaptarnos según tengamos un incremento de tráfico en una zona del planeta u otra.
- Lo cual hace posible seguir ofreciendo el servicio a nuestros clientes/países/localidades más importantes, aunque nos esté cayendo una buena del resto del mundo.

Clasificación, técnicas, herramientas y mitigación

Aplicación: y si no se puede, no se puede

- En el peor caso siempre podemos huir o buscar ayuda antes que autodestruirnos.
- Existen varias aproximaciones a esto:
  - Denial You of Service [PWDOS]
    - ✓ El servidor toma nota de los clientes más “activos” y les corta el servicio.
  - Twitter is over capacity
    - ✓ La técnica de Twitter es parar antes que saturar.
    - ✓ El servicio no está pero lo tratamos elegantemente.



  - A hacer cola
    - ✓ Esta técnica es usada ampliamente por las páginas de descarga directa.
    - ✓ Pero también puede ser útil si fichamos un supercomercial que a la mínima nos pone a vender entradas de U2 o Madonna.
  - Contratamos un servicio Cloud que nos permita escalar
    - ✓ No obstante no todo es jauja, y para poder hacer esto deberemos adaptar nuestros servicios a entornos distribuidos o estaremos en las mismas.
    - ✓ No siempre es viable en tiempo y recursos.

Agenda

- Introducción
- Metodología de Análisis
- Herramientas aportadas
- Taxonomía
- Contramedidas
- Clasificación, técnicas, herramientas y mitigación
- **Bibliografía**

Bibliografía

Porque leo mucho ☺ (1/6)

[AKADDOS] ***Akamai DDoS Defender***,
http://www.akamai.com/html/solutions/ddos_defender.html

[AUTRES] ***Autonomic Response to Distributed Denial of Service Attacks***,
Dan Sterne, Kelly Djahandari, Brett Wilson, Bill Babson, Dan Schnackenberg, Harley Holliday, and Travis Reid.

[BIGDNS] ***GeoDNS BIND patch***
http://www.caraytech.com/geodns/

[BLINC] ***BLINC: Multilevel Traffic Classification in the Dark***,
Thomas Karagiannis, Konstantina Michalis Faloutsos, Papagiannaki

[CAIDA] ***Inferring Internet Denial-of-Service Activity***,
David Moore

[CACHE2] ***Cache-Cache***,
Jonathan Swartz
http://search.cpan.org/dist/Cache-Cache/

[DNSAMP] ***DNS Amplification Attacks***,
Randal Vaughn and Gadi Evron

[DOSAL] ***Denial of Service via Algorithmic Complexity Attacks***,
Scott A. Crosby & Dan S. Wallach

[DOSPK] ***Denial of service in public key protocols***,
Pasi Eronen

Bibliografía

Porque leo mucho ☺ (2/6)

[DPROT] ***DOS and DDOS protection***,
http://www.opensourcerack.com/2010/10/14/dos-and-ddos-protection/

[GEODNS] ***GeoDNS—Geographically-aware, protocol-agnostic load balancing at the DNS level***,
John Hawley

[HASHDOS] ***Efficient Denial of Service Attacks on Web Application Platforms***,
Alexander “alech” Klink & Julian “zeri” Wälde

[HOWDOS] ***Como hacer un ataque DDOS***,
Williams Melgar
http://www.comunidadbloggers.com/2011/02/como-se-hacer-un-ataque-ddos.html

[LABREA] ***LaBrea Tarpit***,
http://labrea.sourceforge.net/

[LEMMA] ***Method of Mitigating DDoS Attacks by Randomly choosing and Dynamically Changing Routing Information***,
Samsom Lemma

[LWDETECT] ***Lightweight Detection of DoS Attacks***,
Sirikarn Pukkawanna, Vasaka Visoottiviseth, Panita Pongpaibool

[MAYDAY] ***Mayday: Distributed Filtering for Internet Services***,
David G. Andersen

Bibliografía

Porque leo mucho ☺ (3/6)

[MGEN] ***Multi-Generator (MGEN)***”,
Naval Research Laboratory (NRL)
http://cs.itd.nrl.navy.mil/work/mgen/

[MEVASIVE] ***mod_evasive***,
Jonathan Zdziarski
http://www.zdziarski.com/blog/?page_id=442

[MITDOS] ***Mitigating denial of service attacks: A tutorial***,
Jarmo Mölsä

[MSHASHDOS] ***ModSecurity Mitigations for ASP.NET HashTable DoS Vulnerability (CVE-2011-3414)***,
http://blog.spiderlabs.com/2012/01/modsecurity-mitigations-for-aspnet-hashtable-dos-vulnerability-cve-2011-3414.html

[NAPHST] ***The NAPTHA DoS vulnerabilities***,
Bob Keyes (SecuriTeam)
http://www.securiteam.com/securitynews/6B0031F0KA.html

[NISTIHG] ***Computer Security Incident Handling Guide***,
Karen Scarfone, Tim Grance & Kelly Masone

[OVERCAST] ***Overcast: Reliable Multicasting with an Overlay Network***,
John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, James W. O’Toole Jr.

Bibliografía

Porque leo mucho ☺ (4/6)

[OVERDOSE] ***OverDoSe: A Generic DDoS Protection Service Using an Overlay Network***,
Elaine Shi, Ion Stoica, David Andersen, Adrian Perrig

[PAXSON] ***An Analysis of Using Reflectors for Distributed DoS Attacks***,
Vern Paxson

[PHLASH] ***PhlashDance: Discovering permanent denial of service attacks against embedded systems***,
Rich Smith, HP Labs

[POKEMON] ***Second Annual Cost of Cyber Crime Study; Benchmark Study of U.S. Companies; August 2011***
Sponsored by ArcSight(an HP Company) Independently conducted by Ponemon Institute LLC

[PWDOS] ***Prevent DoS attacks in your web application***,
Omar AL Zabir
http://omaralzabir.com/prevent_denial_of_service__dos__attacks_in_your_web_application/

[RFC2827] ***RFC2827 Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing***,
Ferguson & Senie

Bibliografía

Porque leo mucho ☺ (5/6)

[RFC5635] ***RFC 5635: Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)***,
W. Kumari & D. McPherson

[SECAPCH] ***20 ways to Secure your Apache Configuration***,
Pete Freitag,
http://www.petefreitag.com/item/505.cfm

[SLOWRD] ***ModSecurity Advanced Topic of the Week: Mitigation of 'Slow Read' Denial of Service Attack***",
http://blog.spiderlabs.com/2012/01/modsecurity-advanced-topic-of-the-week-mitigation-of-slow-read-denial-of-service-attack.html

[SOSDOS] ***SOS: An Architecture For Mitigating DDoS Attacks***,
Angelos D. Keromytis, Vishal Misra, Dan Rubenstein

[SRAW] ***SOCK RAW demystified***,
http://sock-raw.org/papers/sock_raw

[TAXCPS] ***A Taxonomy for Denial of Service Attacks in Content-based Publish/Subscribe Systems***",
Alex Wun, Alex Cheung & Hans-Arno Jacobsen

[TAXDOS] ***A Taxonomy of DDoS Attack and DDoS Defense Mechanisms***,
Jelena Mirkovic, Peter Reiher

[TDOST] **Trends in Denial of Service Attack Technology**,
CERT® Coordination; Center, Kevin J. Houle, & George M. Weaver

Bibliografía

Porque leo mucho ☺ (6/6)

[TFN] ***The 'Tribe Flood Network' distributed denial of service attack tool***, David Dittrich

[TFN2K] ***TFN2K - An Analysis***, Jason Barlow & Woody Thrower

[VARNISH] ***Varnish Cache***, http://www.varnish-cache.org/

Fin!

Hasta aquí hemos llegado

¿ALGUNA PREGUNTA?

Hablad ahora o escribidnos un correo a
Gerardo García Peña <ggarciapena@deloitte.es>

Agradecimientos

Sin su ayuda no hubiera sido posible

- A **Emet-Jon Velasco** por su ayuda y soporte para desarrollar esta presentación.
- A mi **equipo de Deloitte** por su ayuda, excelencia humana y profesional.
- A la **Nopcode** y **Summer Camp Garrotxa** donde se presentó por primera vez DoSIS.
- Al equipo de **ano.lolcathost.org** por cederme un escenario tan potente.
- A **Silvia** y a **Sergi** por el apoyo más importante. ☺
- A **Federico Dios** y **Patrick Sullivan** de Akamai por su atención prestada ante mis dudas.
- Y por supuesto a **todos los autores de la bibliografía** que sin ellos no hubiera sido tan rica en contenidos esta presentación.
- Y recordad: la seguridad no es siempre lo que parece, como la leyenda de Beaver Dam.

Si desea información adicional, por favor, visite www.deloitte.es